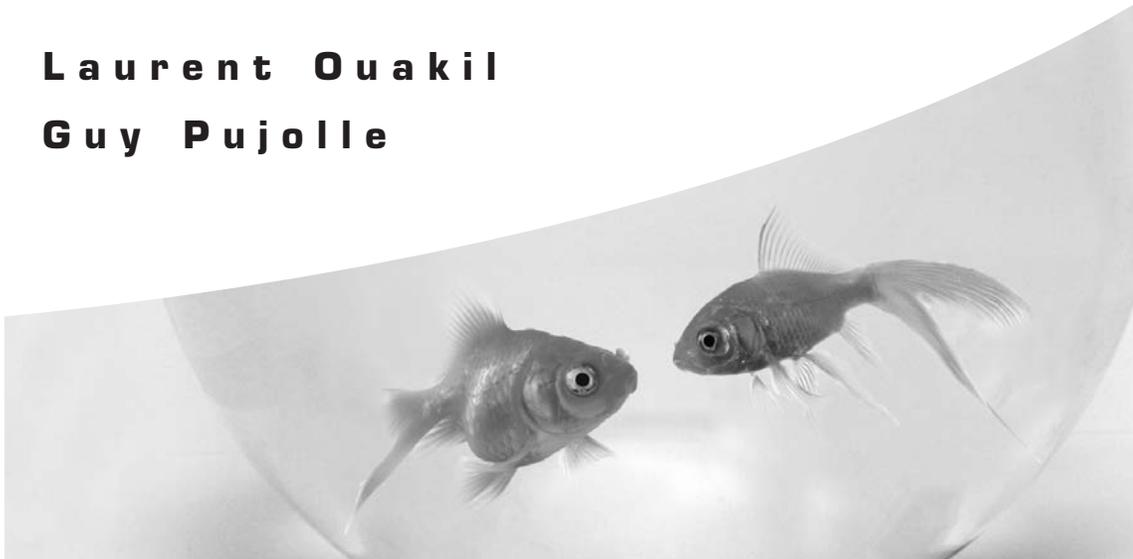


# Téléphonie sur IP

2<sup>e</sup> édition

**SIP, H.323, MGCP, QoS et sécurité,  
Asterisk, VoWiFi, offre multiplay des FAI,  
Skype et autres softphones, architecture IMS...**

**Laurent Ouakil  
Guy Pujolle**



**EYROLLES**

# Avant-propos

---

Lorsque, le 2 juin 1875, le Canadien Alexandre Graham Bell tente de transformer des ondes sonores en impulsions électromagnétiques, nul n'imagine que ce professeur de physiologie vocale, spécialisé dans l'enseignement du langage pour sourds et muets, allait inventer le téléphone.

Accompagné de son assistant Thomas Watson, Bell expérimente le premier modèle de téléphone à distance limitée et à correspondance réduite : placés dans deux pièces distinctes, les deux physiciens disposent entre eux un fil conducteur dont une extrémité est munie d'une lamelle reliée à un électroaimant. L'expérience consiste à écarter cette lamelle de l'électroaimant puis à la relâcher. Le résultat est prodigieux : un son se propage sur le fil conducteur jusqu'à parvenir à l'autre extrémité du fil. Il faudra moins d'un an au scientifique Bell, tout juste âgé de 28 ans, pour perfectionner son prototype et rendre les transmissions d'un bout à l'autre d'un fil conducteur parfaitement intelligibles pour l'oreille humaine.

Le 10 mars 1876, à Boston, Bell communique à distance avec son assistant en prononçant sa célèbre phrase : « Monsieur Watson, veuillez venir dans mon bureau, je vous prie. » Quelques mois plus tard, le téléphone entre dans sa phase de commercialisation. Des opératrices prennent en charge la demande de connexion et assurent la liaison entre les correspondants, et le succès est au rendez-vous.

En 1964, en pleine guerre froide, le projet d'un réseau informatique totalement distribué et dédié aux communications militaires est refusé par les autorités à son initiateur, Paul Baran. Presque en parallèle, les travaux du français Louis Pouzin, mettant au point le tout premier réseau à commutation de paquets, émule la communauté scientifique. Au début des années 70, un réseau imaginé par des laboratoires de recherche académiques voit le jour. Constitué de quatre ordinateurs répartis dans le monde, il est réalisé par l'ARPA (Advanced Research Projects Agency) et prend le nom d'ARPANET. Au même moment, en France, le projet Cyclades relie plusieurs ordinateurs par une technologie de data-gramme.

Ces prototypes démontrent la faisabilité du réseau mondial qui se développera sous le nom d'Internet, et dont le protocole IP (Internet Protocol) est l'emblème. Il faudra toutefois attendre 1989 pour que Tim Berners-Lee invente le protocole HTTP et propose des

liens hypertextes avec le langage HTML pour que le grand public commence à se passionner pour le Word-Wide Web.

Depuis lors, le réseau IP n'a cessé de croître et d'obtenir les faveurs des acteurs des télécommunications. Avec les réseaux IP, la téléphonie connaît un nouvel élan. Elle se place à la jonction du monde des télécommunications et de celui des réseaux informatiques. Les professionnels ont rapidement compris l'intérêt d'une convergence vers un réseau entièrement IP. De son côté, le grand public se passionne pour des programmes tels que Skype, qui allient simplicité et performance, à des tarifs ultra-compétitifs.

Plus qu'un nouveau support de l'information, c'est un nouveau mode de communication qui est inventé avec la téléphonie sur IP. Les fonctionnalités étant accrues, une communication ne se limite plus qu'à la parole téléphonique, mais peut s'enrichir de multiples facettes, qui facilitent son usage, comme la vidéo associée à la parole téléphonique ou le service de présence des softphones, qui indique en temps réel la disponibilité de ses contacts.

Cet enrichissement s'accompagne de performances souvent supérieures à celles du traditionnel réseau RTC. La qualité d'une communication de ToIP est parfois tellement bonne qu'il est impossible de discerner si un correspondant est proche ou à l'autre bout du monde. Peu à peu, les habitudes comportementales des consommateurs sont modifiées. À des coûts très raisonnables et avec une telle commodité d'utilisation, les distances sont abolies, l'interactivité est fidèle, et les communications téléphoniques deviennent tout à la fois plus longues, plus conviviales et plus productives.

L'émergence de la ToIP se poursuit inexorablement depuis plusieurs années. Que l'on soit un particulier ou un professionnel, elle s'impose parallèlement sur différents axes. Pour un utilisateur équipé d'un ordinateur, les solutions de ToIP de type Skype sont nombreuses. Si l'usage d'un ordinateur rebute, les FAI proposent des solutions packagées dans leur offre Internet de base. Dans ce modèle, la ToIP tend à se substituer à la téléphonie fixe standard. Mais elle va aussitôt plus loin en introduisant progressivement sur le marché de la téléphonie sans fil, avec les technologies IP sans fil adéquates, comme Wi-Fi ou WiMax. Lorsque l'utilisateur n'a pas accès à un réseau IP, des terminaux hybrides lui permettent de basculer d'un réseau IP vers le réseau téléphonique classique. En quelque sorte, la transition vers un réseau entièrement IP se fait en douceur.

Les contraintes de cette nouvelle technologie n'en sont pas moins nombreuses, de même que les verrous à lever, en termes de disponibilité, de qualité de service, de sécurité et de mobilité. Ces contraintes sont à évaluer différemment selon le type de communication considéré. Un service de téléphonie ne peut s'accommoder d'une piètre qualité d'écoute sous peine d'être inutilisé. Il nécessite des ressources optimales. Le contrôle et la maîtrise des communications téléphoniques sur IP sont donc des enjeux colossaux pour favoriser l'essor de cette technologie.

## Objectifs de l'ouvrage

L'objectif de ce livre est de faire comprendre, par la théorie et par la pratique, pourquoi la téléphonie sur IP peut être considérée aujourd'hui comme mature et en quoi elle peut avantageusement remplacer la téléphonie standard de type RTC. Cela n'implique pas que les services exploitant cette technologie soient toujours à la hauteur des attentes des utilisateurs. Simplement, les protocoles dédiés à la gestion des flux multimédias sont disponibles et éprouvés pour satisfaire toutes ces exigences.

Toutes les conditions sont réunies pour valoriser ce potentiel et faire de la ToIP une technologie dominante, en phase avec les besoins de tout type.

Cet ouvrage s'adresse à un large public, aux professionnels comme aux particuliers. Il peut être lu et compris par toutes les personnes qui désirent découvrir ou approfondir les vastes possibilités qu'offre la ToIP.

Certains chapitres visent davantage des débutants, d'autres des étudiants, d'autres encore des professionnels du domaine. De nombreux chapitres sont indépendants et peuvent être lus de façon non linéaire, sans nécessiter de connaissances préalables, tandis que d'autres requièrent des bases plus techniques, que l'ouvrage apporte de façon progressive.

## Organisation de l'ouvrage

Ce livre se compose de deux grandes parties et d'une conclusion.

La première partie est dédiée aux notions fondamentales de la ToIP. Elle expose ses fondements théoriques et couvre un vaste état de l'art des normalisations adoptées pour le contrôle et la gestion du multimédia en général et de la voix sur IP en particulier. Elle détaille l'ensemble des spécificités des flux de téléphonie sur IP et s'attarde sur les architectures déployées ainsi que sur la manière dont les communications sont établies entre les interlocuteurs.

Après une présentation de la problématique et des contraintes de la ToIP (chapitres 1 et 2), les trois principaux protocoles de signalisation, que sont H.323, SIP et MGCP, sont détaillés (chapitres 3, 4 et 5). Nous évoquons ensuite la qualité de service (chapitre 6), l'architecture de sécurité (chapitre 7), autant de points cruciaux pour une application aussi sensible que la téléphonie.

La deuxième partie rassemble plusieurs composants disparates qui constituent un reflet de ce que recouvre aujourd'hui la ToIP dans la pratique. Les logiciels de téléphonie sur IP (ou softphones) en sont une composante importante, que nous présentons de manière générale (chapitre 8), avant de mettre l'accent sur les plus connus, tels que Skype (chapitre 9), WLM et Yahoo! Messenger (chapitre 10) puis Jabber et Google Talk (chapitre 11). Bien que certains d'entre eux n'ambitionnent pas directement de traiter de la téléphonie, ils en sont les vecteurs, et nous verrons qu'ils ne respectent pas toujours les protocoles de la standardisation.

Asterisk constitue également une tendance marquée de ces dernières années. Ce logiciel impressionnant permet de réaliser à moindre coût, dans un cadre industriel comme domestique, un commutateur téléphonique, ou PBX, avec une gamme de services associés, tels que la redirection d'appel, le répondeur téléphonique ou la conférence audio. Il initie une petite révolution dans le monde des télécoms, en remplaçant un modèle opaque par un modèle ouvert, ce qui permet une large démocratisation des techniques mises en œuvre. Toutefois, ses bases demeurent encore délicates à comprendre. C'est pourquoi nous consacrons un long développement (chapitre 12) sur cette thématique afin de fournir au lecteur les éléments clés du fonctionnement d'Asterisk.

Nous détaillons ensuite les offres de ToIP des FAI (chapitre 13) et présentons les techniques utilisées pour traverser les pare-feu et les NAT (chapitre 14), un sujet important compte tenu du fait que le NAT reste très employé, chez les particuliers comme chez les professionnels.

La troisième partie de l'ouvrage revient sur les cinq questions clés à se poser avant de passer à la téléphonie sur IP et offre, en conclusion, une vision des futurs développements attendus (chapitre 15). En particulier, et c'est la grande nouveauté de cette deuxième édition, l'IMS est longuement développé en fin d'ouvrage (chapitre 16).

Nous espérons que cette deuxième édition, plus riche en contenu que la précédente et actualisée, reflète les principales problématiques, technologies et tendances, actuelles et à venir, de la téléphonie sur IP, et permette de mieux comprendre les caractéristiques et les enjeux afin d'en être, comme utilisateurs avertis ou fournisseurs de services, les acteurs de demain.

# Table des matières

---

<b>Avant-propos</b> .....	V
<b>Objectifs de l'ouvrage</b> .....	VII
<b>Organisation de l'ouvrage</b> .....	VII

## **PARTIE I**

---

### **Théorie de la ToIP**

#### CHAPITRE 1

<b>Problématiques de la ToIP</b> .....	3
<b>La téléphonie par circuit et par paquets</b> .....	4
<b>La problématique de base de la téléphonie</b> .....	7
<b>Comparaison avec la téléphonie classique</b> .....	8
<b>Avantages de la ToIP</b> .....	10
<b>Les solutions de ToIP</b> .....	13
<b>Questions posées par la mise en place de la ToIP en entreprise</b> .....	18
<b>Conclusion</b> .....	20

#### CHAPITRE 2

<b>Contraintes de la ToIP</b> .....	23
<b>Les contraintes temporelles</b> .....	23
<b>Le processus de resynchronisation de la parole téléphonique</b> .....	26

<b>La téléphonie numérique</b> .....	27
L'échantillonnage .....	28
Techniques de codage .....	30
<b>Les codeurs audio</b> .....	32
<b>Qualité de service de la ToIP</b> .....	35
<b>Caractéristiques du débit</b> .....	37
<b>Le contrôle dans la ToIP</b> .....	40
<b>Conclusion</b> .....	42

### CHAPITRE 3

<b>La signalisation H.323</b> .....	43
<b>Protocoles et normalisation</b> .....	45
La normalisation UIT .....	46
Normes d'interopérabilité .....	47
Les six versions de H.323 .....	48
<b>Architecture et fonctionnalités du protocole H.323</b> .....	53
Les quatre entités d'une architecture H.323 .....	53
Le terminal H.323, équipement des interlocuteurs .....	55
Le gatekeeper, point de contrôle et de gestion .....	57
La passerelle, pour joindre les réseaux ne fonctionnant pas en mode paquet	60
La MCU et les conférences .....	64
<b>Les messages H.323</b> .....	68
Le protocole H.225.0, signalisation d'appel et d'enregistrement .....	69
Le protocole H.245, la signalisation de contrôle de connexion .....	75
Les autres protocoles .....	78
Exemple de scénario d'une communication complète .....	79
<b>Fonctionnalités avancées de H.323</b> .....	81
La procédure Early H.245 .....	81
La procédure FastConnect .....	82
La procédure H.245 tunneling .....	82
La sécurité .....	82
Gatekeeper alternatif et gatekeeper affecté .....	83
<b>Conclusion</b> .....	84

## CHAPITRE 4

<b>Le protocole SIP</b> .....	85
<b>La standardisation SIP (Session Initiation Protocol)</b> .....	85
Historique .....	86
Compatibilité .....	86
Modularité .....	87
Simplicité .....	88
<b>Architecture de SIP</b> .....	89
Se connecter à des réseaux non-IP .....	94
<b>L'adressage SIP</b> .....	94
URI (Universal Ressource Identifier) .....	95
Format des adresses SIP .....	96
Localisation et résolution d'une adresse SIP .....	98
<b>Les messages SIP</b> .....	100
Notion de transaction .....	100
Paramètres généraux pour les requêtes et les réponses .....	101
Le champ VIA pour détecter les boucles lors du routage .....	103
Différence entre Call-Id et CSeq .....	104
Abréviation des en-têtes de messages .....	105
Corps d'un message .....	106
Les requêtes SIP .....	112
Méthodes d'extension du protocole SIP .....	115
Les réponses SIP .....	116
<b>Scénarios de communication</b> .....	120
1. Initialisation d'une communication directe .....	120
2. Enregistrement d'un terminal .....	122
3. Initialisation d'une communication SIP avec un serveur proxy .....	122
4. Localisation par un serveur de redirection et initialisation d'appel directe .....	125
5. Modification d'une communication SIP .....	126
6. Terminaison d'une communication SIP .....	127
<b>Conclusion</b> .....	128

## CHAPITRE 5

<b>Le protocole MGCP</b> .....	129
<b>Historique</b> .....	130
H.248/MeGaCoP .....	131

<b>Architecture et fonctionnement</b> .....	132
Le Call Agent .....	133
Les passerelles multimédias .....	134
Raisons d'être d'un nouveau protocole .....	136
Exemple d'utilisation de MGCP chez les FAI .....	137
Avantages et inconvénients de MGCP .....	138
<b>Principes d'établissement d'une communication</b> .....	139
<b>Les messages MGCP</b> .....	142
Adressage des endpoints .....	142
Identifiant de transaction .....	144
Paramètres généraux pour les requêtes et les réponses .....	145
La ligne d'état MGCP .....	147
Les requêtes .....	147
Les réponses MGCP .....	151
<b>Conclusion</b> .....	155
CHAPITRE 6	
<b>La qualité de service</b> .....	157
<b>Le contrôle et les protocoles de transport</b> .....	157
TCP et le transport de données multimédias temps réel .....	158
UDP et le transport de données multimédias temps réel .....	160
En résumé .....	160
<b>Les protocoles RTP et RTCP</b> .....	161
RTP (Real-time Transport Protocol) .....	161
RTCP (Real-time Transport Control Protocol) .....	166
RTP/RTCP et la qualité de service .....	167
<b>Les contrôles au niveau réseau</b> .....	167
IntServ (Integrated Services) .....	168
DiffServ (Differentiated Services) .....	170
<b>L'ingénierie de trafic</b> .....	177
<b>Conclusion</b> .....	181
CHAPITRE 7	
<b>Architectures et sécurité</b> .....	183
<b>La téléphonie sur Ethernet</b> .....	183
L'intégration voix-données .....	183

<b>La téléphonie sur ATM</b> .....	187
AAL2 .....	187
Les microtrames AAL2 .....	188
<b>La téléphonie sur le relais de trames</b> .....	189
Intégration de la téléphonie dans le relais de trame .....	190
<b>La téléphonie sur réseaux sans fil</b> .....	195
Contraintes de la ToIP sans fil .....	195
La qualité de service .....	197
En résumé .....	205
<b>La téléphonie sur WiMax</b> .....	205
WiMax fixe .....	205
WiMax-Mobile .....	207
Classes de services WiMax pour la ToIP .....	208
<b>La sécurité</b> .....	209
Les attaques .....	210
Les sécurités à mettre en place .....	213
Les infrastructures de sécurité .....	214
La sécurité dans la téléphonie par Wi-Fi .....	215
<b>Conclusion</b> .....	216

## PARTIE II

### Pratique de la ToIP

#### CHAPITRE 8

<b>La ToIP sur softphone</b> .....	219
<b>Introduction aux softphones</b> .....	220
<b>Les services proposés</b> .....	220
La téléphonie .....	221
Liste de contacts, présence et disponibilité .....	223
Messagerie instantanée .....	223
Vidéo et transfert de fichiers .....	224
<b>Les softphones en entreprise</b> .....	225
<b>Les autres softphones</b> .....	225
WengoPhone .....	225
Téléphoner gratuitement d'un PC vers un téléphone fixe .....	228

Les clients de messagerie Web .....	229
La téléphonie Free .....	229
<b>Conclusion</b> .....	235

## CHAPITRE 9

<b>Skype</b> .....	237
<b>Architecture de Skype</b> .....	238
Limiter les ressources .....	239
Traverser les pare-feu .....	239
<b>Les offres Skype</b> .....	240
Partenariats technologiques et commerciaux .....	241
<b>La sécurité</b> .....	242
<b>Utiliser Skype</b> .....	243
Prérequis .....	243
Appeler .....	244
Outils .....	246
<b>Aller plus loin avec Skype</b> .....	247
Ouvrir plusieurs instances de Skype .....	248
Options en ligne de commande .....	252
Commandes textuelles .....	253
Intégrer Skype dans ses pages Web et ses e-mails .....	254
Recommandations et résolution de problèmes .....	256
<b>Conclusion</b> .....	257

## CHAPITRE 10

<b>Windows Live Messenger et Yahoo! Messenger</b> .....	259
<b>Windows Live Messenger</b> .....	259
La gamme de services unifiés Live .....	260
WLM (Windows Live Messenger) .....	261
Utiliser WLM .....	262
Aller plus loin avec WLM .....	264
<b>Yahoo! Messenger</b> .....	271
Utilisation .....	271

<b>Le partenariat Microsoft-Yahoo!</b> .....	277
<b>Conclusion</b> .....	279
CHAPITRE 11	
<b>Jabber et Google Talk</b> .....	281
<b>Jabber</b> .....	281
Architecture de Jabber .....	282
XMPP (eXtensible Messaging and Presence Protocol) .....	284
XEP (XMPP Enhancement Proposals) .....	285
Utilisation .....	287
<b>Google Talk</b> .....	295
Une offre à trois volets .....	296
Utilisation .....	297
<b>Conclusion</b> .....	303
CHAPITRE 12	
<b>Asterisk</b> .....	305
<b>Introduction aux PBX</b> .....	305
<b>Présentation d'Asterisk</b> .....	307
Fonctionnalités .....	308
Compatibilité .....	308
Cible et usage .....	309
<b>Installation de base</b> .....	311
Mise en œuvre de la plate-forme .....	312
Lancement du serveur et exploitation .....	315
<b>Configuration</b> .....	318
Les quatre catégories d'éléments d'Asterisk .....	318
Organisation des fichiers (fichier asterisk.conf) .....	319
Première étape de configuration : Description des utilisateurs et des terminaux (fichiers sip.conf, iax.conf, mgcp.conf, h323.conf, skinny.conf) .....	321
Deuxième étape de configuration : le plan de numérotation (fichier extensions.conf) .....	324

<b>Tester la configuration d'un client</b> .....	336
<b>Optimiser les traitements</b> .....	338
La directive d'inclusion .....	339
Logique de programmation .....	339
Optimisation du routage avec les contextes .....	342
Et la vidéo ? .....	342
Ajouter des sons .....	343
Problèmes éventuels avec les modules .....	345
<b>Ajouter de nouveaux services</b> .....	346
Standard vocal automatique (IVR) .....	346
Conférence .....	347
Le service de messagerie audio (fichier voicemail.conf) .....	349
<b>Aller plus loin avec Asterisk</b> .....	352
Connecter Asterisk à un fournisseur SIP .....	352
AGI (Asterisk Gateway Interface) .....	355
Trixbbox .....	355
Communiquer avec le protocole IAX .....	356
Asterisk sous Windows .....	356
La concurrence .....	357
<b>Conclusion</b> .....	358

## CHAPITRE 13

<b>La téléphonie chez les fournisseurs d'accès</b> .....	359
<b>Les accès xDSL</b> .....	359
Le modem xDSL .....	360
Ethernet dans le premier mile .....	362
Les protocoles de l'ADSL .....	363
Le protocole L2TP .....	365
Les modems VDSL .....	366
La parole et la vidéo sur xDSL .....	366
<b>La téléphonie sur CATV</b> .....	367
<b>La téléphonie sur fibre optique</b> .....	370
<b>La téléphonie sur Quadruple-Play</b> .....	372
<b>Conclusion</b> .....	373

## CHAPITRE 14

<b>Filtrage des flux de ToIP</b> .....	375
<b>Le mécanisme de NAT (Network Address Translation)</b> .....	376
Adresses privées et adresses publiques .....	376
Partager une adresse IP privée .....	377
Avantages du NAT .....	379
<b>Les trois catégories de NAT</b> .....	381
Le NAT statique .....	381
Le NAT dynamique .....	381
Le NAPT .....	383
<b>Les problèmes engendrés par le NAT</b> .....	384
Les protocoles sensibles au NAT .....	384
Recevoir une connexion derrière un NAPT .....	384
La sécurité avec le NAT .....	385
En résumé. ....	385
<b>Le passage des pare-feu</b> .....	386
<b>Méthodes de résolution de la translation d'adresse pour les flux multimédias</b> .....	388
Filtrage applicatif des données. ....	388
Tunneliser les applications .....	389
La gestion du NAT par le client .....	391
En résumé. ....	394
<b>Conclusion</b> .....	395

## PARTIE III

## Conclusion

## CHAPITRE 15

<b>Les cinq problèmes clés de la ToIP</b> .....	399
<b>La sécurité</b> .....	400
L'authentification .....	400
Confidentialité et intégrité .....	406

<b>La disponibilité</b> .....	406
<b>La gestion</b> .....	408
<b>Le contrôle</b> .....	410
<b>La qualité de service</b> .....	411
<b>GAN</b> .....	412
<b>Perpectives</b> .....	413
<b>Conclusion</b> .....	414

## CHAPITRE 16

<b>L'architecture IMS</b> .....	415
<b>Motivations et objectifs de l'IMS</b> .....	416
Le point de vue opérateur .....	417
Le point de vue utilisateur .....	418
<b>Historique de l'IMS</b> .....	419
<b>Introduction à l'IMS</b> .....	420
Protocoles utilisés dans l'IMS .....	420
Notions de réseau visité et de roaming .....	421
<b>Le modèle à quatre couches de l'architecture IMS</b> .....	422
La couche de transport .....	424
La couche de contrôle .....	424
HSS (Home Subscriber Server) : la base de données des utilisateurs .....	425
CSCF (Call State Control Function) : les trois gestionnaires d'appels .....	425
MRF (Multimedia Resource Function) : pour la gestion des conférences .....	429
BGCF (Breakout Gateway Control Function) : pour la liaison avec le réseau RTC .....	430
IMS-MGW, MGCF et SGW : pour l'interfonctionnement avec le réseau RTC .....	430
TrGW et IMS-ALG : pour le support transparent des adresses IPv4 .....	432
<b>Les serveurs d'applications</b> .....	434
SIP AS (SIP Application Server) .....	434
IM-SSF (IP Multimedia-Service Switching Function) .....	434
OSA-SCS (Open Service Access-Service Capability Server) .....	435
<b>Les identités IMS</b> .....	435
Gestionnaire d'identités .....	436

---

<b>Les extensions du protocole SIP pour IMS</b> .....	437
<b>Les commandes Diameter</b> .....	439
<b>Communications avec IMS</b> .....	440
Enregistrement d'un terminal dans le réseau .....	440
Mise en relation de deux utilisateurs .....	444
<b>Conclusion</b> .....	445

## PARTIE IV

---

### Annexe

<b>Références</b> .....	449
<b>Liens web</b> .....	452
Sites de vulgarisation de la ToIP .....	452
Protocoles de ToIP .....	453
Softphones et dérivés .....	454
PBX Asterisk .....	456
Salons sur la VoIP en France .....	457
<b>Index</b> .....	459



# Partie I

## Théorie de la ToIP

L'objectif de cette partie est de présenter le socle théorique sur lequel repose la téléphonie sur IP.

Les deux premiers chapitres évoquent l'intérêt et les difficultés soulevées par la ToIP.

Le chapitre 1 introduit la problématique de ToIP, c'est-à-dire le transport de la parole dans le cadre de la téléphonie, qui implique un processus temps réel. Nous expliquons quels sont les enjeux posés par la voix sur IP, ce à quoi elle peut se substituer et quels bénéfices il est possible d'en tirer.

Le chapitre 2 détaille les contraintes imposées par la ToIP. Pour être pleinement fonctionnel, un service de téléphonie sur IP a des exigences fortes vis-à-vis des terminaux utilisés comme du réseau exploité. Nous verrons comment caractériser ces contraintes et comment les mesurer.

Les trois chapitres suivants présentent les protocoles de signalisation standardisés de la ToIP.

Le chapitre 3 décrit le protocole H.323. Celui-ci a longtemps fait office de référence en matière de protocole de signalisation pour le multimédia en général, et pour la téléphonie sur IP en particulier. Porteur d'un fort héritage du monde des télécoms, il s'est imposé sur le marché. Nous montrons sur quelle architecture il repose et comment s'effectuent les communications qu'il met en place.

Le chapitre 4 décrit le protocole SIP, concurrent du protocole H.323 et disposant d'atouts remarquables qui favorisent son émergence. Nous montrons pourquoi ce protocole est en passe de détrôner H.323, pourquoi il s'intègre mieux dans un réseau IP, comment il simplifie les communications et comment on l'utilise.

Le chapitre 5 décrit le protocole MGCP, un autre protocole de signalisation, qui est complémentaire de H.323 et SIP. Par sa vision d'opérateur et en offrant la faculté de concentrer l'intelligence du réseau au sein d'entités spécialisées, le protocole MGCP s'est imposé, tant et si bien que ses évolutions peinent à justifier leur utilité. Nous montrons en quoi le protocole MGCP allie simplicité et puissance et de quelle manière il fonctionne.

Les deux derniers chapitres de cette partie s'intéressent aux questions de qualité de service, d'architectures réseau et de sécurité.

Le chapitre 6 expose les différents protocoles et technologies permettant de mettre en place une gestion de la qualité de service pour la téléphonie sur IP. La qualité de service apportée aux flux de parole téléphonique est primordiale. Nous détaillons les protocoles à mettre en œuvre pour la maîtriser dans un réseau.

Le chapitre 7 fournit les principales caractéristiques des architectures réseau dédiées à la VoIP et présente quelques notions importantes relatives à la sécurité de la téléphonie sur IP. Les flux téléphoniques pouvant emprunter un réseau partagé par d'autres catégories de données, nous indiquons, selon les types de réseaux, quelles sont les adaptations à considérer et comment faire pour protéger les flux de parole téléphonique.

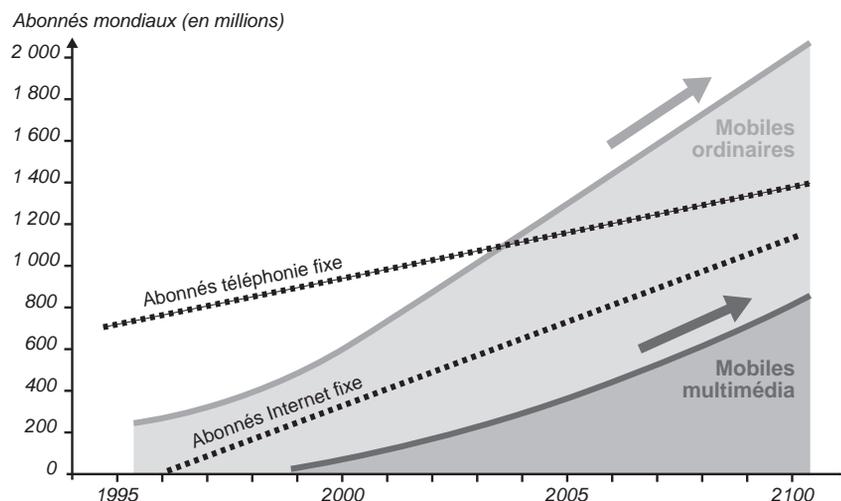
# Problématiques de la ToIP

La téléphonie est un des moyens de communication préférés des êtres humains, et le nombre de terminaux téléphoniques vendus dans le monde ne cesse d'augmenter.

La figure 1.1 illustre le nombre de terminaux pouvant servir de terminal téléphonique. On peut noter que le nombre des terminaux mobiles dépasse largement celui des terminaux fixes. On peut également noter que le nombre de terminaux fixes continue d'augmenter, quoique nettement moins que celui des mobiles. La figure indique en outre le nombre de terminaux, fixes ou mobiles, intégrant des fonctions multimédias. Toutes ces courbes révèlent la croissance globale de la téléphonie.

**Figure 1.1**

*Abonnés aux réseaux de télécommunications (source UMTS Forum)*



La téléphonie a été une véritable poule aux œufs d'or pour les opérateurs, qui ont longtemps maintenu leurs tarifs à des niveaux assez élevés, alors même que leurs infrastructures étaient largement amorties.

Aujourd'hui, la position de ces opérateurs est rapidement menacée par l'arrivée massive de la téléphonie sur IP, dont la tarification tend vers la gratuité. En France, fin 2006, la téléphonie sur IP représente déjà près de 50 % du marché de la téléphonie. Aux environs de 2009, on estime que près de 100 % du transport de la parole s'effectuera par l'intermédiaire de paquets IP.

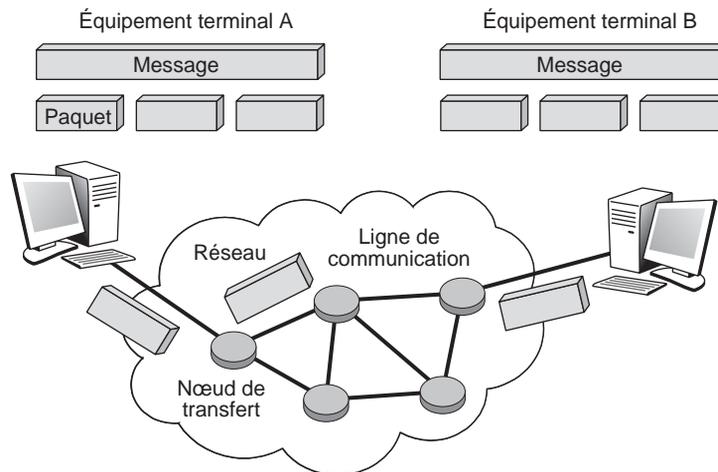
Nous donnerons au cours des sections qui suivent quelques indications sur les problématiques techniques de la téléphonie par paquets. Nous examinerons ensuite les premières grandes caractéristiques de cette technologie et terminerons en présentant les différents environnements de la téléphonie IP : grand public, opérateurs et entreprises.

## La téléphonie par circuit et par paquets

Dans la communication à transfert de paquets, toutes les informations à transporter sont découpées en paquets pour être acheminées d'une extrémité à une autre du réseau. Cette technique est illustrée à la figure 1.2.

L'équipement terminal A souhaite envoyer un message à B. Le message est découpé en trois paquets, qui sont émis de l'équipement terminal vers le premier nœud du réseau, lequel les envoie à un deuxième nœud, et ainsi de suite, jusqu'à ce qu'ils arrivent à l'équipement terminal B. Dans l'équipement terminal les paquets rassemblés reconstituent le message de départ.

**Figure 1.2**  
*La technique de transfert de paquets*



Le paquet peut en fait provenir de différents médias. Sur la figure 1.2, nous supposons que la source est un message composé de données, comme une page de texte préparée au

moyen d'un traitement de texte. Le terme message est cependant beaucoup plus vaste et recoupe toutes les formes sous lesquelles de l'information peut se présenter. Cela va d'une page Web à un flot de parole téléphonique représentant une conversation.

Dans la parole téléphonique, l'information est regroupée pour être placée dans un paquet, comme illustré à la figure 1.3. Le combiné téléphonique produit des octets, provenant de la numérisation de la parole, c'est-à-dire le passage d'un signal analogique à un signal sous forme de 0 et de 1, qui remplissent petit à petit le paquet. Dès que celui-ci est plein, il est émis vers le destinataire. Une fois le paquet arrivé à la station terminale, le processus inverse s'effectue, restituant les éléments binaires régulièrement à partir du paquet pour reconstituer la parole téléphonique.

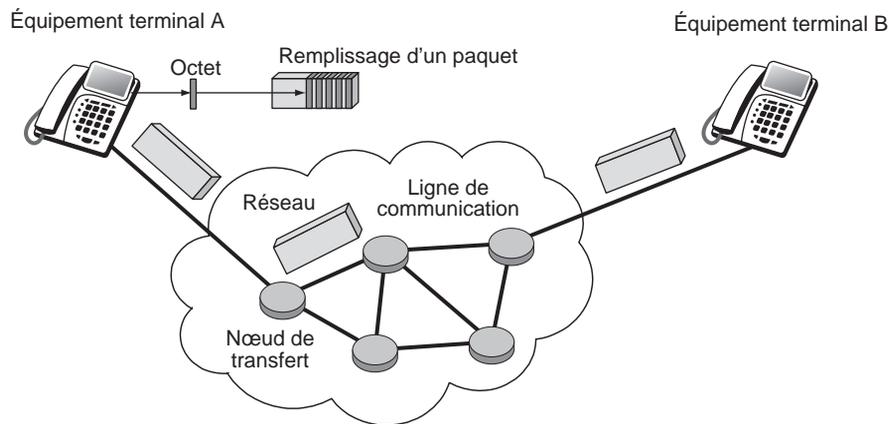


Figure 1.3

*Un flot de paquets téléphoniques*

Le réseau de transfert est lui-même composé de nœuds, appelés nœuds de transfert, reliés entre eux par des lignes de communication, sur lesquelles sont émis les éléments binaires constituant les paquets. Le travail d'un nœud de transfert consiste à recevoir des paquets et à déterminer vers quel nœud suivant ces derniers doivent être acheminés.

Le paquet forme donc l'entité de base, transférée de nœud en nœud jusqu'à atteindre le récepteur. Ce paquet est regroupé avec d'autres paquets pour reconstituer l'information transmise. L'action consistant à remplir un paquet avec des éléments binaires en général regroupés par octet s'appelle la mise en paquet, ou encore la paquetsation, et l'action inverse, consistant à retrouver un flot d'octets à partir d'un paquet, la dépaquetsation.

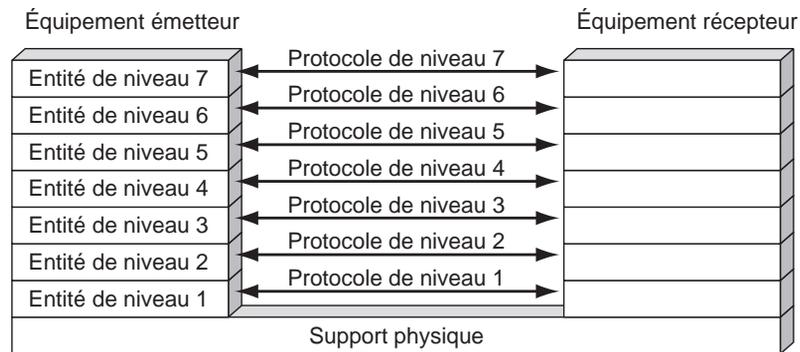
L'architecture d'un réseau est définie principalement par la façon dont les paquets sont transmis d'une extrémité du réseau à une autre. De nombreuses variantes existent pour cela, comme celle consistant à faire passer les paquets toujours par la même route ou, au contraire, à les faire transiter par des routes distinctes de façon à minimiser les délais de traversée.

Pour identifier correctement toutes les composantes nécessaires à la bonne marche d'un réseau à transfert de paquets, un modèle de référence a été mis au point. Ce modèle définit une partition de l'architecture en sept niveaux, prenant en charge l'ensemble des fonctions nécessaires au transport et à la gestion des paquets. Ces sept couches de protocoles ne sont pas toutes indispensables, notamment aux réseaux sans visée généraliste. Chaque niveau, ou couche, offre un service au niveau supérieur et utilise les services du niveau inférieur.

Pour offrir ces services, les couches disposent de protocoles qui appliquent les algorithmes nécessaires à la bonne marche des opérations, comme l'illustre la figure 1.4.

Figure 1.4

Architecture protocolaire d'un réseau à sept niveaux



Nous supposons ici que l'architecture protocolaire est découpée en sept niveaux, ce qui est le cas du modèle de référence. Nous ne décrivons que succinctement les couches basses qui nous intéressent.

Le niveau 3 représente le niveau paquet : il définit les algorithmes nécessaires pour que les entités de niveau 3, les paquets, soient acheminées correctement de l'émetteur au récepteur. Le niveau 7 correspond au niveau application. Le rôle du protocole de niveau 7 est de transporter correctement l'entité de niveau 7, le message utilisateur, de l'équipement émetteur à l'équipement récepteur.

Le niveau 2, ou niveau trame, permet de transférer le paquet sur une ligne physique. En effet, un paquet ne contenant pas de délimiteur, le récepteur ne peut en déterminer la fin ni identifier le commencement du paquet suivant. Pour transporter un paquet, il faut donc le mettre dans une trame, qui, elle, comporte des délimiteurs. On peut aussi encapsuler un paquet dans un autre paquet, lui-même encapsulé dans une trame.

Il est important de distinguer les mots *paquet* et *trame* de façon à bien différencier les entités qui ne sont pas transportables directement, comme le paquet IP, et les entités transportables directement par la couche physique, comme les trames Ethernet ou ATM.

Dans la téléphonie sur IP, une suite d'octets de téléphonie est encapsulée dans un paquet IP de niveau 3, lui-même encapsulé dans une trame véhiculée sur le support physique.

Cependant, comme la téléphonie est une application temps réel, les paquets ne peuvent attendre trop longtemps dans le réseau. Il faut donc introduire des contrôles afin de permettre une traversée rapide du réseau. Nous détaillons ces contrôles au chapitre 6.

## La problématique de base de la téléphonie

La voix sur IP adresse deux types d'applications : celles qui, comme la téléphonie, mettent en jeu une interaction humaine, laquelle implique un temps de transit très court, et celles qui transportent des paroles unidirectionnelles, qui n'exigent pas de temps réel. Cette dernière catégorie rassemble essentiellement des transferts de fichiers contenant de la parole. Dans ce livre, nous nous intéressons uniquement à la parole téléphonique.

La téléphonie transportée par paquets, et plus particulièrement par paquet IP, permet d'intégrer dans un même réseau les services de données et la téléphonie. Les entreprises sont de plus en plus nombreuses à intégrer leur environnement téléphonique dans leur réseau à transfert de paquets. Les avantages de cette intégration sont, bien sûr, la baisse des frais de communication, mais aussi la simplification de la maintenance de leurs réseaux, qui passent de deux (téléphonie et données) à un seul (données).

La difficulté de la téléphonie par paquets réside dans la très forte contrainte temporelle due à l'interaction entre individus. Le temps de latence doit être inférieur à 300 ms si l'on veut garder une interaction humaine acceptable. Si l'on souhaite une bonne qualité de la conversation, la latence ne doit pas dépasser 150 ms.

Un cas encore plus complexe se produit lorsqu'il y a un écho, c'est-à-dire un signal qui revient dans l'oreille de l'émetteur. L'écho se produit lorsque le signal rencontre un obstacle, comme l'arrivée sur le combiné téléphonique. L'écho qui repart en sens inverse est numérisé par un codec (codeur-décodeur) et traverse sans problème un réseau numérique. La valeur normalisée de la latence de l'écho étant de 56 ms, pour que l'écho ne soit pas gênant à l'oreille, il faut que le temps aller ne dépasse pas 28 ms, en supposant un réseau symétrique prenant le même temps de transit à l'aller qu'au retour. Il faut donc que, dans les équipements terminaux, les logiciels extrémité soient capables de gérer les retards et de resynchroniser les octets qui arrivent. Les équipements modernes, comme les terminaux GSM, possèdent des supprimeurs d'écho évitant cette contrainte temporelle forte.

Une autre caractéristique essentielle de la téléphonie provient du besoin d'avertir par une sonnerie la personne qui est appelée. La communication téléphonique est pour cela décomposée en deux phases : une première permettant d'avertir le destinataire, et une seconde correspondant au transport de la parole proprement dite. Il existe en réalité une troisième phase, qui consiste en la finalisation de la communication lorsqu'un des deux terminaux raccroche. Cette phase utilise le même type de protocole que la première : un protocole de signalisation.

## Comparaison avec la téléphonie classique

La téléphonie classique, dite par circuit, présente les mêmes contraintes temporelles que la téléphonie par paquet. Le temps de transit doit être limité pour satisfaire le besoin d'interactivité entre individus.

La limitation du temps de transit entre l'émetteur et le récepteur est relativement simple à réaliser dans une technologie circuit. Les ressources étant réservées, la voie est toujours dégagée sur le circuit, et les ressources appartiennent uniquement aux signaux qui transitent entre l'émetteur et le récepteur. En revanche, dans un transfert de paquets, aucune ressource n'est réservée, et il est impossible de savoir quel sera le temps d'attente des paquets dans les nœuds de transfert.

Dans la première génération de téléphonie, les signaux étaient analogiques. Ils parcouraient le circuit sous la même forme que le son sortant de la bouche et n'utilisaient que 3 200 Hz de bande passante. Ils sont ensuite devenus numériques.

Dans la téléphonie traditionnelle numérique, le signal analogique est numérisé grâce à un codeur-décodeur, appelé codec. Le codec transforme le signal analogique en une suite de 0 et de 1. Le temps de transit est du même ordre de grandeur que le transfert du signal analogique, car le signal ne s'arrête nulle part. La seule perte de temps pourrait provenir du codec, mais ces équipements très rapides ne modifient pas fondamentalement le temps de transit. En revanche, dans un réseau à transfert de paquets, de nombreux obstacles se dressent tout au long du cheminement des informations binaires.

L'élément le plus contraignant de l'application de téléphonie par paquet reste le délai pour aller d'une extrémité à l'autre, puisqu'il faut traverser les deux terminaux, émetteur et récepteur, de type PC par exemple, ainsi que les modems, les réseaux d'accès, les passerelles, les routeurs, etc.

On peut considérer que le temps de traversée d'un PC et de son codec demande quelques millisecondes, la paquetsation de 5 à 16 millisecondes, la traversée d'un modem quelques millisecondes également, celui d'un routeur ou d'une passerelle de l'ordre de la milliseconde (s'il n'y a aucun paquet en attente) et celui d'un réseau IP quelques dizaines de millisecondes.

L'addition de ces temps montre que la limite maximale de 300 ms permettant l'interactivité est rapidement atteinte. La figure 1.5 illustre ce processus.

Le déroulement d'une communication téléphonique sur IP parcourt les cinq grandes étapes suivantes :

1. **Mise en place la communication.** Une signalisation démarre la session. Le premier élément à considérer est la localisation du récepteur (*User Location*). Elle s'effectue par une conversion de l'adresse du destinataire (adresse IP ou adresse téléphonique classique) en une adresse IP d'une machine qui puisse joindre le destinataire (qui peut être le destinataire lui-même). Le récepteur peut être un combiné téléphonique classique sur un réseau d'opérateur télécoms ou une station de travail (lorsque la communication s'effectue d'un combiné téléphonique vers un PC). Le protocole DHCP (Dynamic Host Configuration Protocol) et les passerelles spécialisées (*gate-keeper*) sont employés à cette fin.

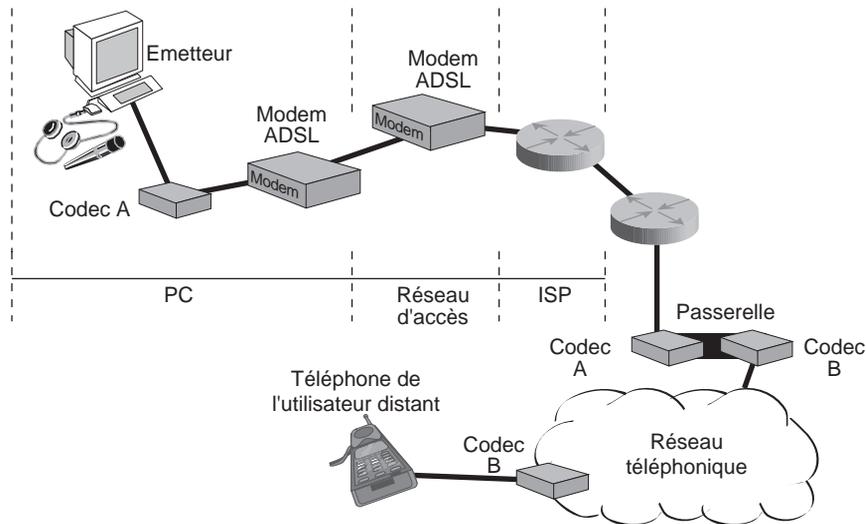


Figure 1.5

*Équipements à traverser par une communication téléphonique sur IP*

2. **Établissement de la communication.** Cela passe par une acceptation du terminal destinataire, que ce dernier soit un téléphone, une boîte vocale ou un serveur Web. Plusieurs protocoles de signalisation sont utilisés pour cela, en particulier le protocole SIP (Session Initiation Protocol) de l'IETF. Comme son nom l'indique, SIP est utilisé pour initialiser la session. Une requête SIP contient un ensemble d'en-têtes, qui décrivent l'appel, suivis du corps du message, qui contient la description de la demande de session. SIP est un protocole client-serveur, qui utilise la syntaxe et la sémantique de HTTP. Le serveur gère la demande et fournit une réponse au client.

Trois types de serveurs gèrent différents éléments : un serveur d'enregistrement (Registration Server), un serveur relais (Proxy Server) et un serveur de redirection (Redirect Server). Ces serveurs travaillent à trouver la route : le serveur proxy détermine le prochain serveur (Next-Hop Server), qui, à son tour, trouve le suivant, et ainsi de suite. Des champs supplémentaires de l'en-tête gèrent des options, comme le transfert d'appel ou la gestion des conférences téléphoniques.

3. **Transport de l'information téléphonique.** Le protocole RTP (Real-time Transport Protocol) prend le relais pour transporter l'information téléphonique proprement dite. Son rôle est d'organiser les paquets à l'entrée du réseau et de les contrôler à la sortie de façon à reformer le flot avec ses caractéristiques de départ (vérification du synchronisme, des pertes, etc.). C'est un protocole de niveau transport, qui essaye de corriger les défauts apportés par le réseau.
4. **Changement de réseau.** Un autre lieu de transit important de la ToIP est constitué par les passerelles, qui permettent de passer d'un réseau à transfert de paquets à un

réseau à commutation de circuits, en prenant en charge les problèmes d'adressage, de signalisation et de transcodage que cela pose. Ces passerelles ne cessent de se multiplier entre FAI et opérateurs télécoms.

5. **Arrivée au destinataire.** De nouveau, le protocole SIP envoie une requête à la passerelle pour déterminer si elle est capable de réaliser la liaison circuit de façon à atteindre le destinataire. En théorie, chaque passerelle peut appeler n'importe quel numéro de téléphone. Cependant, pour réduire les coûts, mieux vaut choisir une passerelle locale, qui garantit que la partie du transport sur le réseau téléphonique classique est le moins cher possible.

Cet exemple classique illustre la relative complexité de la téléphonie sur IP. De nombreuses variantes existent, mais elles ne diffèrent que par les protocoles utilisés. À cette complexité s'ajoutent les problèmes liés à la traversée du réseau, qui doit garantir des temps de transit acceptables pour que l'application téléphonique puisse se dérouler dans de bonnes conditions.

## Avantages de la ToIP

La téléphonie n'a jamais été une application simple. Les contraintes temps réel et de synchronisation pèsent lourdement sur sa mise en œuvre, et la téléphonie par paquet ne fait que compliquer le transport.

Cependant, plusieurs raisons expliquent le succès de la téléphonie par paquet, et plus spécifiquement de la téléphonie sur IP :

- **Convergence.** Quel que soit le type de données véhiculées, le réseau est unique : les flux de voix, de vidéo, de textes et d'applicatifs transitent sur le même réseau. Les communications deviennent plus riches, et sans avoir besoin de multiplier les canaux de transport. Les utilisateurs peuvent, par exemple, envoyer un compte rendu d'activité en même temps qu'ils téléphonent à leur correspondant. Pour les utilisateurs, la convivialité est accrue. En entreprise, la productivité est améliorée. Pour les administrateurs, un seul réseau est à administrer, ce qui simplifie grandement la gestion.
- **Optimisation des ressources.** Le réseau IP utilisant un transfert de paquets, l'utilisation des ressources est optimisée en comparaison des solutions de type commutation de circuits. Dans le réseau RTC, qui est à commutation de circuits, des ressources sont dédiées pour toute la durée de la communication, qu'elles soient utilisées ou non. Or les très nombreux silences d'une conversation téléphonique rendent le dimensionnement du canal réservé systématiquement trop grand. Pour que la voix supporte simultanément la superposition des deux paroles correspondant aux deux intervenants d'une communication téléphonique (full-duplex), les réseaux RTC doivent allouer pour chaque intervenant des canaux différents, l'un en émission, l'autre en réception. Dans la pratique, lors d'une conversation téléphonique, une seule personne parle en même

temps. Les ressources sont donc globalement gaspillées. C'est pourquoi la réservation effectuée dans les réseaux RTC représente un coût nettement supérieur à celui des réseaux IP.

- **Coût de transport quasiment nul.** Grâce à l'intégration de la téléphonie parmi de nombreuses autres applications, le coût du transport devient pratiquement nul. Le réseau permettant d'effectuer le transport est le réseau cœur des opérateurs, celui qui effectue tous les transports de données. Ces opérateurs, qui étaient auparavant obligés de maintenir au moins deux réseaux, celui de téléphonie et celui de données, n'en ont plus qu'un seul à maintenir. L'intégration supplémentaire de la télévision dans le réseau de données fait également chuter les coûts de transport de cette application.
- **Services exclusifs.** Certains services sont propres aux réseaux IP. Par exemple, le service de présence, consistant à détecter si un utilisateur est connecté au réseau ou non, ne nécessite aucune réservation de ressources dans un réseau IP, à la différence du réseau RTC. De façon analogue, pour le nomadisme des utilisateurs, il est plus simple de passer, partout dans le monde, par le réseau IP plutôt que par le réseau RTC.
- **Disparition des commutateurs locaux.** Liée à la précédente, cette nouvelle donne résulte de la possibilité de gérer les téléphones depuis le réseau de l'opérateur (système Centrex). Des solutions intermédiaires, comme les PBX-IP, permettent de passer petit à petit des circuits numériques aux liaisons paquet IP.

La téléphonie devient ainsi une application du réseau IP comme une autre, si ce n'est qu'elle nécessite une qualité de service particulière. De ce fait, les modems ADSL qui amènent chez l'utilisateur la connectivité IP constituent la porte d'entrée de la téléphonie IP. Le modem l'intègre avec les applications de données (messagerie, transfert de fichiers, P2P), la télévision, la visiophonie, etc.

Début 2007, cette intégration n'était pas encore finalisée puisque la plupart des postes téléphoniques ne sont pas encore des postes IP capables d'émettre directement des paquets IP. Il faut un point de connexion spécifique sur le modem pour indiquer que le flux est une parole téléphonique.

De même, le flux de télévision se distingue des autres applications par un accès spécifique sur le modem. Cependant, dès que les téléphones et les télévisions seront IP, le réseau domestique ne distinguera plus ces applications particulières, et ce sera le modem qui, en filtrant les flux, découvrira les paquets de téléphonie et les paquets de télévision pour les traiter en conséquence.

Cette différenciation est illustrée aux figures 1.6 et 1.7. La première présente l'état actuel, où les flux de données, de vidéo et de téléphonie sont différenciés par la prise par laquelle ils transitent, et la seconde celui de demain, où tous les flux sont intégrés sur le réseau domestique et sont différenciés par le biais d'un filtre applicatif dans le modem ADSL.

Figure 1.6

*Différentiation de trafic par un modem ADSL de première génération*

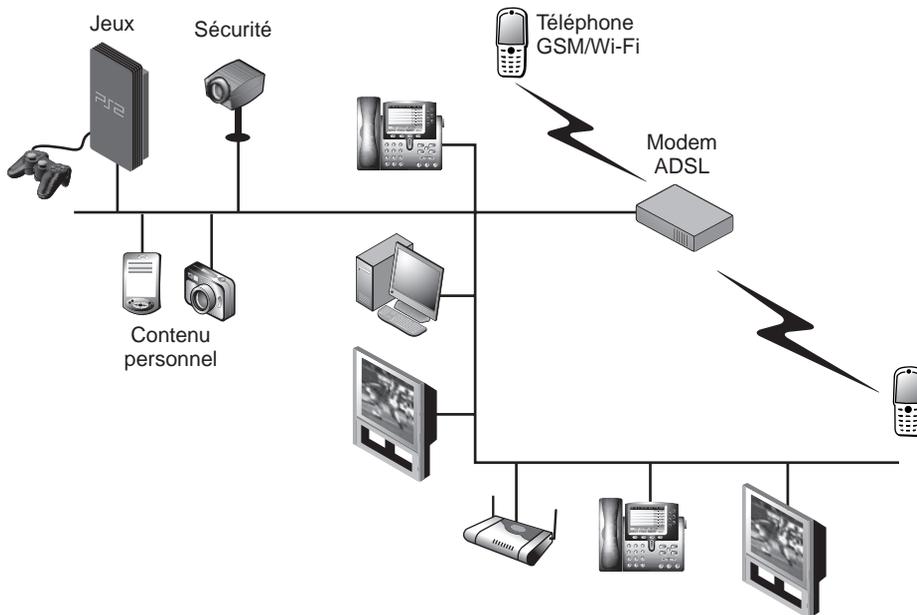
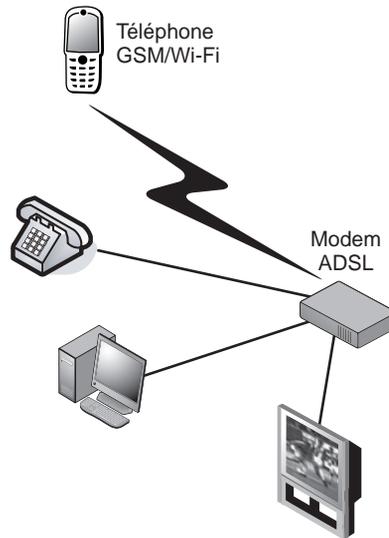


Figure 1.7

*Différentiation de trafic par un modem ADSL de nouvelle génération*

Cette même évolution vaut pour les petites et moyennes entreprises, pour lesquelles le PBX-IP deviendra une sorte de gros modem ADSL, de nombreuses fonctionnalités étant exportées vers le réseau de l'opérateur ou des fournisseurs de services particuliers.

## Les solutions de ToIP

Le développement de la ToIP a vu se succéder sur plusieurs années différentes générations de services et de configurations.

La première génération de téléphonie IP grand public a été proposée par des opérateurs alternatifs afin d'offrir des communications internationales à tarif local. Ce service consiste à rassembler un grand nombre de voies téléphoniques classiques sur le commutateur local et à les encapsuler dans un même paquet IP. Ce paquet IP peut devenir assez important suivant le nombre de voix multiplexées et le nombre d'octets de chaque voix.

L'utilisateur se connecte en local sur le commutateur de l'opérateur historique. L'opérateur alternatif récupère les différentes voix et les multiplexe sur Internet ou sur une même liaison IP, transatlantique par exemple. À la sortie du réseau IP, les voies de parole retrouvent leur composition normale sur le commutateur local et sont envoyées de façon classique aux destinataires au travers de la boucle locale de l'opérateur de télécommunications.

Si la téléphonie locale est gratuite, comme aux États-Unis, le coût total est approximativement égal à la tarification locale de départ. Les opérateurs de téléphonie classique suivent plus ou moins les mêmes principes, tout en tentant de préserver une marge bénéficiaire importante. D'où une chute des prix beaucoup plus lente.

Cette solution est illustrée à la figure 1.8.

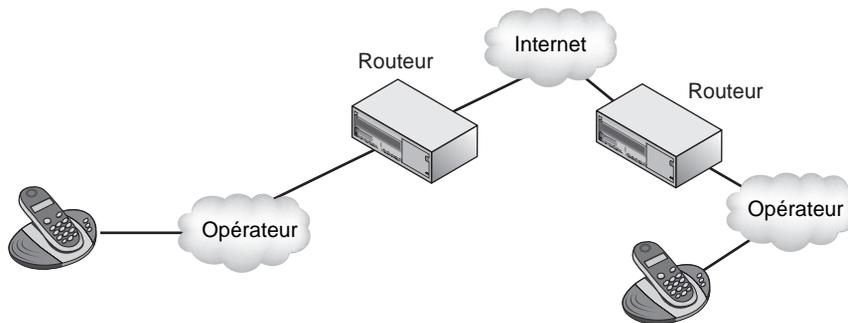


Figure 1.8

*La première génération de téléphonie sur IP*

La deuxième génération a vu les opérateurs de télécommunications offrir des accès Internet au travers de la boucle locale *via* des modems standards permettant des débits de l'ordre de 50 Kbit/s.

Sur cet accès Internet peuvent être raccordés des ordinateurs personnels. Si l'ordinateur est équipé d'un micro et d'un haut-parleur, il est possible d'utiliser l'ordinateur personnel comme téléphone et de faire transiter les paquets de téléphonie sur Internet après les avoir acheminés sur la boucle locale de l'opérateur.

Cette amélioration est illustrée à la figure 1.9.

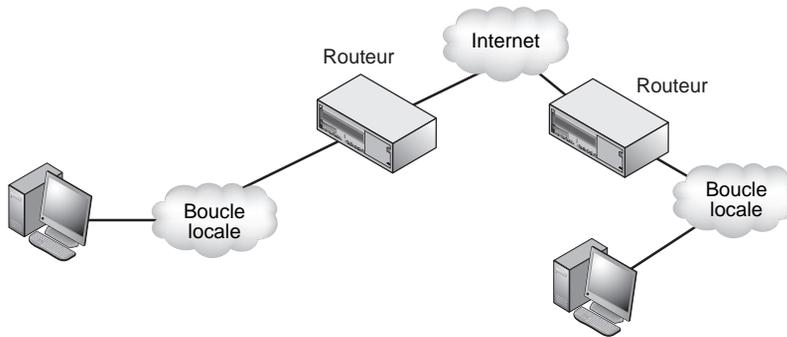


Figure 1.9

*La téléphonie au travers de l'ordinateur personnel*

Dans la troisième génération, au lieu d'utiliser l'ordinateur comme téléphone, un combiné analogique est connecté au PC, équipé d'une carte d'acquisition de la parole téléphonique.

L'ordinateur personnel joue ici le rôle d'une passerelle, transformant le signal analogique du combiné en un flux d'octets de téléphonie numérisés par un codec intégré à l'ordinateur. Les octets sont envoyés par un modem vers le routeur de l'opérateur, auquel revient la charge de la paquetsation et de l'envoi des paquets IP.

Cette étape est illustrée à la figure 1.10.

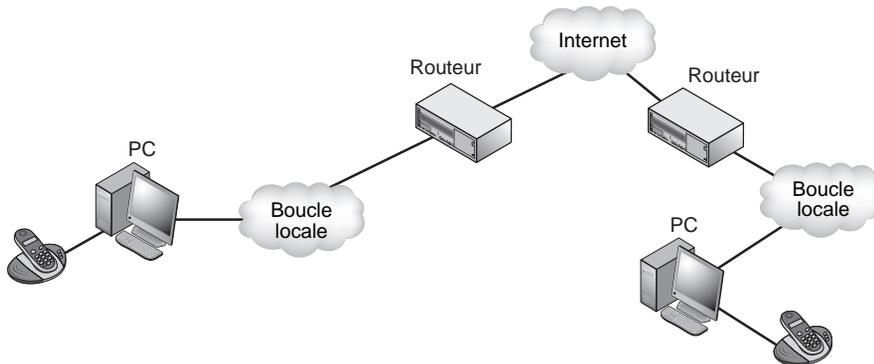


Figure 1.10

*Téléphonie IP utilisant l'ordinateur personnel comme intermédiaire*

La quatrième génération est caractérisée par l'arrivée de modems ADSL munis de plusieurs prises, chacune prenant en charge un média particulier et un protocole associé. Le modem ADSL permet de connecter des téléphones standards. Les conversions nécessaires sont effectuées dans le modem, qui devient de ce fait une véritable Internet

Box, le travail spécifique de la partie modem devenant mineur par rapport à l'ensemble des fonctionnalités réseau réalisées.

La boucle locale de l'opérateur transporte les paquets IP. Pour sa part, le téléphone demeure analogique.

Cette solution est illustrée à la figure 1.11.

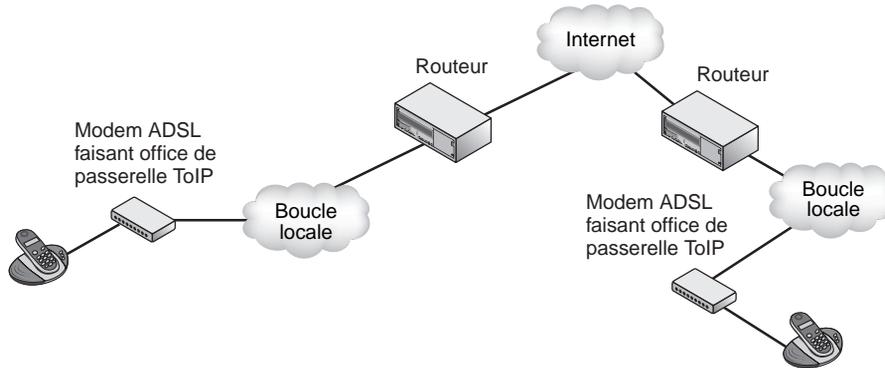


Figure 1.11

*Apparition du modem ADSL dans la chaîne de transmission de la téléphonie*

La cinquième génération du processus aboutit à de la téléphonie IP de bout en bout. La paquetsation est repoussée dans l'équipement terminal de l'utilisateur. Le téléphone devient un téléphone IP.

La figure 1.12 illustre cette solution. Le téléphone IP n'est pas connecté directement sur la boucle locale de l'opérateur mais sur le réseau d'entreprise, lui-même connecté à l'opérateur. Le téléphone IP fait en réalité office de routeur. Il intègre en outre un codec et assure la paquetsation IP et l'encapsulation des paquets IP dans une trame Ethernet. La trame Ethernet est ensuite transmise sur le réseau d'entreprise.

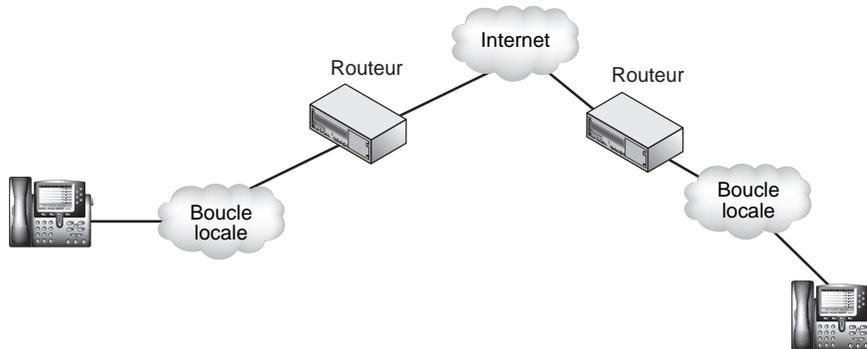


Figure 1.12

*La téléphonie IP de bout en bout*

Avec l'arrivée massive d'ordinateurs personnels suffisamment puissants pour émuler un téléphone IP, la ToIP est devenue une téléphonie de bout en bout gratuite, puisque la téléphonie devient une application comme une autre transitant par l'intermédiaire du modem ADSL.

Du fait de cette configuration, de nouvelles applications ont fait leur apparition pour proposer des services grand public. Parmi celles-ci, Skype ou MSN (Microsoft Network) proposent de la téléphonie sur IP de bout en bout.

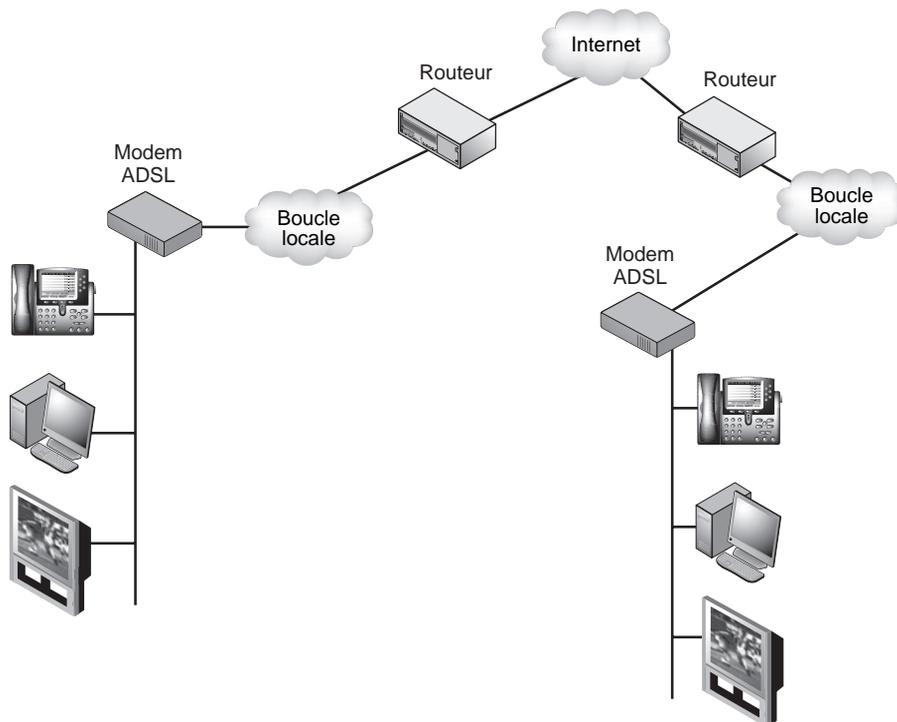
Il faut dans les deux cas disposer d'un modem ADSL aux deux extrémités de la communication afin que le débit soit acceptable sur la boucle locale. Skype fait appel à une technique P2P (peer-to-peer) à des fins de simplicité et pour ne pas avoir à implémenter un contrôle centralisé. La signalisation de MSN est gérée par une base de données centralisée mais qui peut être distribuée sur plusieurs sites.

Nous examinons en détail dans la suite de l'ouvrage ces solutions de ToIP, qu'elles proviennent des opérateurs ou d'applications uniquement terminales. Le modem ADSL joue le rôle de codec et de paquetsiseur. Le téléphone est branché sur une prise spécifique reliée au codec. La télévision et les données ont leur propre prise spécifique.

En cas d'utilisation d'un logiciel de téléphonie sur l'ordinateur portable, le flux de téléphonie est multiplexé avec l'ensemble des données et n'est pas traité de façon spécifique. On appelle cette solution, le Double-Play lorsqu'il y a un canal de données et un canal téléphonique et Triple-Play lorsqu'un canal de télévision est ajouté (voir figure 1.13).

Figure 1.13

Le Triple-Play

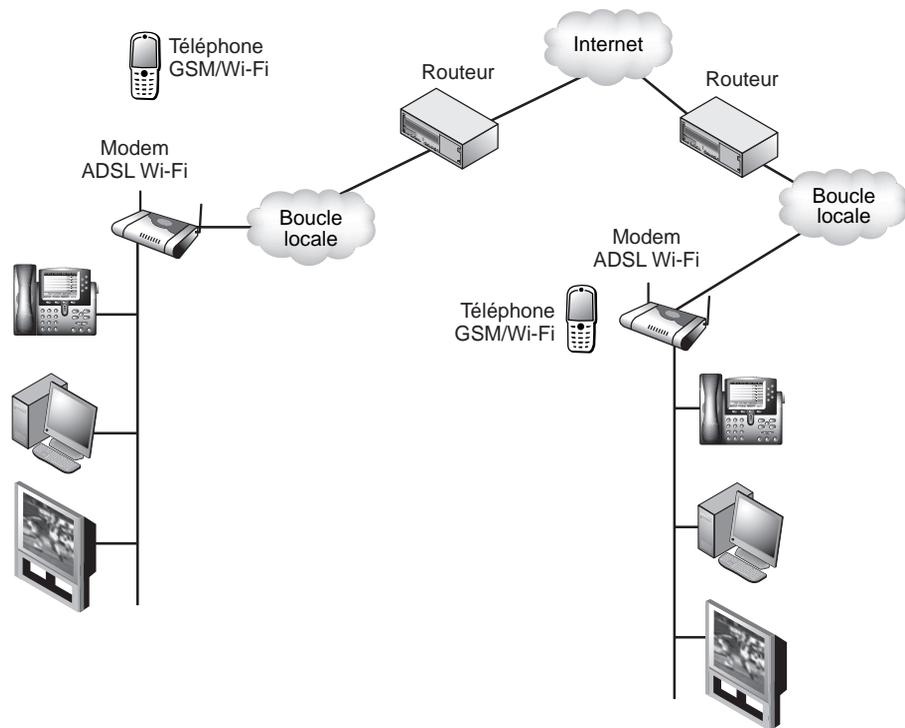


Si l'on ajoute un canal supplémentaire, comme le canal de mobilité provenant d'un terminal mobile de type GSM/Wi-Fi, on parle de Quadruple-Play. Lorsque ce téléphone est situé près d'un modem incorporant un réseau Wi-Fi, le mobile se connecte en Wi-Fi. S'il n'est pas situé dans une zone Wi-Fi, le téléphone utilise le mode GSM. Il est possible de commencer à téléphoner en Wi-Fi et de continuer en GSM lorsqu'on sort de la zone Wi-Fi. En sens inverse, le téléphone peut éventuellement repasser en Wi-Fi.

Cette solution est illustrée à la figure 1.14.

Figure 1.14

*Le Quadruple-Play*

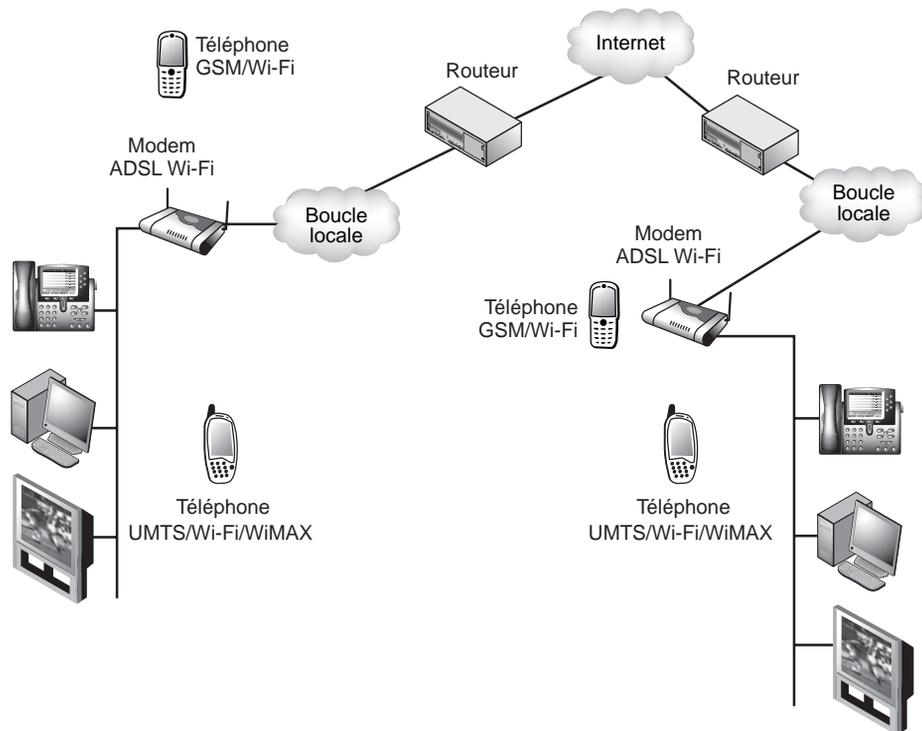


La figure 1.15 illustre la génération suivante, dite Penta-Play, dédiée à la vidéo mobile. Sur un mobile à écran vidéo, un utilisateur peut se connecter sur un réseau Wi-Fi et regarder la télévision. La connexion avec le modem ADSL s'effectue en mode hertzien de type Wi-Fi.

Dans cette solution comme dans la précédente, le téléphone GSM/Wi-Fi peut se connecter à tous les modems de l'opérateur Internet auquel l'utilisateur a souscrit.

La téléphonie sur IP est encore peu présente dans le monde de la communication mobile, mais elle devrait se généraliser dès que les accès Internet lui seront ouverts, ce que les opérateurs interdisent aujourd'hui. Ce déploiement s'effectuera par l'intermédiaire de l'Internet hertzien, mais prendra son essor véritable avec l'arrivée des produits WiMax.

Figure 1.15

*Le Penta-Play*

Pour le moment, les réseaux de mobiles peuvent transporter des paquets IP, qui ne sont jamais qu'un ensemble d'éléments binaires au même titre que toutes autres suites d'éléments binaires. Il est donc possible de mettre en place des applications de téléphonie sur un terminal mobile assez puissant. Le coût de la communication étant celui du transport des données, la téléphonie n'est plus qu'une application parmi les autres.

## Questions posées par la mise en place de la ToIP en entreprise

Nous examinons dans cette section les questions à se poser lors de la mise en place d'un environnement de ToIP en entreprise. Le cas des particuliers est différent puisque c'est à l'opérateur de résoudre les problèmes liés à ces questions.

Cinq questions principales se posent :

- **Sécurité.** Autrefois, les réseaux étaient fortement sécurisés grâce à la notion de circuit. En entrant dans le monde IP, la téléphonie rencontre un monde encore mal sécurisé, qui connaît des problèmes d'authentification, de confidentialité et d'intégrité.
- **Disponibilité.** Autrefois, les réseaux avaient une disponibilité dite à cinq « neuf », signifiant qu'ils fonctionnaient 99,999 % du temps. Les meilleurs réseaux des opérateurs

IP n'ont généralement qu'une disponibilité à trois « neuf » (99,9 % du temps). De nombreux autres réseaux IP ne sont disponibles qu'à 99 % du temps.

- **Gestion.** Les trois réseaux de la génération précédente (données, parole, vidéo) possédaient trois systèmes de gestion relativement simples. Avec l'intégration, il n'y a plus qu'un seul système de gestion, de ce fait assez complexe.
- **Contrôle.** Autrefois les réseaux étaient contrôlés par des algorithmes assez simples. L'intégration des différents flux dans le même réseau complexifie énormément le contrôle de l'ensemble.
- **Qualité de service.** La qualité de service étant liée à l'infrastructure, la nouvelle génération de réseaux doit être capable de prendre en charge les qualités de service de chaque application transitant sur le même réseau, ce qui n'est pas facile.

Dans la suite de l'ouvrage, nous nous penchons en détail sur les réponses techniques apportées à ces questions. Nous y reviendrons également en toute fin d'ouvrage pour synthétiser les éléments de réponses qui auront été apportés tout au long de ce livre.

Revenons quelques instants sur la sécurité, qui passe par la mise en place de pare-feu, permettant dans la mesure du possible d'arrêter les voies de parole interdites. Comme nous le verrons, ces solutions sont complexes et peu utilisées. Les pare-feu traditionnels filtrant les trafics à partir des numéros de port sont généralement incapables de stopper les flux de téléphonie. Il est donc nécessaire de recourir à des pare-feu applicatifs, capables d'identifier les applications de niveau 7 (applicatif).

Pour l'authentification et l'autorisation de l'émetteur, le chiffrement et la signature électronique sont nécessaires. Il faut en outre vérifier que la parole n'a pas été déformée, voire remplacée par une autre.

Le tableau 1.1 donne les idées de grandeur des durées et des coûts engendrés par l'indisponibilité du réseau.

**Tableau 1.1 – Taux de disponibilité et coûts de l'indisponibilité**

Nombre de « neuf »	Disponibilité	Durée d'indisponibilité	Type de réseau	Coût
1	90 %	36,5 j/an		C
2	99 %	3,65 j/an		2 C
3	99,9 %	8,8 h/an	Bon réseau IP	4 C
4	99,99 %	53 min/an		8 C
5	99,999 %	5 min/an	Téléphonie classique	16 C
6	99,9999 %	32 s/an		32 C

La durée d'indisponibilité passe de cinq minutes par an dans le cas d'un réseau téléphonique classique à presque neuf heures de pannes dans un très bon réseau IP. Il est toujours possible d'augmenter la disponibilité par duplication des lignes ou des chemins.

Pour passer d'une disponibilité de trois « neuf » à une disponibilité de cinq « neuf », il faut cependant multiplier les coûts par quatre. La dernière colonne du tableau montre que si le coût est de  $C$  pour un réseau de disponibilité 90 %, il est de  $2C$  pour gagner un « neuf », et il faut encore multiplier par 2 pour gagner un autre « neuf », etc.

La gestion devient également de plus en plus complexe. De nombreux groupes de travail se sont formés depuis plusieurs années pour promouvoir telle ou telle solution. Aujourd'hui, presque tout reste à faire du fait du passage de la gestion à des technologies IP.

Le contrôle est dans une situation similaire. Il est aujourd'hui quasiment impossible de contrôler efficacement un très grand réseau d'opérateur Internet. De nouvelles techniques sont en train d'émerger, comme les contrôles dits « autonomic » (parfois traduits en français par « auto-organisés »), qui permettent d'automatiser la commande de la plupart des algorithmes de contrôle. Le terme « autonomic » indique un processus autonome et spontané.

La qualité de service est un problème primordial. À ce titre, elle est abondamment couverte dans l'ouvrage. Sans qualité de service, la parole téléphonique ne peut traverser un réseau IP sans dommage. Comme nous le verrons, deux types de qualités de service peuvent être mis en place : soit le flux de la couche application s'adapte au réseau, soit le réseau s'adapte à la demande du flux applicatif.

La première solution est la plus ancienne. Elle correspond à une adaptation du flux par rapport à un réseau qui réagit en best-effort. Les applications téléphoniques telles que Skype utilisent cette solution. La seconde solution prend en compte les possibilités du réseau en permettant à un flux dont on connaît les caractéristiques de traverser le réseau dans les meilleures conditions de garantie. La seconde génération d'Internet correspond à cette solution. La plupart des offres de téléphonie sur IP dans l'entreprise reposent sur elle.

## Conclusion

La téléphonie reste une des applications dominantes du monde des réseaux, et ce pour encore de nombreuses années, en raison notamment de l'émergence de nouveaux et immenses marchés, comme celui de la Chine. L'application de téléphonie ne représente plus mi-2008 que 30 % en moyenne du chiffre d'affaires des opérateurs, tout en n'occupant que 4 % du débit total des communications.

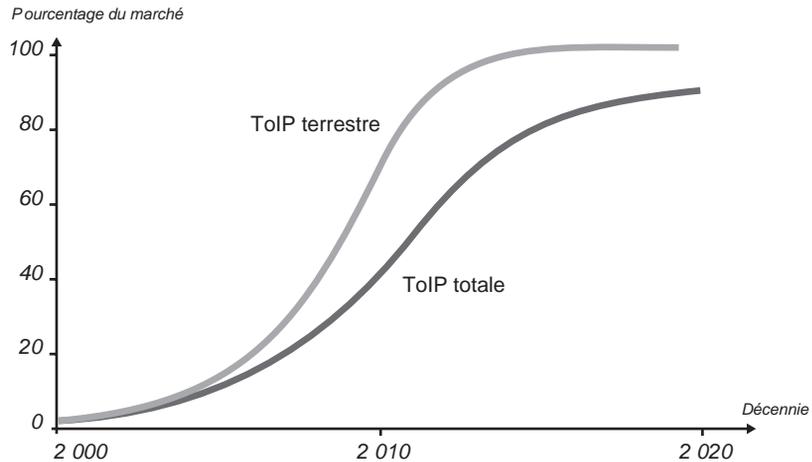
À cette même date (mi-2008), la ToIP mobilise près de 65 % des débits téléphoniques dits terrestres (excluant les mobiles). Le passage vers le tout-IP téléphonique, permettant d'intégrer les services de données et la téléphonie dans un même réseau, sera effectif vers la fin de 2009 en France.

Cependant, la qualité est très variable en fonction des efforts effectués par les gestionnaires de réseaux d'entreprise et les opérateurs de réseaux de télécommunications. Les problèmes à résoudre sont nombreux et parfois complexes. La ToIP n'est pas une application

simple à mettre en œuvre dans le contexte de l'intégration de tous les services de télécommunications sur le même réseau.

La figure 1.16 illustre la place que tiendra la ToIP dans les années à venir.

**Figure 1.16**  
*Évolution de la ToIP sur vingt ans*



En 2010, pratiquement tout le marché de la téléphonie terrestre sera passé en IP. Si l'on considère la téléphonie hertzienne, sa montée en puissance sera beaucoup plus longue. Avec l'UMTS et ses successeurs, le monde de la téléphonie hertzienne suit les traces du GSM, qui n'est pas une solution IP native. Il faudra donc attendre l'extension massive des réseaux sans fil IP de types Wi-Fi, WiMax et autres WiMedia et WiRAN avant de rejoindre les courbes terrestres.

Il est à noter que l'UMTS apporte une solution de téléphonie par paquet mais non-IP.



# 2

## Contraintes de la ToIP

---

La téléphonie sur IP possède les mêmes contraintes de communication temps réel que la téléphonie classique.

Lorsque deux personnes sont l'une en face de l'autre, le temps de transit du signal sortant de la bouche d'un utilisateur est quasiment nul. Lorsque les deux personnes sont à distance et communiquent par l'intermédiaire d'un réseau, la même contrainte doit être vérifiée. Cette contrainte est de 300 ms entre le moment où le signal sort de la bouche jusqu'au moment où il arrive à l'oreille du destinataire.

La valeur de 300 ms correspond à une limite supérieure. Pour ne pas avoir l'impression que le correspondant est situé à l'autre bout de la Terre, un délai de 150 ms est préférable. Nous allons détailler cette contrainte du temps de transit, ainsi que les autres contraintes qui pèsent sur la ToIP.

### Les contraintes temporelles

La principale difficulté pour réaliser de la téléphonie par paquet provient de la contrainte temporelle très forte due à l'interaction entre individus. Le temps de latence, c'est-à-dire le temps qui s'écoule entre l'entrée d'un paquet dans le réseau et son temps de sortie du réseau doit être inférieur à 300 ms si l'on veut garder une interaction humaine acceptable. Si l'on souhaite une bonne qualité de la conversation, il ne faut pas que la latence soit supérieure à 150 ms.

Un cas encore plus complexe se produit lorsqu'il y a un écho, c'est-à-dire un signal qui revient dans l'oreille de l'émetteur. L'écho qui repart en sens inverse est numérisé par un codec (codeur/décodeur) et traverse sans problème un réseau numérique. La valeur normalisée de la latence de l'écho étant de 56 ms, pour que l'écho ne soit pas gênant à l'oreille, il ne faut pas que le temps de transit de la communication dépasse 28 ms dans

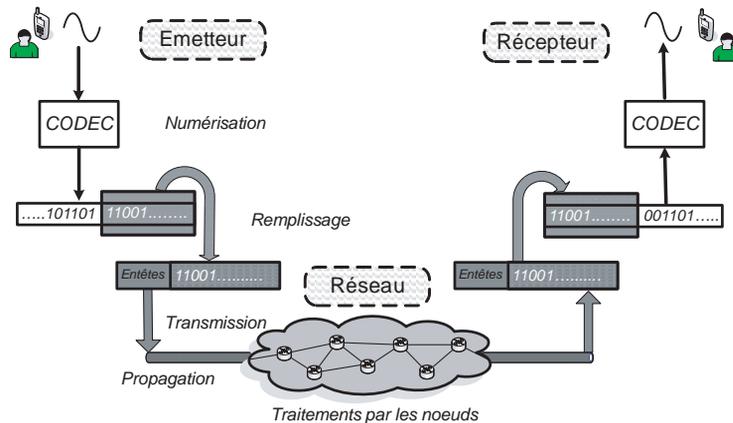
un sens, en supposant un réseau symétrique, demandant le même temps de transit à l'aller et au retour.

Dans les équipements terminaux, les logiciels aux extrémités doivent être capables de gérer les retards et de resynchroniser les octets qui se présentent. En règle générale, les téléphones IP ou les ordinateurs personnels possèdent des supprimeurs d'écho évitant cette contrainte temporelle forte.

Le temps de transfert d'un flux de parole téléphonique est constitué de la somme des cinq temps suivants (voir figure 2.1) :

**Figure 2.1**

*Temps de transfert d'un paquet*



- **Temps de numérisation de la voix** : la parole téléphonique est un signal analogique impossible à coder sur un ordinateur. Il faut la numériser en données numériques (un ensemble de 1 et de 0) à l'aide d'un codec (codeur/décodeur), dont nous expliquerons plus loin le fonctionnement. Le temps de numérisation est généralement négligeable, mais le codec va déterminer la vitesse à laquelle les données seront émises.
- **Temps de remplissage des paquets** : les données envoyées sont assemblées en paquets. Ces paquets comportent des en-têtes (avec entre autre l'adresse du récepteur pour router le paquet et l'adresse de l'émetteur pour recevoir une réponse), qui sont placés une fois le paquet constitué. Plus il y a de données dans le paquet, moins il y a d'en-têtes, et donc de consommation de bande passante. En revanche, plus l'attente avant d'envoyer le paquet est longue, plus l'interactivité de la communication est compromise. Il est donc nécessaire d'avoir suffisamment de données pour remplir le paquet avec une certaine taille avant de l'envoyer dans le réseau. On peut définir le temps de remplissage comme le temps mis par le codec pour remplir un paquet de taille fixée (cette taille ne prend pas en compte les en-têtes qui sont ajoutés automatiquement et indépendamment du codec).

Remarquons que nous ne considérons pas le temps que met l'émetteur pour l'encapsulation du paquet (avec l'ajout des en-têtes qui correspondent à chaque couche réseau traversée, conformément au modèle OSI), ni le temps que met le récepteur pour remonter

le paquet au niveau du logiciel de téléphonie (c'est-à-dire le temps de la décapsulation des en-têtes du paquet), qui sont des temps négligeables.

- **Temps de propagation** : un signal binaire envoyé d'un endroit arrive à un autre endroit après un certain temps, qui dépend de la distance à parcourir. Le temps de propagation peut donc être défini comme le rapport de la distance à parcourir entre l'émetteur et le récepteur sur la vitesse de propagation du signal. On prend généralement une vitesse de propagation d'un signal de 200 000 km/s.
- **Temps de transmission** : des données arrivent d'un point à un autre selon un temps qui dépend de la quantité de données émise et du débit auquel fonctionnent les liens entre l'émetteur et le récepteur. On peut donc définir le temps de transmission comme le rapport de la quantité de données à envoyer sur le débit du lien considéré.
- **Temps de traitement par les nœuds intermédiaires** : pour aller de l'émetteur au récepteur, les flux de données parcourent un ensemble de routeurs intermédiaires qui les aiguillent jusqu'à leur source. Chacun de ces nœuds ajoute un délai supplémentaire, qui constitue le temps de traitement des nœuds intermédiaires. Ce temps est généralement de l'ordre de la milliseconde pour chaque nœud.

Prenons un exemple pour illustrer l'ensemble des composants constituant le temps de transfert. On considère un réseau de type Ethernet à 100 Mbit/s. L'application logicielle de l'émetteur numérise la parole téléphonique en un temps négligeable. Elle utilise un codeur qui fonctionne à une vitesse de 8 Kbit/s et génère la transmission de paquets d'une taille de 64 octets (comprenant 16 octets d'en-tête). Le temps de propagation considéré est de 200 000 km/s, et la liaison entre l'émetteur et le récepteur comporte 7 nœuds, chacun traitant un paquet en 1 ms.

Nous allons chercher la distance maximale  $D_{\max}$  entre les correspondants pour assurer un temps de transfert d'au plus 150 ms.

Le temps de transfert vaut ainsi :

$$T_{\text{transfert}} = T_{\text{numérisation}} + T_{\text{remplissage}} + T_{\text{propagation}} + T_{\text{transmission}} + T_{\text{traitement\_noeud}}$$

Détaillons chacun de ces temps séparément :

$$T_{\text{numérisation}} = 0 \text{ ms (négligé)}$$

$$T_{\text{remplissage}} = (64 - 16) \text{ octets} / 8 \text{ Kbit/s} = 384 \text{ bits} / 8,10^3 \text{ bits} = 48 \text{ ms}$$

$$T_{\text{propagation}} = D_{\max} / (200\,000 \text{ km/s}) = D_{\max} / (200 \text{ km/ms})$$

$$T_{\text{transmission}} = 64 \text{ octets} / 100 \text{ Mbit/s} = 512 \text{ bits} / 10^8 \text{ bits} = 0,005\,12 \text{ ms (négligeable)}$$

$$T_{\text{traitement\_noeud}} = 7 \times 1 = 7 \text{ ms}$$

Pour que le temps de transfert soit inférieur à 150 ms, il faut donc que :

$$T_{\text{transfert}} = 0 + 48 + D_{\max} / 200 \text{ km} + 0 + 7 < 150$$

Soit une distance  $D_{\max}$  de :

$$D_{\max} < (150 - 55) \times 200 = 19\,000 \text{ km}$$

Dans ces conditions, la distance entre l'émetteur et le récepteur doit être inférieure à 19 000 km pour assurer un temps de transfert d'au plus 150 ms.

## Le processus de resynchronisation de la parole téléphonique

Du fait de l'interactivité entre deux interlocuteurs, la téléphonie IP implique une contrainte temporelle de 600 ms (300 ms dans chaque sens). Il faut donc, après le transport des échantillons dans le réseau, les resynchroniser au récepteur de sorte qu'un flot régulier soit remis au codec. La difficulté provient du temps de transport asynchrone, assez aléatoire, du réseau, qui implique que les paquets sont remis au récepteur à des instants quelconques.

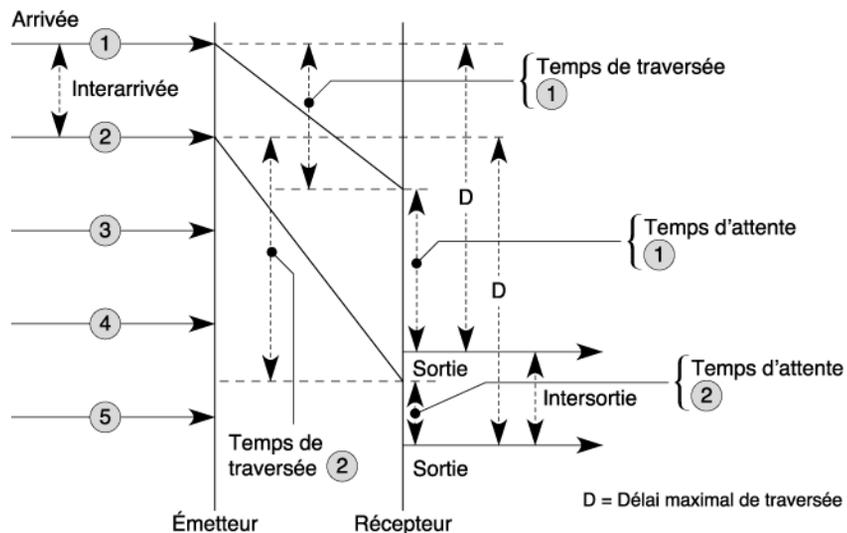
Pour réaliser une resynchronisation, il faut mémoriser les paquets au récepteur pendant un certain temps, appelé temps de synchronisation. Pour déterminer ce temps de synchronisation, on doit au préalable déterminer le délai maximal d'attente entre le moment de l'entrée du paquet dans le réseau et celui de sa délivrance au codec.

Ce délai doit bien sûr être inférieur à la contrainte temporelle de l'application. Si l'application est une téléphonie avec écho, il faut choisir une valeur de l'ordre de 15 ms. Si l'application est de la téléphonie IP de bout en bout, une valeur classique est de l'ordre de 100 ms.

Soit  $D$  la valeur du délai maximal de traversée du réseau. Le temps de synchronisation est défini comme le temps écoulé depuis l'entrée dans le réseau augmenté de la valeur  $D$ . En d'autres termes, entre les temps d'entrée et de sortie, il y a un décalage de  $D$ .

Ce processus est illustré à la figure 2.2.

**Figure 2.2**  
*Processus de synchronisation de la parole téléphonique*



Pour réaliser un tel algorithme, il faut que le nœud de sortie connaisse les temps d'accès dans le réseau afin de pouvoir synchroniser les paquets à ces temps d'accès augmentés du délai maximal de traversée. Pour cela, on utilise aux deux extrémités des horloges synchrones ou dont le synchronisme permet les adjonctions de temps.

Différents algorithmes de synchronisation sont utilisés dans les réseaux qui transportent de la parole. Ils travaillent soit en synchronisation directe, soit en synchronisation différentielle. Dans le premier cas, les horloges indiquent exactement la même valeur de temps. Dans le second cas, les horloges tournent à la même vitesse, même si elles sont décalées. On utilise des trames de synchronisation pour déterminer la différence entre l'horloge d'entrée et celle de sortie.

## La téléphonie numérique

Les trois opérations successives nécessaires à la numérisation de la parole, qu'elle soit téléphonique ou non, sont les suivantes :

1. **Échantillonnage.** Consiste à prendre des points du signal analogique au fur et à mesure qu'il se déroule. Il est évident que plus la bande passante est importante, plus il faut prendre d'échantillons par seconde. C'est le théorème d'échantillonnage qui donne la solution : il faut échantillonner à une valeur égale à au moins deux fois la bande passante. Pour une bande passante de 3 100 Hz, correspondant à la bande des 300 à 3 400 Hz, il faut échantillonner au moins 6 200 fois par seconde. Si la bande passante est de 20 000 Hz, il faut au moins 40 000 échantillons par seconde. On comprend ainsi pourquoi la bande passante de la téléphonie est limitée.
2. **Quantification.** Consiste à représenter un échantillon par une valeur numérique au moyen d'une loi de correspondance. Cette phase consiste à trouver une loi de correspondance telle que la valeur des signaux ait le plus de signification possible. Cette quantification détermine la justesse avec laquelle le codage peut s'effectuer. Si le codage est sur 7 bits, cela implique 128 niveaux possibles. Plus la largeur de bande est importante, plus la longueur du codage doit être importante. Si l'on reprend l'exemple précédent, les 128 niveaux de codage sont équidistribués, mais la parole reste essentiellement dans un niveau de fréquences situé entre 500 Hz et 1 500 Hz sur les 3 100 Hz de bande passante. Les différences entre les échantillons sont relativement importantes dans la bande des 1 000 Hz, la plus utilisée. Mieux vaut avoir des intervalles plus petits sur les fréquences fortement utilisées et des intervalles plus grands sur les fréquences peu utilisées. La loi de correspondance uniforme n'est généralement pas la meilleure solution. Il faut trouver des lois qui favorisent les intervalles de fréquences fortement utilisés. En règle générale, on utilise des lois semi-logarithmiques.
3. **Codage.** Consiste à donner une valeur numérique aux échantillons. Ce sont ces valeurs qui sont transportées dans les paquets. Dans l'exemple précédent, si l'on souhaite déterminer 128 intervalles, il faut choisir 7 bits pour le codage.

## L'échantillonnage

La largeur de bande de la voix téléphonique analogique est de 3 100 Hz. Pour numériser ce signal correctement sans perte de qualité, puisqu'elle est déjà relativement mauvaise, il faut échantillonner au moins 6 200 fois par seconde. La normalisation a opté pour un échantillonnage de 8 000 fois par seconde.

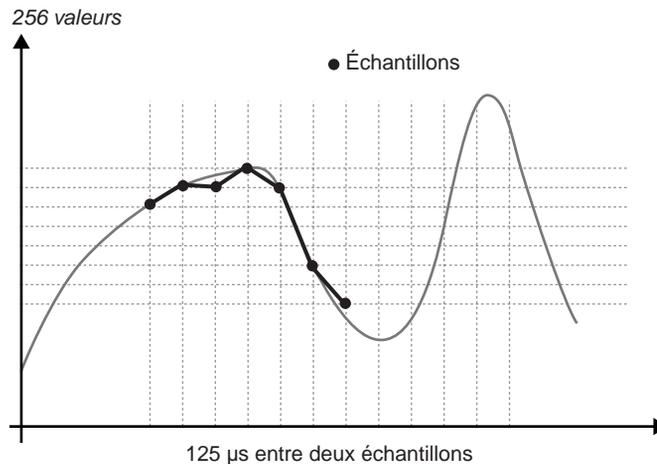
La quantification s'effectue par des lois semi-logarithmiques. L'amplitude maximale permise se trouve divisée en 128 échelons positifs pour la version américaine PCM (Pulse Code Modulation), auxquels il faut ajouter 128 échelons négatifs dans la version européenne MIC (modulation, impulsion et codage). Le codage s'effectue donc soit sur 128 valeurs, soit sur 256 valeurs, ce qui demande en binaire 7 ou 8 bits de codage.

La valeur totale du débit de la numérisation de la parole téléphonique s'obtient en multipliant le nombre d'échantillons par le nombre d'échelons, ce qui donne :

- $8\,000 \times 7 \text{ bit/s} = 56 \text{ Kbit/s}$  en Amérique du Nord et au Japon ;
- $8\,000 \times 8 \text{ bit/s} = 64 \text{ Kbit/s}$  en Europe.

La figure 2.3 illustre ce processus.

**Figure 2.3**  
*Processus de numérisation de la parole*



On voit que la numérisation est une approximation : on choisit la valeur de l'échantillon la plus proche possible de la valeur réelle. Il y a toujours une différence entre la valeur réelle et la valeur de l'échantillon choisie sur le quadrillage. En effet, seules les valeurs du quadrillage ont une valeur numérique. Lorsque le récepteur reçoit la valeur de l'échantillon, il considère que c'est la valeur exacte du signal, et il relie les échantillons entre eux pour réaliser une courbe, qui est en réalité constituée d'une succession de droites tirées entre deux points.

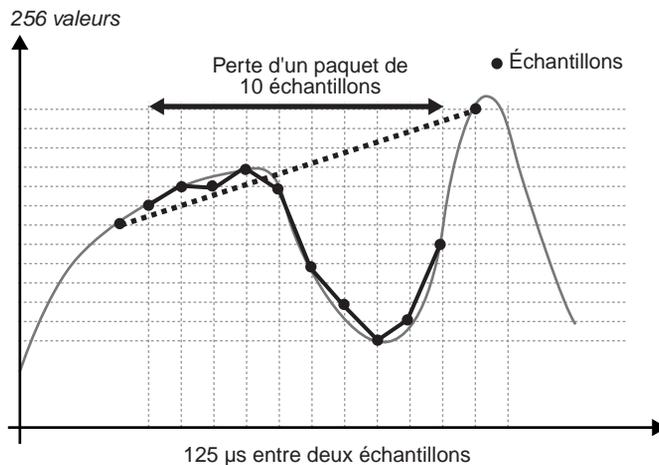
Si le nombre d'échantillons est suffisant, le nouveau signal obtenu est presque identique au signal de départ, et la qualité de la parole est maintenue.

Les échantillons sont placés dans un paquet à la vitesse d'échantillonnage. Dans le cas classique, les octets arrivent toutes les  $125 \mu\text{s}$ . Pour que le temps de réponse de bout en bout soit acceptable, il ne faut pas dépasser une certaine valeur, qui est généralement limitée à 16 ms dans la téléphonie sur IP, ce qui représente dans le cas classique 128 octets. Comme nous allons le voir, cette valeur de 16 ms est beaucoup plus contraignante lorsque la parole est compressée puisque le nombre d'octets qui pourront être transportés est beaucoup plus faible.

La perte d'un échantillon de temps en temps n'est pas grave. Il suffit de remplacer l'octet manquant par un octet estimé à partir du précédent et du suivant. La perte d'un paquet n'est pas non plus catastrophique, puisque le temps perdu n'est que de quelques millisecondes.

La figure 2.4 illustre le cas où un paquet contenant 10 échantillons est perdu. Pour avoir une courbe continue, on relie le dernier échantillon reçu au premier de la trame reçue. Avec cette solution, l'oreille ne peut détecter la perte du paquet, sachant que même dans le cas d'une assez forte compression de la parole, par exemple à 8 Kbit/s, cela ne représente que 10 ms.

**Figure 2.4**  
*Perte d'un paquet  
de 10 échantillons*



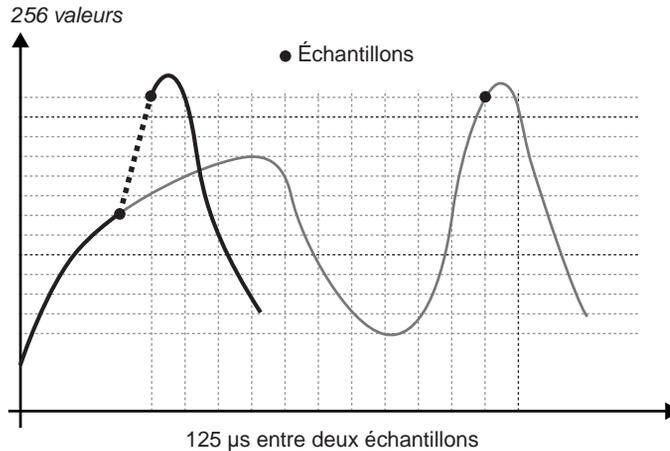
La perte d'un paquet doit cependant être détectée. En effet, le risque serait qu'un paquet soit perdu et que le récepteur, à la réception du paquet suivant, recolle le premier octet de ce nouveau paquet juste après le dernier octet du précédent (voir figure 2.5).

Dans ce cas, le signal reformé peut changer de fréquence brutalement, s'accompagnant d'un bruit gênant pour l'oreille. Il faut donc détecter la perte d'un paquet, par exemple par une numérotation régulière des paquets, et laisser un intervalle correspondant à la longueur du paquet qui sera rempli par une droite reliant le dernier octet bien reçu au premier octet du paquet arrivé après le paquet perdu. C'est la droite en pointillés illustrée à la figure 2.4.

Il est à noter qu'un algorithme astucieux pourrait approximer beaucoup mieux qu'une droite le signal entre ces deux points en tenant compte des pentes du signal aux deux octets à joindre.

**Figure 2.5**

*Risque de rupture en cas de non-détection d'un paquet perdu*



## Techniques de codage

Il est possible de compresser la parole numérisée par différentes techniques. La plus classique d'entre elles est le codage différentiel, qui consiste à travailler sur les différences entre les échantillons plutôt que d'effectuer un codage absolu, comme dans le cas présenté précédemment, où chaque point était codé indépendamment du point précédent ou du point suivant.

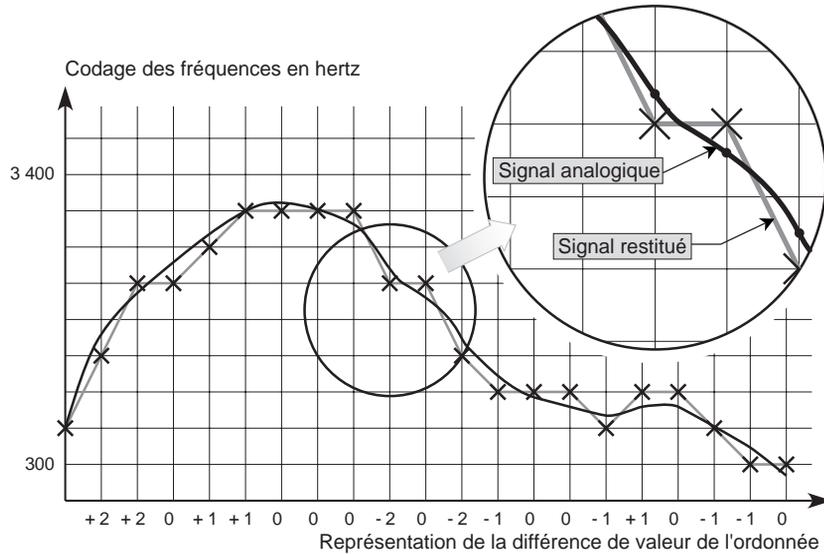
Le codage différentiel permet de compresser le flot de données. Au lieu de coder la valeur complète de l'échantillon, on ne transmet que la différence avec l'échantillon précédent. Comme le nombre d'échantillons est important toutes les secondes, la différence entre deux échantillons est généralement très petite.

Un exemple de codage différentiel est illustré à la figure 2.6. Le codage de la différence demande moins d'éléments binaires que le codage complet d'un échantillon. Chaque fois qu'une différence est transmise, on effectue une approximation puisque la valeur d'un échantillon est codée par la meilleure valeur possible, mais qui n'est pas exacte. De ce fait, on accumule les approximations. Au bout de quelques dizaines d'échantillons, la valeur peut devenir fortement erronée.

C'est la raison pour laquelle il faut envoyer à intervalle régulier une valeur complète d'un échantillon pour continuer la transmission. On peut en conclure que la compression génère un flot variable dans le temps : régulièrement, on a un codage complet de l'échantillon (sur 8 bits pour la parole téléphonique), puis, entre deux valeurs complètes, les échantillons ne demandent plus que 3 ou 4 bits de codage, ce qui permet de diminuer le débit.

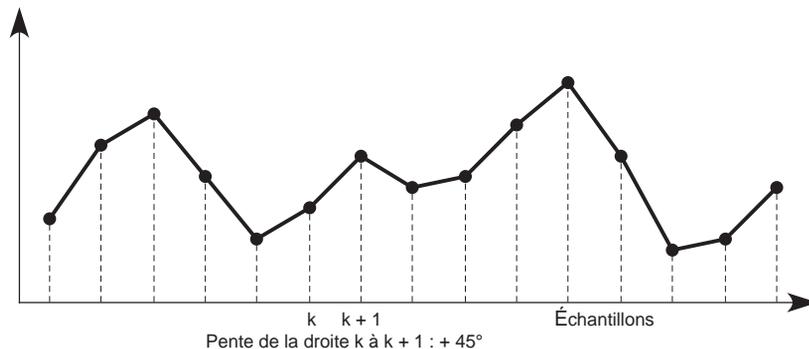
Parmi les autres techniques de numérisation de la parole, signalons celles qui consistent à travailler en temps réel ou en temps différé. Dans le premier cas, l'algorithme qui permet de traduire la loi intermédiaire de quantification doit être exécuté en temps réel.

**Figure 2.6**  
*Codage différentiel de la parole téléphonique*



Les éléments binaires obtenus ne sont pas compressés, ou très peu. Dans le second cas, la parole peut être stockée sur des volumes beaucoup plus faibles, mais le temps nécessaire pour effectuer la décompression est trop long pour régénérer un flot synchrone d’octets, et donc le signal analogique de sortie. Il faut une mémorisation intermédiaire qui supprime le caractère temps réel de la parole. Pour les messageries numériques, une compression est presque toujours effectuée afin que les capacités de stockage ne soient pas trop sollicitées. Dans ce cas, on descend à des débits inférieurs à 2 Kbit/s.

**Figure 2.7**  
*Numérisation par la méthode Delta*



On peut citer parmi les techniques temps réel les méthodes  $\Delta$  (Delta) ou  $\Delta_M$  (Delta Modulation), qui s’appuient sur le codage d’un échantillon en relation avec le précédent. Par exemple, on peut définir le point d’échantillonnage  $k + 1$  par la pente de la droite reliant les échantillons  $k$  et  $k + 1$ , comme illustré à la figure 2.7. On envoie la valeur exacte du premier échantillon, puis on ne transmet que les pentes. Étant donné que la

pende de la droite ne donne qu'une approximation du point suivant, il faut régulièrement émettre un nouvel échantillon avec sa valeur exacte.

Grâce à ces méthodes, le débit de la parole numérique peut descendre à 32, 16 ou 8 Kbit/s, voire moins. Si l'on descend jusqu'à 2 Kbit/s, on obtient une parole synthétique, de médiocre qualité. Nous n'avons pris pour exemple jusqu'ici que la parole numérique. Il va de soi que toutes les informations analogiques peuvent être numérisées de la même façon.

Bien d'autres solutions que celles que nous venons de voir ont été développées pour compresser la parole téléphonique en utilisant les qualités et les défauts de l'oreille :

- AD-PCM (Adaptive Differential-Pulse Code Modulation), ou modulation par impulsion et codage différentiel adaptatif ;
- SBC (Sub-Band Coding) ;
- LPC (Linear Predictive Coding) ;
- CELP (Code Excited Linear Prediction).

De nombreuses extensions pour obtenir une meilleure qualité de la téléphonie ont été proposées et expérimentées. La plupart concernent l'élargissement de la partie du spectre utilisée pour le codage. L'extension la plus classique permet de coder la parole entre 50 et 7 000 Hz, ce qui donne une bande passante de 6 950 Hz à la place des 3 100 Hz standards.

## Les codeurs audio

De nombreux codeurs audio sont associés aux différentes techniques détaillées précédemment. On trouve notamment les codecs classiques mais aussi de nouveaux codeurs bas débit. On peut classer les codecs en trois grandes classes :

- codeurs en forme d'ondes utilisant l'onde sans compression ;
- codeurs paramétriques utilisant un modèle de production vocale ;
- codeurs hybrides combinant les deux précédents.

Selon ces différentes classes, il existe plusieurs sortes de codecs :

- Codecs PCM (Pulse Code Modulation), les premiers à être apparus, fonctionnant à un débit fixe.
- Codecs AD-PCM (Adaptive Differential-Pulse Code Modulation) fonctionnant à des débits variables s'adaptant au débit de la liaison ou du réseau. Les débits les plus classiques sont de 32, 24 ou 16 Kbit/s.
- Codecs adaptés aux réseaux de mobiles, comme le GSM-EFR (Enhanced Full Rate), normalisé par l'ETSI sous la recommandation GSM 06.60 en 1996 ou encore adapté à l'UMTS.
- Codecs Internet, comme le CELP (Code Excited Linear Prediction).

Pour l'audio haute définition, on considère une bande passante plus importante puisque l'oreille humaine est sensible aux fréquences de 20 à 20 000 Hz. L'échantillonnage s'effectue sur 40 kHz, et c'est la valeur de 44,1 kHz qui a été choisie. Le codage effectué sur un CD tient sur 16 bits par échantillon, ce qui donne 705,6 Kbit/s. Cependant, il existe de nombreuses solutions, certaines normalisées et d'autres propriétaires.

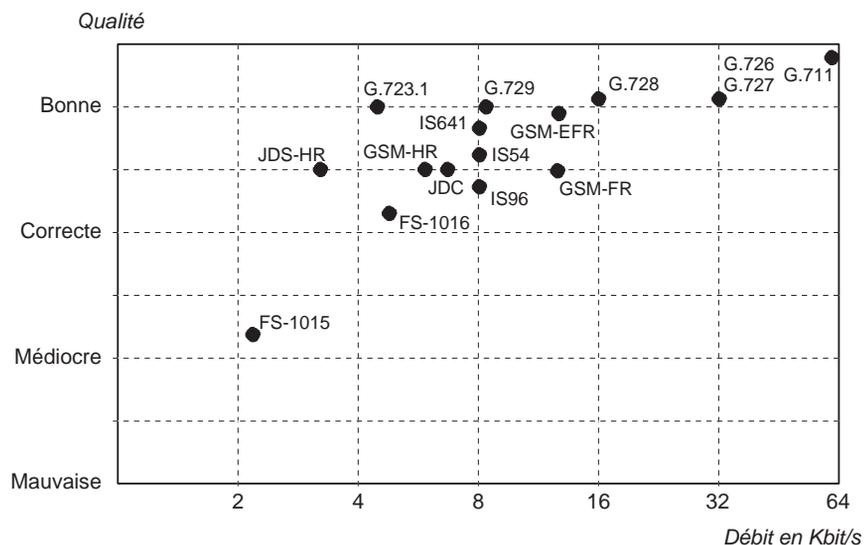
Dans le domaine normalisé, on peut citer AMR-WB (Adaptative Multi-Rate-WideBand) de l'UIT-T (Recommandation G.722.2), datant de 2002.

Parmi les nombreux codeurs propriétaires du marché, citons notamment les suivants :

- StreamWorks à 8,5 Kbit/s ;
- VoxWare à 2,4 Kbit/s avec le codeur RT24 ;
- Microsoft à 5,3 Kbit/s avec une utilisation partielle de la norme G.723 ;
- VocalTec à 7,7 Kbit/s.

La figure 2.8 illustre les performances des différentes normes de codeurs de la voix téléphonique en termes de qualité et de débit, en se fondant sur un échantillonnage standard à 8 kHz. L'ordonnée représente la qualité du son en réception. Il s'agit évidemment d'un critère subjectif, mais qui peut être calculé mathématiquement, comme nous le verrons. Nous avons aussi représenté les codeurs utilisés dans les réseaux de mobiles GSM et les normes régionales.

**Figure 2.8**  
*Performances des codeurs audio*



Les principales recommandations illustrées sur la figure sont les suivantes :

- G.711 : numérisation classique à 64 Kbit/s en Europe ou 56 Kbit/s en Amérique du Nord.

- G.723 : compression de la parole utilisée par de nombreux industriels, entre autres Microsoft, dans l'environnement Windows. Le débit descend à presque 5 Kbit/s.
- G.726 : compression de la parole en codage différentiel adaptatif en 16, 24, 32 ou 40 Kbit/s.
- G.727 : utilise aussi un codage différentiel, mais apporte des compléments au codage précédent. Cette recommandation indique comment changer, en cours de numérisation, le nombre de bits utilisés pour coder les échantillons. Elle est particulièrement utile dans le cadre de réseaux demandant à l'application de s'adapter à sa charge.
- G.728 : compression à 16 Kbit/s utilisant une technique de prédiction, qui consiste à coder la différence entre la valeur réelle et une valeur estimée de l'échantillon à partir des échantillons précédents. On comprend que cette différence puisse être encore plus petite que dans la technique différentielle. Si l'estimation est bonne, la valeur à transporter avoisine toujours 0. Très peu de bits sont alors nécessaires pour acheminer cette différence.
- FS : standards provenant du département de la Défense américain (DOD).
- G.723.1 : donne un débit compris entre 5,3 et 6,4 Kbit/s.
- G.729 et G.729 A : donnent un débit de 8 Kbit/s, mais la qualité de la communication est meilleure. Ces codecs ont été choisis pour compresser la voix dans l'UMTS.

Les codeurs les plus récents sont G.723.1, G.729 et G.729.A.

Le tableau 2.1 récapitule les caractéristiques de ces codeurs.

**Tableau 2.1 – Caractéristiques des principaux codeurs**

Standard	G.711	G.729 <sup>(2)</sup>	G.723.1 <sup>(4)(2)</sup>	GSM <sup>(3)</sup> 06.10 (1988)	GSM 06.60 (1996)	DOD 1016 <sup>(2)</sup>
<b>Débit (Kbit/s)</b>	64	8	6,3/5,3	13	12,2	4,8
<b>Complex. MIPS</b>	0,1	22	16/18	2,5	15,4	-
<b>Trame (ms)</b>	0,125	10	30	20	20	-
<b>Qualité MOS <sup>(1)</sup></b>	4,2	4,0	3,9/3,7	3,6/3,8	4,1	3

(1) MOS (Mean Opinion Scores)

(2) CELP (Code Excited Linear Predictive)

(3) RLP-LTP (Regular Pulse Excited with Long Term Prediction)

(4) MP-MLQ (MultiPulse-Maximum Likelihood Quantization)

Dans ce tableau, nous avons indiqué, en plus du débit qui sort du codec, la complexité du processeur (deuxième ligne) nécessaire pour effectuer les calculs lors de la décompression, qui demande généralement davantage de puissance que la compression. On voit bien que le codage classique G.711 est de loin le plus simple puisqu'il n'y a pas de compression.

La troisième ligne indique la longueur de la trame. Dans le cas de G.711, on peut considérer que cette solution n'a jamais vraiment été utilisée pour du transfert de paquets mais seulement pour de la commutation de circuits, où les octets sont envoyés immédiatement. On peut de ce fait considérer que la trame a une longueur d'un octet et que l'émission d'une trame s'effectue toutes les 125  $\mu$ s.

Pour le codage G.729, le débit de sortie moyen étant de 1 octet toutes les millisecondes, il y a en moyenne 10 octets dans la trame. Cependant, à certains instants, le débit peut être légèrement plus important. Dans ce cas, la trame de 10 ms peut transporter jusqu'à 16 octets. Dans d'autre cas, la trame peut transporter moins de 10 octets.

La dernière ligne du tableau indique la valeur MOS (Mean Opinion Score). Cette valeur correspond à l'opinion de personnes appelées à écouter les différentes sortes de voix compressées et à s'en former une opinion. Ce n'est que depuis quelques années qu'un calcul mathématique de cette valeur a pu être réalisé en ne considérant que les caractéristiques de la communication. Le MOS calculé est une norme de l'ETSI et de l'UIT-T.

## Qualité de service de la ToIP

Nous avons déjà abordé un certain nombre de caractéristiques permettant de définir la qualité de la parole téléphonique. Nous allons les approfondir dans cette section.

D'une manière générale, on retient trois facteurs pour déterminer la qualité de service d'une application de téléphonie :

- **Qualité de la transmission de la voix.** C'est la partie technique qui prend en compte le signal de départ et qui essaie de le retranscrire au mieux au niveau du récepteur.
- **Efficacité de la conversation.** C'est l'interactivité plus ou moins grande entre les deux individus en train de converser.
- **Intelligibilité de la communication.** C'est la façon dont s'expriment les individus en communication.

Ce dernier facteur ne dépend que des individus qui parlent, mais l'impact des deux premiers facteurs est important sur le troisième. Si l'intelligibilité est faible et qu'en plus la qualité de la transmission et l'efficacité de la conversation sont mauvaises, il y a de fortes chances que les paroles ne soient pas comprises.

Dans le premier critère, il faut retenir surtout la bande passante utilisée et la retranscription plus ou moins bonne en fonction des codecs et des taux de perte de paquets. Dans le deuxième, c'est le temps de réponse qui donne une conversation plus ou moins hachée. Le troisième critère dépend de la prononciation des personnes en communication. Avec une bande étroite de 3 100 Hz, il est difficile de différencier un « s » d'un « f ». Si l'on augmente à 6 950 Hz de bande passante la différenciation est beaucoup plus aisée.

Des facteurs externes sont également à prendre en compte dans la qualité perçue.

Les principaux facteurs externes sont les suivants :

- Bruit de ligne de la communication.
- Bruit corrélé au signal qui provient généralement du codec et essentiellement du choix de la quantification.
- Bruit de fond provenant de l'endroit où se trouve le micro.

Il est donc très difficile d'évaluer la qualité de la voix en dehors d'une écoute d'un utilisateur, qui est capable de prendre en compte l'ensemble des paramètres importants, d'où l'origine de la technique MOS (Mean Opinion Score).

Le type de test le plus utilisé dans cette évaluation subjective de la qualité téléphonique est le test ACR (Absolute Category Rating). Cette recommandation de l'UIT-T datant de 1996 (référence P.800) utilise une échelle notée sur 5 points, avec ou sans annotations, appelée échelle MOS.

Les valeurs de cette échelle de qualité sont récapitulées au tableau 2.2.

**Tableau 2.2 – Échelle de qualité du test MOS**

Excellente	Bonne	Correcte	Faible	Mauvaise
5	4	3	2	1

L'inconvénient de cette solution est évidemment sa subjectivité. Les organismes internationaux de normalisation se sont penchés depuis longtemps sur l'évaluation de la qualité, et différentes approches ont été proposées avant d'aboutir à un calcul mathématique du facteur MOS.

Le premier modèle proposé par l'UIT-T, le modèle E, estime la qualité globale du système grâce à des mesures instrumentales et à une description du système à l'aide de nombreux paramètres. Ce modèle provient de l'intégration de plusieurs modèles de l'ETSI dans une norme de 1996, appelée ETSI ETR 250. Cette norme a été reprise au niveau international par l'UIT-T pour définir en 2005 la qualité de la parole téléphonique dans un réseau sous la recommandation G.107.

Cette recommandation s'appuie sur le fait que les dégradations s'ajoutent les unes aux autres sur une échelle de qualité prédéterminée. Si un signal traverse plusieurs équipements, les dégradations des équipements s'additionnent.

L'échelle de qualité R peut ainsi s'exprimer sous la forme :

$$R = R_0 - I_s - I_d - I_{e,eff} + A$$

où

- $R_0$  représente le rapport signal sur bruit par rapport au point 0 dB, en prenant en compte le niveau de la voix et les différents bruits présents sur la ligne.
- $I_s$  représente la dégradation qui s'effectue sur la parole elle-même.

- Id représente la dégradation provenant du délai ou de l'écho.
- Ie,eff représente la dégradation de la qualité de la parole subie pendant la transmission au travers d'un ou de plusieurs codecs. Les pertes de paquets peuvent être prises ou non en compte dans ce paramètre.
- A représente un paramètre dont l'objectif est de déterminer les avantages à utiliser la parole numérisée. Ce paramètre a essentiellement été ajouté pour tenir compte de la mobilité des téléphones, ce qui représente un avantage important par rapport à la qualité intrinsèque de la voix et qui fait penser au client qu'une qualité plus médiocre est aussi bonne qu'une qualité meilleure mais sur un téléphone fixe.

L'échelle R est comprise entre 0 et 100, où 0 correspond à la plus mauvaise qualité et 100 à la meilleure. La valeur R de la référence de la téléphonie bande étroite (G.711 sur l'échelle) utilisée sur un canal sans bruit vaut :

$$R = 93,2$$

Pour obtenir la valeur MOS équivalente, l'UIT-T a introduit la formule :

$$\text{MOS} = \begin{cases} 1 & \text{pour : } R < 0 \\ 1 + 0,035 * R + R(R - 60)(100 - R) * 7.10^{-6} & \text{pour : } 0 < R < 100 \\ 4,5 & \text{pour : } R < 100 \end{cases}$$

De plus en plus de systèmes utilisent ce calcul pour déterminer si la parole téléphonique passe correctement ou non. Si la valeur MOS est trop faible, le système applique un contrôle sur les flux qui ne sont pas de la parole téléphonique jusqu'à ce que le MOS remonte à une valeur acceptable. De même, de nombreux réseaux testent en permanence la valeur MOS sur des paquets de contrôle afin de déterminer si le réseau est apte à véhiculer de la parole téléphonique.

## Caractéristiques du débit

Les octets qui sortent du codec donnent une première estimation de la valeur du débit, qui ne tient compte que du flux de parole. Cependant, ce qui transite dans le réseau est bien différent : il faut envelopper les octets de parole dans un paquet, un paquet IP en général, puis encapsuler le paquet IP dans une trame. De plus, il faut une signalisation pour mettre en place le mode connecté correspondant au déclenchement de la sonnerie chez le destinataire. Enfin, une signalisation peut être ajoutée afin d'ouvrir le chemin par lequel transiteront les paquets de paroles.

Le débit total est donc bien supérieur à celui de la seule voix téléphonique.

Le tableau 2.3 indique l'efficacité de la communication lorsque la paquetsation s'effectue dans un paquet IPv4, en négligeant dans un premier temps la trame utilisée.

Dans cette première évaluation, nous supposons que la zone de données est complètement remplie par des octets de parole. Cette zone de données est encapsulée dans un paquet IPv4. Les calculs s'effectuent pour différentes durées de la zone de données : 5,

10, 20 et 40 ms. Cela demande un temps de remplissage dépendant de la vitesse du codec.

Le tableau indique en pourcentage l'efficacité de la communication, c'est-à-dire la proportion d'octets téléphoniques transitant sur la voie de communication par rapport à l'ensemble des bits transmis. Suivant la structure des options d'IPv4, nous avons calculé le maximum et le minimum d'efficacité, le maximum étant obtenu lorsque le paquet IPv4 est le plus petit possible et le minimum lorsqu'il est le plus long possible.

**Tableau 2.3 – Efficacité d'une communication de ToIP (IPv4)**

Codec	Temps de remplissage de la zone de données (en milliseconde)			
	5	10	20	40
G.711	47,6 %	64,5 %	78,4 %	87,9 %
G.711	38,5 %	55,6 %	71,4 %	83,3 %
G.726	31,3 %	47,6 %	64,5 %	78,4 %
G.726	23,8 %	38,5 %	55,6 %	71,4 %
G.729	10,2 %	18,5 %	313,3 %	47,6 %
G.729	7,2 %	13,5 %	23,8 %	38,5 %

Plus le codec donne naissance à un flot compressé, plus l'efficacité est faible, puisque plus le nombre d'octets utiles dans le paquet est faible. De même, plus le temps de remplissage est faible, plus l'efficacité diminue.

Une autre façon de raisonner consiste à déterminer le flux réel d'une communication de téléphonie sur IP. Ce flux permet d'évaluer le débit des liaisons dont une entreprise a besoin pour y intégrer un système de téléphonie ou qu'un opérateur doit mettre en place en fonction du nombre de paroles téléphoniques devant transiter dans son réseau.

Le tableau 2.4 donne une première évaluation de ces débits, toujours en calculant dans chaque cas le maximum et le minimum que peut procurer IPv4.

**Tableau 2.4 – Débits réels lors d'une communication de ToIP (IPv4)**

Codec	Algorithme de codage	Débit de la parole téléphonique	Durée de remplissage de la zone de données	Débit réel
G.711	PCM	64 Kbit/s	0,125 ms	80 Kbit/s
G.723.1	ACELP	5,6 Kbit/s	30 ms	16,27 Kbit/s
G.723.1	ACELP	6,4 Kbit/s	30 ms	17,07 Kbit/s
G.726	ADPCM	32 Kbit/s	0,125 ms	48 Kbit/s
G.728	LD-CELP	16 Kbit/s	0,625 ms	32 Kbit/s
G.729(A)	CS-CELP	8 Kbit/s	10 ms	24 Kbit/s

Si l'on tient compte de la structure de la trame, c'est-à-dire des octets de supervision supplémentaires nécessaires pour transporter les informations de téléphonie, le débit nécessaire pour transporter la parole téléphonique augmente.

Le tableau 2.5 donne une idée de l'efficacité de réseaux Ethernet à 10-100 Mbit/s et 1 Gbit/s et d'un réseau ATM.

Il faut bien différencier les deux types de réseaux Ethernet, car la longueur minimale de la trame est différente : 64 octets dans le premier cas et 512 octets dans le second.

En ce qui concerne ATM, la trame est de longueur constante et contient 48 octets de données et 5 octets de supervision. On suppose dans notre calcul que les 48 octets de données transportent le paquet IP, qui, de ce fait, doit être découpé en plusieurs fragments pour être encapsulé dans des trames ATM. Nous utilisons pour ce faire, la couche AAL5 (ATM Adaptation Layer). D'autres options sont possibles dans le monde ATM, notamment AAL1 et AAL2, mais, dans ces deux cas, il n'y a pas de paquet IP encapsulé. Les octets téléphoniques sont directement mis dans la trame ATM en AAL1, et les octets de plusieurs voix téléphoniques peuvent être multiplexés dans une même trame ATM en AAL2.

Le tableau restreint les calculs aux cas où IPv4 est utilisé avec le plus d'options.

**Tableau 2.5 – Efficacité des réseaux Ethernet et ATM (IPv4)**

Codec	Temps de remplissage de la zone de données en milliseconde			
	5	10	20	40
G.711, Ethernet 10-100 Mbit/s	0,31	0,48	0,65	0,78
G.711, Ethernet 1 Gbit/s	0,08	0,16	0,31	0,62
G.711, ATM	0,25	0,91	0,60	0,91
G.726, Ethernet 10-100 Mbit/s	0,19	0,31	0,48	0,65
G.726, Ethernet 1 Gbit/s	0,04	0,08	0,16	0,31
G.726, ATM	0,13	0,25	0,55	0,60
G.729, Ethernet 10-100 Mbit/s	0,08	0,15	0,27	0,42
G.729, Ethernet 1 Gbit/s	0,02	0,03	0,06	0,12
G.729, ATM	0,08	0,16	0,30	0,30

Le tableau illustre la très mauvaise utilisation générale du support physique pour une communication de téléphonie sur IP. Il mérite cependant quelques explications, étant donné que les valeurs indiquées ne semblent pas homogènes.

Tout d'abord, les mauvais résultats de l'Ethernet GbE (1 Gbit/s) proviennent de la trame minimale, qui est de 512 octets. Même pour transporter 8 octets, comme c'est le cas avec le codeur G.729, pour une zone de données de 5 ms, il faut 512 octets de trame Ethernet. Il est évident qu'une solution pour augmenter l'efficacité consisterait à multiplexer plusieurs paroles téléphoniques dans une même trame ou à multiplexer la parole téléphonique avec d'autres applications. Ce n'est pour le moment que très peu usité, car

il n'y a aucune normalisation d'un tel multiplexage. Les techniques de cette sorte restent encore aujourd'hui propriétaires.

Les résultats sont encore plus hiératiques pour le transport par une trame ATM. En effet, les trames étant de longueur constante, si la longueur du paquet IP n'est pas un multiple de 48 octets, le paquet doit être divisé en plusieurs morceaux de 48 octets, le dernier fragment faisant moins de 48 octets. Suivant que l'on tombe sur un multiple de 48 octets et que l'on utilise plus ou moins de trames, l'efficacité varie du tout au tout.

Le tableau 2.6 indique les valeurs des flux correspondant à ces différents cas de figure.

**Tableau 2.6 – Débits réels des réseaux Ethernet et ATM (IPv4)**

Codec	Temps de remplissage de la zone de données en milliseconde			
	5	10	20	40
G.711, Ethernet 10-100 Mbit/s	206	133	74	82
G.711, Ethernet 1 Gbit/s	832	416	208	104
G.711, ATM	256	70	107	70
G.726, Ethernet 10-100 Mbit/s	168	103	67	49
G.726, Ethernet 1 Gbit/s	768	384	192	96
G.726, ATM	246	128	58	53
G.729, Ethernet 10-100 Mbit/s	100	53	30	15
G.729, Ethernet 1 Gbit/s	400	266	133	67
G.729, ATM	100	50	27	27

On remarque que la téléphonie sur IP occupe une bande passante assez importante. Il est vivement déconseillé de choisir un réseau GbE pour y effectuer de la téléphonie seule, puisque le nombre de paroles téléphoniques pouvant transiter sur une liaison est le même que pour un réseau Ethernet à 100 Mbit/s, celui-ci valant aujourd'hui cinq fois moins cher.

Pour faire diminuer le trafic réel, il faut que la zone de données du paquet soit la plus longue possible, sauf en ATM, du fait du découpage complexe des paquets dans les trames. Un compromis est à trouver entre une zone de données assez longue pour optimiser l'efficacité et le temps de paquets, qui, s'il devient trop long, sera inacceptable pour la communication.

Nous revenons sur cette problématique à la section suivante.

## Le contrôle dans la ToIP

Comme indiqué en début de chapitre, la téléphonie sur IP est une application temps réel qui n'accepte qu'un temps de réponse inférieur à 300 ms. Dans l'Internet de première génération, le réseau ne doit pas être trop chargé pour que cette contrainte soit respectée.

Dans les réseaux d'entreprise et ceux des fournisseurs d'accès à Internet et des opérateurs, le passage de la parole est possible à condition de contrôler le réseau afin que le temps total de transport, y compris la paquetsation et la dépaquetsation, soit limité.

De nombreuses solutions ont été proposées, notamment par l'IMTC (International Multimedia Teleconferencing Consortium). Il a d'abord fallu définir un codeur normalisé. Le choix s'est porté sur G.723, mais d'autres solutions sont possibles, comme le codeur G.711.

Le paquet IP doit non seulement être le plus court possible, mais il doit multiplexer plusieurs voies de parole dans un même paquet, afin de raccourcir le temps de remplissage et de limiter les temps de transfert dans le réseau. Si les routeurs peuvent gérer des priorités, ce qui est possible en utilisant des services de type DiffServ, la parole téléphonique est acheminée beaucoup plus facilement dans le laps de temps exigé.

Plusieurs organismes de normalisation de droit ou de fait travaillent sur ce sujet particulièrement prometteur. Dans les organismes de droit, l'ETSI, l'organisme de normalisation européen, a mis sur pied le groupe TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks). Le projet porte sur la parole et le fax entre utilisateurs connectés, en particulier sur des réseaux IP. Le cas où un utilisateur travaille sur un réseau IP et un autre sur un réseau à commutation de circuits, qu'il soit téléphonique, GSM ou UMTS, entre également dans le cadre des études de TIPHON.

Les activités de TIPHON concernent également la validation des solutions pour transporter la parole téléphonique par le biais de maquettes en vraie grandeur. Il s'agit d'expériences menées conjointement par l'ETSI, l'UIT-T et l'IETF mais aussi avec les groupes IMTC et VoIP (Voice over IP).

L'UIT-T travaille de son côté activement à la normalisation de la ToIP dans trois groupes du SG 16 : le WP1 pour les modems (série V), le WP2 pour les codecs (série G) et le WP3 pour les terminaux (série H). L'objectif de l'UIT-T est de développer un environnement complet, et non simplement un terminal ou un protocole.

Au sein de l'IETF, de nombreux groupes de travail s'attaquent à des problèmes spécifiques, parmi lesquels :

- AVT (Audio Video Transport), qui utilise le protocole RTP (RFC 1889 et 1890) pour les communications temps réel.
- MMUSIC (Multiparty Multimedia Session Control), qui utilise le protocole SIP, que nous introduisons plus loin.
- IPTel (IP Telephony), qui définit un protocole de localisation des passerelles et un langage permettant de mettre en communication des circuits et des flots IP.
- PINT (PSTN IP Internetworking), qui utilise également le protocole SIP.
- FAX (Fax over IP), afin de stocker et émettre des fax par l'intermédiaire de messages électroniques.

- MeGaCo (Media Gateway Control), qui détermine un protocole entre une passerelle et son contrôleur.
- SIGTRAN (Signal Translation), qui propose de passer les commandes de signalisation CCITT n° 7 dans des paquets IP.
- ENUM (E.164/IP translations), qui gère les translations d'adresses E.164 vers des adresses IP.

Le respect de la contrainte temporelle est une première priorité pour le transport de la parole téléphonique. Une seconde priorité concerne la mise en place d'une signalisation afin de mettre en connexion les deux utilisateurs qui veulent se parler.

Les protocoles de signalisation utilisés pour le transport et la gestion de la parole sous forme de paquets IP regroupent essentiellement H.323 et SIP (Session Initiation Protocol). Nous verrons en détail le protocole H.323 au chapitre suivant. Ce protocole a été défini dans un environnement de télécommunications, à la différence de SIP, qui provient du monde de l'informatique et plus spécifiquement du Web. SIP peut utiliser le protocole HTTP ainsi que la sécurité afférente. Il peut en outre s'accommoder des pare-feu. SIP met en place des sessions, qui ne sont que des appels téléphoniques entre un client et un serveur. Six primitives HTTP sont utilisées pour cela : INVITE, BYE, OPTIONS, ACK, REGISTER et CANCEL.

## Conclusion

Nous avons introduit dans ce chapitre les principales contraintes de la téléphonie sur IP. La complexité de cette dernière, fortement accrue par rapport à la téléphonie classique sur circuit, est le prix à payer pour passer à l'intégration de la téléphonie dans le monde plus vaste des données.

On peut en déduire que le coût d'installation d'un réseau de téléphonie sur IP est relativement important puisqu'il faut mettre en place tout un nouvel environnement, incluant des terminaux de type téléphone IP ou PC, un système de signalisation pour mettre en place les connexions et un contrôle du réseau pour que les temps de réponse restent faibles. La rentabilité d'un tel environnement n'est possible que sur plusieurs années.

D'autres contraintes, telles que la sécurité, la disponibilité ou l'utilisation des techniques P2P, sont abordées dans la suite de l'ouvrage.

## La signalisation H.323

La signalisation désigne la transmission d'un ensemble de signaux et d'informations de contrôle échangés entre les intervenants d'une communication. Ces intervenants peuvent être des entités en bout de liaison (terminaux) ou des entités intermédiaires de contrôle et de gestion des communications. Leurs échanges permettent l'initiation, la négociation, l'établissement, le maintien et la fermeture de la connexion.

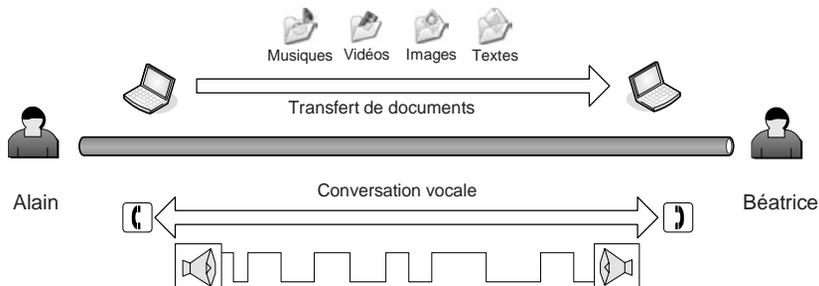
Il convient de distinguer deux types de transferts pour comprendre à quoi correspond la signalisation :

- le transfert de données brutes ;
- le transfert d'informations de contrôle.

Le transfert de données brutes concerne les échanges de données binaires d'un poste vers un autre. L'objectif de ce transfert est de reproduire à l'identique des données en les faisant transiter par un réseau. Par exemple, deux correspondants peuvent s'échanger un fichier audio MP3 ou des images bitmap, comme à la figure 3.1, où l'utilisateur Alain envoie des données vers le poste de Béatrice.

**Figure 3.1**

*Transfert de données brutes*



De la même façon, si l'on considère une conversation téléphonique en cours, les intervenants produisent des sons qui doivent être recomposés et diffusés chez leurs correspondants.

Dans tous ces cas, seul l'envoi des données a de l'importance, ce qui relève d'un transport d'informations.

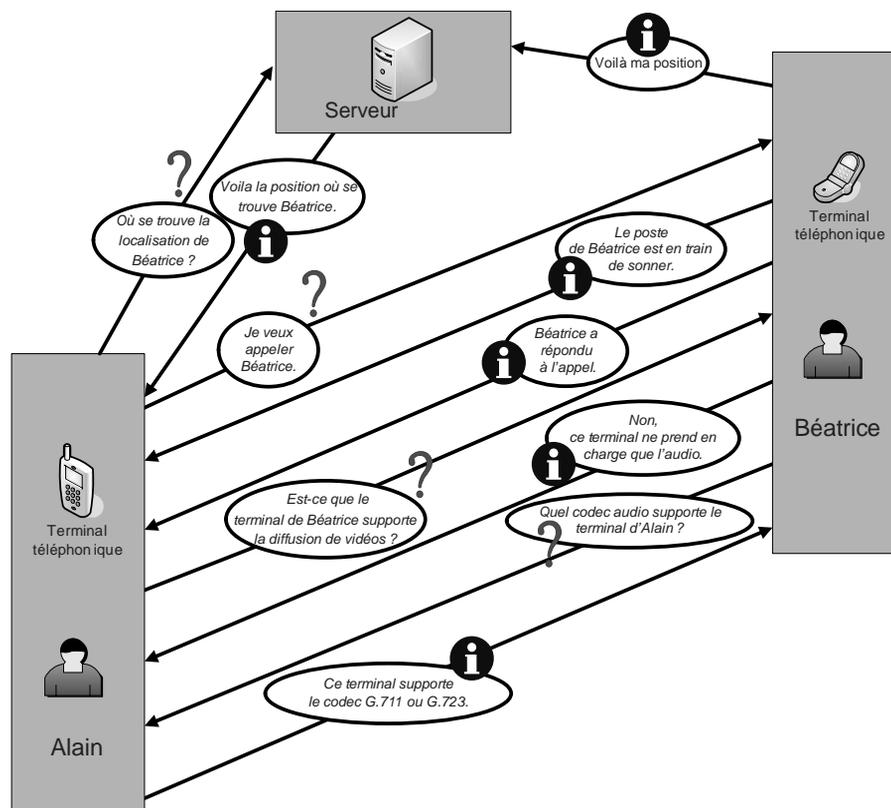
Le transfert d'informations de contrôle concerne les échanges de type protocolaire exécutant une action prédéfinie, et donc nécessairement limitée en possibilités. L'objectif de ce transfert est d'assurer la maîtrise et la gestion du flux.

Dans le cas typique d'une application de téléphonie, lorsqu'une personne en appelle une autre, elle n'a initialement pas de « données » à lui transmettre, mais veut simplement être mise en relation avec son correspondant. Cette mise en relation nécessite d'abord de localiser l'appelé, puis de faire sonner son poste, afin de lui signaler l'appel. Pour la localisation comme pour l'avertissement d'appel, on parle de signalisation.

La figure 3.2 illustre quelques exemples de messages de signalisation transportant des requêtes et réponses à caractère descriptif.

Figure 3.2

Transferts d'informations de contrôle (signalisation)



De la même manière, lorsque la sonnerie d'appel retentit dans le terminal appelé, l'appelant en est immédiatement informé par une tonalité particulière sur son terminal téléphonique. Il s'agit là aussi d'une information de signalisation.

Si un correspondant ne répond pas à un appel, il est probable que sa messagerie téléphonique va s'enclencher. Cette redirection d'appel du poste appelé vers sa messagerie est également une information de signalisation. Elle ne transporte aucune information de données brutes, mais vise à signaler à l'appelant que l'appelé n'est pas disponible et que sa messagerie est opérationnelle.

Ces deux catégories de transfert sont liées : ce n'est que lorsque l'appelé a répondu à l'appel que commence le transfert d'informations brutes, c'est-à-dire le transport de la voix, qui doit être fidèlement retransmise d'un correspondant à l'autre. La signalisation n'est que l'étape préalable qui a mis en place la connexion entre les différents utilisateurs pour permettre la communication.

Dans le modèle OSI, la signalisation téléphonique correspond à une fonctionnalité de niveau 7 (couche applicative). Elle n'est donc jamais assurée par les entités réseau de routage pur, comme les routeurs et commutateurs, qui fonctionnent à des couches inférieures. Des entités dédiées sont exploitées à ces fins : il s'agit de serveurs au niveau du cœur du réseau et des terminaux (téléphone, ordinateur ou PDA, par exemple) en bordure de réseau, au niveau de l'utilisateur.

Pour être comprise et correctement interprétée de l'ensemble des entités participant aux mécanismes de signalisation, celle-ci doit respecter une syntaxe particulière. C'est tout l'objet de la spécification d'un protocole de signalisation.

Le protocole H.323 figure parmi les plus réputés des protocoles de signalisation pour la téléphonie sur IP. H.323 n'est en réalité que la référence du protocole. Son nom complet est *Packet-based Multimedia Communications Systems*, ou « Systèmes de communication multimédia fonctionnant en mode paquet ». Comme ce nom l'indique, il peut être utilisé pour tous les réseaux à commutation de paquets, en particulier IP.

Ce protocole est spécifié pour le traitement de la signalisation des données multimédias avec de fortes contraintes temporelles, comme la voix ou la vidéo, mais aussi la réalité virtuelle ou les jeux en réseau.

Ce chapitre se propose de faire un tour d'horizon complet du protocole H.323. Réputé complexe, ce protocole demeure cependant l'un des plus exploités.

## Protocoles et normalisation

1996 a été une année charnière et un tournant particulièrement important pour l'essor de la ToIP. Si cette dernière suscitait de l'intérêt auparavant, ce n'est qu'à partir de cette année-là qu'elle commença à prendre son envol en revêtant un caractère normalisé.

Le handicap majeur qui freinait jusqu'alors le développement de la téléphonie IP résidait dans l'incompatibilité des protocoles utilisés pour mettre en œuvre une communication

audio ou vidéo. Lorsque deux opérateurs distincts utilisent des normes de communication différentes, il est impossible pour un utilisateur exploitant le réseau d'un opérateur de communiquer avec un utilisateur affilié à l'autre opérateur. Pour assurer la compatibilité des communications, il était indispensable de se fonder sur des bases communes.

Pendant longtemps, un très grand nombre de constructeurs ont tenté d'imposer leur propre protocole comme standard. Aucun de ces protocoles propriétaires, le plus souvent coûteux et sans légitimité particulière vis-à-vis des autres, n'a réussi à s'imposer. Chaque constructeur cultivait ainsi sa différence, se montrant réfractaire aux autres propositions et cherchant sans cesse à améliorer le sien.

De leur côté, les entreprises se montraient sceptiques et plus que réservées à l'idée d'installer des réseaux de téléphonie fondés sur des protocoles instables. Du reste, l'utilisation d'Internet demeurait encore modeste à l'époque, et l'on ne parlait, dans le meilleur des cas, que d'exploiter la téléphonie IP dans un réseau local.

En 1996, l'éditeur de logiciel Netscape, qui possédait une part de marché de 80 % avec son navigateur Web Navigator, annonça la sortie prochaine de son logiciel de téléphonie CoolTalk. Netscape nourrit l'attente, mais il ne put tenir la vedette. La conception d'un protocole fédérateur devait passer par une institution, pas par une entreprise.

Cette même année 1996, l'ITU (International Telecommunications Union) proposa la famille de protocoles H.32x, très fortement soutenu par Microsoft et Intel. L'ITU parvint rapidement à convaincre les différents équipementiers et fournisseurs de services de la nécessité d'adopter pour norme commune ces protocoles H.32x.

Sans être précurseurs ni de la téléphonie, ni de la vidéo, ni même de la conférence, ces protocoles constituent immanquablement l'initiative la plus aboutie et la plus marquante des débuts de la signalisation multimédia. La généralisation progressive et systématique de H.323 finit par faire céder les plus récalcitrants des acteurs du multimédia, qui abandonnèrent leurs solutions propriétaires, pourtant très évoluées. La ToIP venait de trouver son protocole fédérateur et pouvait prendre son envol.

Depuis, le protocole H.323 a été adopté, implémenté ou supporté par de très nombreux industriels, à commencer par Cisco, IBM, Intel, Microsoft et Netscape.

## ***La normalisation UIT***

Fondée en 1865, l'ITU, en français UIT (Union internationale des télécommunications), est une des organisations internationales de normalisation les plus anciennes. Initialement, la lettre T désignait le télégraphe, et ce n'est qu'en 1932 qu'elle en vint à incarner le téléphone.

Installée à Genève, l'UIT dépend de l'ONU depuis 1947. Son rôle est de proposer des modèles de communication afin de favoriser les télécommunications et les services associés, tout en réglementant au niveau mondial les usages des protocoles. C'est notamment à cet organisme que l'on doit la norme HD (haute définition), utilisée pour la diffusion cinématographique.

L'organisation comporte les trois comités suivants, dont les noms ont été modifiés en 1993 à fins d'unification :

- UIT-T, pôle de standardisation des télécommunications. Ce comité normalise tout ce qui a trait aux transmissions, au transport et aux télécommunications. Il reprend l'activité de l'ancien CCITT (Comité consultatif international des télégraphes et des téléphones).
- UIT-R, pôle des radiocommunications. Ce comité normalise tout ce qui a trait aux signaux vidéo et télévisuels, avec la radio analogique et numérique. Il reprend l'activité de l'ancien CCIR (Comité consultatif international des radiocommunications).
- UIT-D, pôle de développement des télécommunications. Ce comité est chargé de promouvoir l'assistance technique vers les pays en voie de développement. Il reprend l'activité de l'ancien BDT (Bureau de développement des télécommunications).

L'UIT-T a notamment défini des standards classés et référencés selon une codification particulière. La lettre indique la série (de la première à la dernière lettre de l'alphabet) et est suivie d'un chiffre identifiant chacune des recommandations de la série.

On retiendra notamment les séries de recommandations suivantes :

- série E pour les recommandations liées aux généralités des réseaux, des services et des opérations, par exemple E.164 pour le plan de numérotation de la téléphonie publique internationale ;
- série G pour les recommandations liées aux systèmes et supports de transmission multimédia, par exemple G.711 pour le codage audio avec compression ;
- série H pour les recommandations liées aux systèmes audiovisuels et multimédias, par exemple H.323 pour les systèmes multimédias fonctionnant en mode paquet ;
- série Q pour les recommandations liées aux commutations et aux signaux, par exemple Q.931 pour les réseaux RNIS ;
- série V pour les recommandations liées aux communications sur un réseau téléphonique commuté, par exemple V.90 pour les communications avec des modems à 56 Kbit/s en lien descendant et 33,6 Kbit/s en lien montant ;
- série X pour les recommandations liées aux réseaux de données et aux communications entre systèmes ouverts, par exemple X.25 pour les communications en mode point-à-point par commutation de paquets.

### ***Normes d'interopérabilité***

Pour garantir le respect de la norme et vérifier l'interopérabilité des plates-formes développées par les industriels, plusieurs organismes ont été mis en place. Ils jouent un rôle d'intermédiaire entre les spécifications abondantes des industriels et celles des concepteurs de la norme.

Les deux organismes de ce type parmi les plus importants sont les suivants :

- TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks). Fondé par l'ETSI, ce forum permet aux acteurs de vérifier la conformité et l'interopérabilité de leurs spécifications avec la norme H.323 à partir d'une plate-forme de test, appelée TIPHON-Net mise à leur disposition.
- iNOW! Consortium (Interoperability Now!). À destination des industriels, ce consortium définit les spécifications nécessaires à l'interopérabilité des entités et traite de différents aspects, tels que la sécurité ou la facturation. Créé à l'initiative de VocalTec et Lucent en 1998, il a ensuite été rejoint par d'autres acteurs de renom, notamment Alcatel, Cisco, Siemens et Ericsson. Un label iNow! est attribué aux matériels compatibles avec l'ensemble de leurs spécifications.

Tous deux continuent aujourd'hui d'œuvrer à ces fins.

### ***Les six versions de H.323***

Les premiers travaux sur H.323 ont débuté en mai 1995. Depuis lors, six versions standardisées se sont succédées, apportant leurs lots de nouveautés et d'améliorations.

Le protocole H.323 impose une compatibilité ascendante, ce qui veut dire que les fonctionnalités et méthodes présentes dans les premières versions du protocole restent supportées dans toutes celles qui suivent. Cette section présente les évolutions du protocole au cours du temps.

#### **H.323 version 1 (mai 1996)**

Initialement prévue dans le cadre très restreint des réseaux locaux (LAN) n'apportant aucune garantie de qualité de service, la version 1 de la recommandation H.323 de l'UIT-T prit le nom de « Systèmes et équipements visiophoniques pour réseaux locaux offrant une qualité de service non garantie ».

Il faudra attendre les versions suivantes pour qu'elle soit renommée « Système de communication multimédia fonctionnant en mode paquet ». Tout porte à croire que ses concepteurs n'avaient pas imaginé rencontrer un tel succès auprès des industriels, et que le protocole a ensuite évolué pour adresser des réseaux plus étendus, de type Internet.

Cette version balbutiante présentait de sévères limitations, notamment des performances, illustrées, par exemple, par la lenteur de la mise en place d'une communication ou la sécurité, totalement absente. Surtout, la spécification était imprécise quant à la manière d'implémenter le protocole, ce qui entraîna d'importants problèmes d'interopérabilité entre les différents constructeurs.

#### **H.323 version 2 (janvier 1998)**

Remarquable à bien des égards, la version 2 améliora considérablement un protocole encore instable et perfectible, en particulier les délais d'établissement d'une communication grâce à la procédure de FastConnect, qui permettait de paralléliser les annonces.

Une autre nouveauté fut incarnée par la procédure H.245 tunneling, qui permettait d'encapsuler des messages H.245 dans des messages H.225.0 (Q.931). De nouveaux services étaient supportés par le protocole, dont les classiques services de renvoi et de transfert d'appel, et des mécanismes de sécurité étaient rassemblés dans la spécification H.235. Cette dernière couvrait la plupart des mécanismes de sécurité, incluant l'authentification et le cryptage des flux de données.

Des paramètres de gestion de la qualité de service étaient ajoutés dans les messages de signalisation, permettant de s'intégrer dans une architecture de type DiffServ ou RSVP, par exemple. Néanmoins, il convient de noter que la gestion elle-même de la qualité de service ne faisait pas partie du protocole, H.323 n'offrant aucune garantie de réservation de ressources. Seuls des paramètres de qualité de service pouvaient être ajoutés dans la structure des paquets et pouvaient être utilisés par les terminaux. Au niveau du réseau cœur, ces paramètres devaient être exploités et traités par des mécanismes externes au protocole.

Plus généralement, la manière d'ajouter des services supplémentaires était décrite dans le document H.450.1, qui définissait une plate-forme générique.

Les documents suivants, numérotés H.450.x (avec  $x > 1$ ) décrivent de tels services :

- H.450.2 détaille le service de transfert d'appel, qui transforme une communication entre deux postes A et B en une communication entre A et un autre poste, C. Elle est classiquement utilisée dans les entreprises pour mettre l'appelant en relation avec la personne souhaitée.
- H.450.3 détaille le service de redirection d'appel, qui remplace un poste appelé par un autre, avec ou sans condition. Par exemple, au bout de sept sonneries sans réponse sur le poste d'Alice, tous les appels sont redirigés vers le poste de Bertrand ou bien tous les appels sans condition sur le poste d'Alice sont redirigés vers le poste de Bertrand.

Grâce au support du DTMF (Dual-Tone Multi-Frequency), le protocole H.323 version 2 permettait la création de nouveaux services vocaux. Au contraire de la téléphonie par impulsion, les codes DTMF correspondent à des fréquences. En assignant à chaque touche du terminal un code DTMF unique, il devenait possible à un serveur d'interpréter les saisies de l'utilisateur appelant et de lui fournir un service adéquat en retour. Auparavant, les messages de signalisation ne transmettaient qu'une partie de ces informations DTMF, ce qui ne permettait pas d'interpréter pleinement ces signaux.

Enfin, la recommandation H.323 permettait d'utiliser des alias à la place des adresses IP afin d'identifier les utilisateurs. Ces alias respectent le format des URL traditionnellement utilisées pour désigner une ressource unique sur Internet.

### **H.323 version 3 (septembre 1999)**

Globalement, la version 3 de H.323 apporte moins de nouveautés fondamentales que la précédente. Si la version 2 corrigeait en profondeur plusieurs imperfections de la norme initiale, la 3 contribuait à l'amélioration du protocole, sans le bouleverser en profondeur.

Retenons notamment les trois améliorations suivantes de cette version :

- Gestion de nouveaux services complétant la gamme existante, tels que les suivants :
  - CLIP (Connected Line Identification Presentation), ou présentation de l'identification de l'appel, aussi connu par son appellation commerciale d'affichage du numéro, qui permet à l'appelé de connaître le numéro d'appel de l'appelant.
  - CLIR (Connected Line Identification Restriction), ou restriction de l'identification de l'appel, plus connu sous son appellation commerciale de masquage du numéro, qui permet à l'appelant de limiter les possibilités d'identification de son numéro.
- Ajout de services destinés à compléter la série H.450.x, tels que la mise en attente ou la notification d'appel ou de message en attente.
- Intégration avec la signalisation SS7, utilisée classiquement dans les réseaux téléphoniques commutés.

L'annexe E de la norme prévoyait l'utilisation du protocole de transport UDP au lieu de TCP, les deux protocoles pouvant être utilisés au choix.

### H.323 version 4 (novembre 2000)

La version 4 axait ses développements sur la robustesse, à la fois en termes de passage à l'échelle (scalabilité), de flexibilité et de fiabilité. Le protocole confirmait ainsi sa suprématie par une technologie solide et véritablement en phase avec les besoins et les usages de tous types, y compris professionnels. La recommandation proposait pour cela des changements radicaux par rapport aux versions précédentes.

Afin d'obtenir un cadre de développement stable à la norme, cette version 4 proposait de formaliser les améliorations sous forme d'extensions au protocole, mais sans modifier ses fondations. Autrement dit, les améliorations ne remettaient plus en cause le principe de fonctionnement du protocole mais se présentaient sous la forme de modules génériques, appelées GEF (Generic Extensibility Framework).

Le protocole H.323 devenait de la sorte stable, tout en autorisant des enrichissements progressifs. Il offrait en outre un bon niveau de souplesse puisque les équipementiers étaient libres d'implémenter certaines extensions des GEF et pas d'autres, tout en restant compatibles avec le socle du standard. De fait, le protocole atteignait une certaine maturité, et ses acteurs n'étaient plus obligés de suivre en permanence les évolutions et de mettre à jour la norme pour garantir la compatibilité.

La notion de *gatekeeper alternative*, permettant le basculement des appels en cas de panne d'un gatekeeper, ou « garde-barrière », était explicitée dans le document. À cette fin, l'annexe R proposait des mécanismes permettant de modifier dynamiquement le routage des appels en cas de panne. Nous reviendrons plus loin dans ce chapitre sur les fonctionnalités évoluées de ce mécanisme.

Dans cette version, le protocole H.323 se rapprochait du protocole MGCP (Media Gateway Control Protocol), dont les travaux étaient menés en parallèle. Il offrait en effet une nouvelle conception architecturale, qui décomposait l'équipement de passerelle

originale, jugé trop lourd, en deux sous-parties. Cette nouvelle répartition reprenait le modèle proposé conjointement par le groupe de travail numéro 16 de l'UIT-T et le groupe de travail MeGaCo de l'IETF. L'UIT en proposera une nouvelle recommandation, numérotée H.248, que nous détaillons ultérieurement dans ce chapitre.

Le protocole RTP (Real-time Transport Protocol) permet de séparer les flux audio et vidéo, ce qui offre aux récepteurs une plus grande flexibilité en leur permettant de choisir indifféremment de recevoir l'un ou l'autre, avec un système de priorité. L'inconvénient de ce système est que le récepteur doit synchroniser les deux flux pour transmettre de façon parfaitement homogène la diffusion du son avec la vidéo en simultané. Cela suppose des capacités complémentaires, à la fois de l'émetteur, qui sépare la voix de la vidéo, et du récepteur, qui assure la synchronisation des deux flux.

Avec la version 4 de H.323, le protocole proposait une solution de rechange facultative permettant de multiplexer la voix et la vidéo dans un même flux, de manière que l'émetteur n'ait plus à se soucier de la synchronisation de la vidéo par rapport à la voix et qu'il puisse jouer les données sans que des décalages du son et de l'image soient perceptibles.

En plus de proposer la gestion de nouveaux services, la version 4 permettait la mobilité de l'utilisateur et l'intégration avec les réseaux GSM et UTM. En outre, le concept « d'enregistrements additionnels » donnait aux utilisateurs la possibilité de s'enregistrer plusieurs fois auprès des gatekeepers avec plusieurs pseudonymes différents. Les paquets UDP étant trop courts pour permettre de spécifier dans une même requête tous les pseudonymes à enregistrer, ce mécanisme d'enregistrements additionnels permettait de générer à la suite plusieurs courtes requêtes venant compléter les précédents enregistrements.

L'adressage H.323 était fixé, et une URL H.323 pouvait désormais prendre la forme *h323:utilisateur@domaine*, où le préfixe *h323* spécifiait qu'il s'agissait d'une adresse à interpréter par le protocole H.323, la partie *utilisateur* était un identifiant de l'utilisateur (ou éventuellement d'un service) et la partie *domaine* désignait l'entité capable de traduire cette URL, classiquement le gatekeeper susceptible de prendre en charge la résolution de cette adresse. La façon de résoudre effectivement cette adresse ne sera donnée que dans la version suivante.

### H.323 version 5 (juillet 2003)

Cette version est mineure par rapport aux précédentes. On peut la considérer comme une version de maintenance, qui répondait à un certain nombre de demandes et besoins.

Reprenant la philosophie de stabilité initiée par la version 4, avec le cadre générique des GEF, la version 5 proposait des améliorations, avec la série de recommandations H.460.x, dont le tout premier document, H.460.1, expliquait ce nouveau dispositif. La recommandation H.460.9 permettait quant à elle aux terminaux de fournir les statistiques RTCP.

L'annexe O expliquait comment utiliser les serveurs de domaines DNS (Domain Name Server) pour effectuer les résolutions de noms des adresses (URL) utilisées dans les identifications des abonnés H.323. L'interrogation des serveurs DNS pouvait

s'effectuer selon différents procédés, tel ENUM (tElephone NUmber Mapping), qui associe un nom identifiant un utilisateur (par exemple l'utilisateur dont l'identifiant est *albert@exemple.com*) avec un numéro de téléphone conventionnel (au format à dix chiffres, comme 0102030405), ou A Record (Address Record), qui associe un nom d'utilisateur avec une adresse IP (192.168.1.15 pour une adresse en réseau local, par exemple).

La version 5 gérait le protocole SCTP (Stream Control Transmission Protocol) comme solution de rechange aux protocoles de transport TCP et UDP.

### H.323 version 6 (juin 2006)

Au centre de cette dernière mouture, on retrouve une philosophie modulaire, avec de multiples perfectionnements et des procédures simplifiées et épurées, destinées à rendre H.323 encore plus accessible. Quelques améliorations sont aussi proposées, comme des supports plus larges, par exemple, de codecs (GSM, iLBC et H.264 sont pris en charge) ou de spécifications de QoS (H.361 notamment).

Le concept de *gatekeeper affectée*, imposant un gatekeeper fixe à un terminal, complète celui de *gatekeeper alternatif* offert depuis la version 4. Nous reviendrons sur ce mécanisme en fin de chapitre.

En termes de sécurité, les mécanismes sont complètement refondus. Le document de référence H.235 est restructuré et décomposé en plusieurs recommandations, numérotées de H.235.0 à H.235.9.

Les recommandations H.460.17, H.460.18 et H.460.18 répondent au problème de la traversée des réseaux avec translation d'adresse IP, ou NAT (Network Address Translation) et des filtres pare-feu (firewalls), qui pénalisait le protocole H.323. Pendant longtemps, les communications H.323 ne pouvaient en effet être mises en place dans les entreprises utilisant un plan d'adressage privé et des solutions de pare-feu, car le protocole H.323 utilise des ports dynamiques qui ne sont généralement pas supportés par les pare-feu ordinaires.

La translation des adresses privées et logiciels des pare-feu est une solution déployée aujourd'hui presque systématiquement dans les entreprises, ainsi que bien souvent chez les particuliers. Si certains pare-feu perfectionnés et onéreux proposent des méthodes propriétaires pour permettre aux flux H.323 d'être filtrés correctement, la solution générale n'est véritablement donnée que dans ces nouvelles recommandations H.460.

Ces dernières spécifient les procédures à implémenter dans les gatekeepers et les terminaux pour passer les translations d'adresses et traverser les pare-feu. Le principe de ces procédures est de conserver une connexion persistante TCP entre les terminaux et le gatekeeper pour assurer les communications.

Une nouvelle entité est introduite pour permettre aux terminaux n'implémentant pas encore les procédures de la version 6 de H.323 de traverser quand même les réseaux nattés et filtrés. Il s'agit en ce cas d'un proxy particulier auquel s'adressent

les terminaux et qui agit comme un intermédiaire pour relayer les messages de signalisation vers leur destinataire. En quelque sorte, si les terminaux n'arrivent pas à joindre leurs correspondants parce que leurs flux sont difficilement interprétés, le proxy interprète et reformate les flux avant de les envoyer vers leur destinataire. Ces derniers utilisent eux aussi le proxy afin que leurs flux soient conformes à ce qu'attendent les émetteurs.

## Architecture et fonctionnalités du protocole H.323

Le protocole H.323 s'articule autour d'une architecture particulière décrite dans ce qui suit. Cette architecture concentre les fonctionnalités autour d'entités, ce qui explique pourquoi le protocole H.323 est considéré comme fortement centralisé.

Nous allons définir et détailler chacune des entités introduites par le protocole H.323.

### *Les quatre entités d'une architecture H.323*

Le protocole H.323 axe très fortement ses communications sur une typologie d'équipements.

La terminologie anglaise étant couramment employée dans les documentations françaises, il convient de la connaître. Dans ce qui suit, les premiers termes donnés peuvent être considérés comme les plus courants.

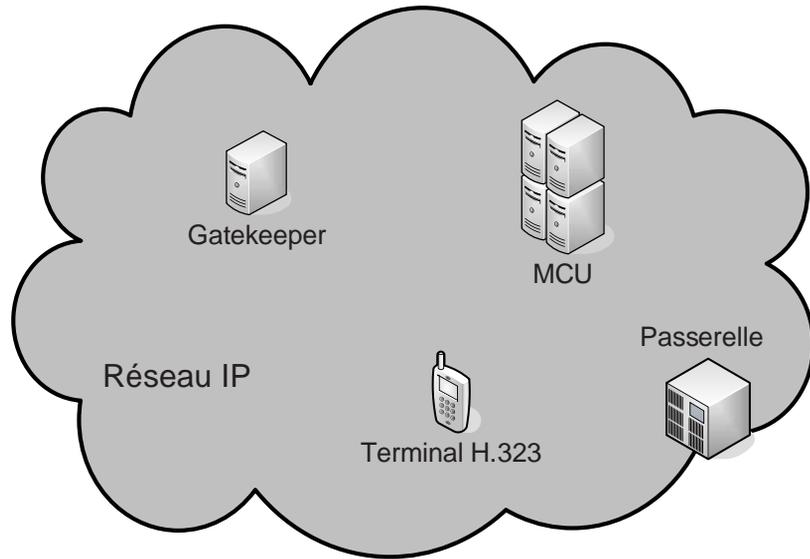
Une architecture H.323 est généralement composée des quatre catégories d'entités suivantes :

- **Terminaux** (au minimum deux). Ce sont les équipements de traitement destinés aux utilisateurs, leur permettant d'émettre et de recevoir des appels. Deux terminaux doivent au minimum être présents pour qu'une communication ait lieu.
- **Gatekeeper**, ou garde-barrière. C'est l'équipement permettant la localisation des utilisateurs. Ces derniers peuvent s'identifier entre eux par des noms, auxquels il faut attribuer l'adresse IP correspondante dans le réseau ou, si l'appelé n'est pas situé dans un réseau IP, la localisation de l'entité intermédiaire à joindre pour l'appel. Outre cette fonction primordiale, un gatekeeper remplit tout un ensemble de fonctions complémentaires de gestion et de contrôle des communications, certaines étant indispensables et d'autres facultatives.
- **Passerelle**, ou gateway. C'est l'équipement permettant à des utilisateurs du réseau IP de joindre les utilisateurs qui sont actifs sur d'autres types de réseaux téléphoniques, RTC, RNIS ou ATM. On peut avoir autant de passerelles différentes que nécessaire, suivant la nature des réseaux non-IP à interconnecter.
- **MCU** (Multipoint Control Unit), ou unité de contrôle multipoint, parfois appelée pont multipoint. C'est l'équipement permettant la gestion des conférences, c'est-à-dire les communications multimédias mettant en jeu plus de deux interlocuteurs. Ces derniers doivent préalablement se connecter à la MCU, sur laquelle s'établissent les demandes et négociations des paramètres à utiliser lors de la conférence.

Ces quatre entités sont illustrées à la figure 3.3.

**Figure 3.3**

*Architecture de H.323*



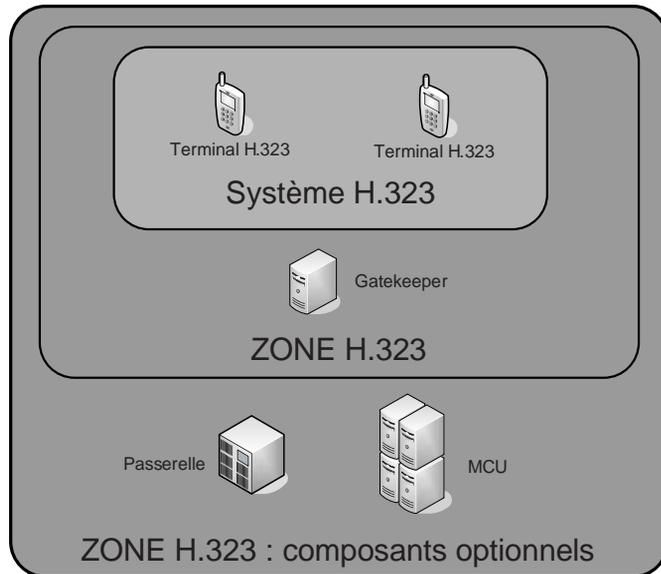
Avant de détailler chacune de ces entités, les deux définitions suivantes doivent être connues :

- **Points de terminaison.** Terminaux, gateway et MCU sont des entités auxquelles les émetteurs peuvent s'adresser directement pour communiquer. Contrairement au gatekeeper, qui joue un rôle intermédiaire de contrôle et de gestion, ces entités sont des points de terminaison des appels (aussi appelés endpoints).
- **Zone et système H.323.** La nomenclature H.323 définit deux notions qu'il convient de bien connaître et différencier :
  - Un système H.323 est défini comme un ensemble de deux terminaux au minimum, d'autres éléments pouvant être ajoutés.
  - Une zone H.323 est un ensemble de deux terminaux avec un gatekeeper au minimum, d'autres éléments pouvant être ajoutés.

Autrement dit, une zone H.323 est un système H.323 associé à un gatekeeper et éventuellement, mais pas nécessairement, à des entités additionnelles, comme une MCU ou une passerelle. Chaque entité peut être présente en grand nombre.

La figure 3.4 illustre ces notions de manière hiérarchique. Système et zone correspondent à des considérations logiques. Cela signifie que plusieurs réseaux locaux peuvent être regroupés dans une même zone H.323 et dépendre d'un même gatekeeper. À l'inverse, il est possible d'avoir plusieurs zones H.323 et de les faire communiquer entre elles.

**Figure 3.4**  
*Système et zones H.323*



### ***Le terminal H.323, équipement des interlocuteurs***

Équipement de base des interlocuteurs, le terminal peut prendre la forme d'un téléphone IP, en apparence semblable à n'importe quel autre appareil téléphonique utilisé dans la téléphonie RTC, ou d'un logiciel téléphonique installé sur un ordinateur ou un assistant personnel de type PDA équipé d'un micro et d'une sortie audio. On parle en ce cas de softphone.

#### **Prérequis fonctionnels des terminaux H.323**

Pour qu'un terminal soit de type H.323, il doit respecter les prérequis fonctionnels suivants :

- Support des protocoles H.225.0 et H.245 (obligatoire). Ces protocoles, dont le premier utilise des protocoles hérités du RNIS, avec Q.931 et RAS, ont à leur charge d'effectuer la partie signalisation proprement dite dans un système H.323. C'est pourquoi leur gestion est requise par les terminaux. Ces protocoles sont détaillés dans la suite du chapitre.
- Support des protocoles RTP/RTCP (obligatoire). Une fois la liaison établie entre les interlocuteurs, la session multimédia peut commencer. Le transport des données recourt au protocole RTP, auquel est associé le protocole RTCP afin que l'application téléphonique H.323 utilisée dans le terminal puisse réguler son débit selon l'état du réseau. Ces deux protocoles sont donc aussi nécessaires au terminal H.323.
- Support du codec G.711 (obligatoire). Un terminal H.323 doit être capable de gérer l'audio et, suivant les usages, les textes, images et éventuellement vidéos. Pour cela, il doit nécessairement supporter au moins le codec audio G.711, selon l'une de ces deux variantes : PCM (Pulse Code Modulation), la loi mu utilisée en Amérique du Nord et

en Asie, et MIC (modulation, impulsion et codage), la loi A utilisée en Europe et dans le reste du monde. Le support des autres codecs audio et de l'ensemble des codecs vidéo est laissé libre et optionnel dans la spécification du protocole H.323.

- Support de liaisons asymétriques (optionnel). Les terminaux peuvent être disposés de façon à établir des communications asymétriques, pour lesquelles la réception de données se fait avec un codec différent de celui utilisé pour l'envoi. Par exemple, un même terminal peut utiliser le codec G.222 en réception et le codec G.711 en émission. Cela permet d'affiner les débits selon les capacités des terminaux.

À titre d'exemple, considérons deux terminaux A et B, dont le premier a un débit descendant (réception ou download) fort mais un débit montant (envoi ou upload) faible, et le second un débit montant et descendant fort. Le terminal A peut utiliser un codec de très bonne qualité pour la réception et un codec de moins bonne qualité pour l'envoi. Parallèlement, le terminal B doit s'adapter aux capacités de son correspondant en utilisant le même codec de bonne qualité pour l'envoi et le même codec de moins bonne qualité pour la réception. Les liaisons sont de la sorte asymétriques.

- Support du multicast (optionnel). Si le terminal doit servir à la mise en place de conférences, le multicast doit être géré par le terminal. Il permet de dialoguer sans l'intervention d'une entité spécialisée, telle qu'une MCU, en diffusant ses messages dans le réseau, sous réserve que ce dernier dispose de routeurs qui autorisent la diffusion en multicast.

Comme l'illustre la figure 3.5, des terminaux peuvent parfaitement communiquer entre eux en utilisant le protocole H.323 et sans l'intervention d'autres éléments architecturaux. Ils forment ainsi un système H.323 autonome, mais leurs communications ne peuvent profiter de la gamme de services fournie par les autres entités. En particulier, les utilisateurs doivent impérativement connaître l'adresse IP de leur correspondant pour pouvoir les joindre. En outre, ils restent cloisonnés dans un réseau purement IP, ce qui représente une contrainte très limitative pour H.323, qui vise à offrir une large communication entre différents types de réseaux.

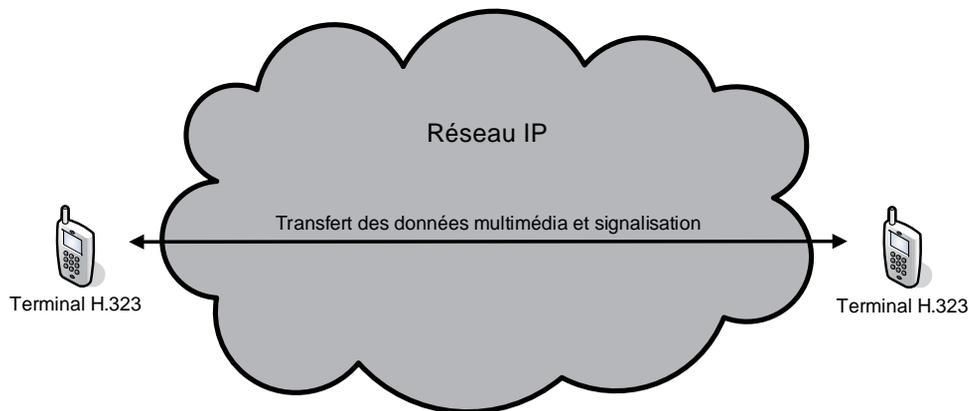


Figure 3.5

Communication entre deux terminaux H.323

## Le gatekeeper, point de contrôle et de gestion

Facultatif de manière générale, le gatekeeper est requis pour toutes les opérations de contrôle et de gestion des communications. Il offre de la valeur ajoutée aux communications en proposant plusieurs fonctions, dont la première consiste à assurer la localisation des abonnés. Progressivement, le gatekeeper est devenu un élément central dans lequel se concentrent toutes les fonctionnalités additionnelles, offrant une gamme de services complémentaires. L'architecture de H.323 est donc fortement centralisée autour de lui.

Si un gatekeeper est présent dans une zone H.323, tous les terminaux doivent nécessairement s'y enregistrer et y solliciter l'autorisation d'effectuer des appels, en émission comme en réception.

### Localisation des abonnés

Pour permettre la localisation des utilisateurs dans un réseau IP utilisant H.323, le gatekeeper effectue la conversion d'un alias en une adresse IP.

Un alias est un identifiant associé à un utilisateur. Chaque utilisateur est localisé dans le réseau IP par une adresse IP, mais cette adresse peut être attribuée dynamiquement. Pour être joignables, les utilisateurs ne sont pas identifiés par cette adresse IP, qui est impropre à les qualifier pleinement et univoquement, mais par un alias qui les représente et que les utilisateurs peuvent s'échanger pour se contacter.

Le gatekeeper se charge d'effectuer la correspondance entre les alias et les adresses IP. Les utilisateurs qui ne sont pas situés dans un réseau IP doivent aussi pouvoir être joignables par les utilisateurs du réseau IP. C'est à nouveau le gatekeeper qui permet de les localiser.

Un alias peut être défini de plusieurs façons :

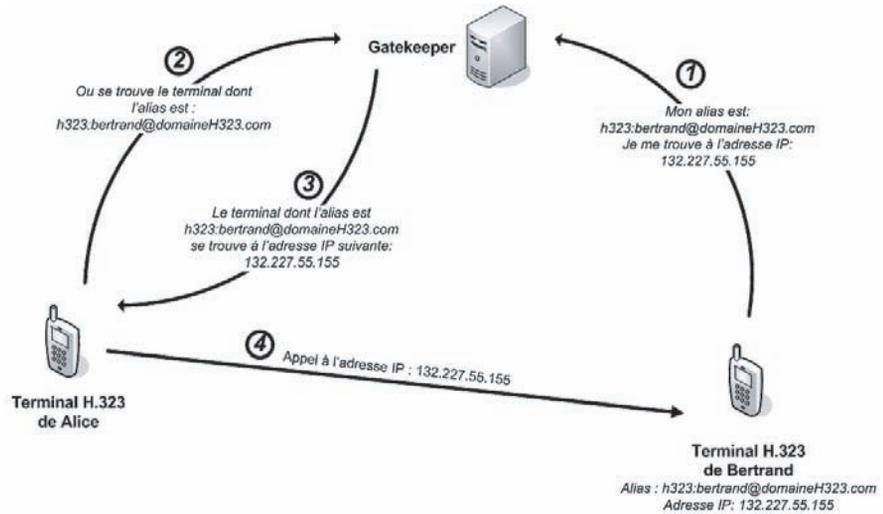
- une adresse de type e-mail, éventuellement préfixée de l'indication *h323*: spécifiant qu'il s'agit d'un alias H.323 ;
- une adresse de type numéro de téléphone (recommandation E.164 de l'UIT-T) ;
- une chaîne de caractères Unicode quelconque ;
- une adresse de type URL ;
- une adresse IP, éventuellement suffixée du numéro de port à utiliser.

Les adresses suivantes sont donc des alias H.323 valides : *albert@domaineH323.com*, *albert323*, *132.227.55.155:1720*, *0323323323*, etc.

La translation d'un alias vers une adresse IP est illustrée à la figure 3.6. Lors de sa connexion au réseau IP, Bertrand indique au gatekeeper sa localisation dans le réseau (étape 1), comme tout utilisateur qui se connecte. Le gatekeeper a sauvegardé cette association de l'alias avec l'adresse IP correspondante dans sa base de données. Lorsqu'Alice souhaite joindre Bertrand, elle ignore sa localisation mais dispose de son alias. En sollicitant le gatekeeper (étape 2), Alice peut donc déterminer la localisation de Bertrand (étape 3) puis initier un appel vers ce dernier (étape 4). Nous verrons plus loin à quelles requêtes et réponses correspond chacune de ces étapes.

Figure 3.6

Traduction d'adresses  
par le gatekeeper



### Autres fonctionnalités du gatekeeper

Initialement chargé d'assurer seulement cette traduction d'adresses, le gatekeeper est progressivement devenu un équipement de point de contrôle dans lequel se concentre l'ensemble des fonctionnalités complémentaires du réseau.

Parmi elles, les fonctionnalités suivantes sont spécifiées dans la norme comme indispensables et implémentées systématiquement dans tous les gatekeepers :

- Contrôle d'admission. Si la bande passante ne permet pas d'établir un nouvel appel dans une zone H.323, la passerelle est habilitée à interdire de nouveaux appels et à établir une liste de priorités d'appels licites.
- AAA (Authentication, Authorization, Accounting), ou authentification, autorisation et comptabilisation. L'authentification permet de connaître l'identité de la personne connectée, tandis que l'autorisation indique quels sont les droits (et éventuellement les conditions) attribués à la personne qui s'est authentifiée.
- Gestion des flux. Le gatekeeper peut implémenter un gestionnaire de bande passante pour décider de l'allocation de bande affectée aux terminaux. Il est en outre possible de limiter le nombre d'intervenants dans une conférence et de rejeter certaines demandes de flux (par exemple en n'autorisant que la voix à un utilisateur qui réclame l'audio et la vidéo).

Optionnellement, il est possible d'implémenter au sein de ce serveur des fonctionnalités de gestion et de surveillance de la zone H.323 de façon à assurer divers services, notamment les suivants :

- gestion des coûts, par le biais d'un système de facturation s'interfaçant avec les appels et calculant la durée de chaque communication ;
- mise en place d'annuaire, particulièrement utile en entreprise, mais tout aussi disponible dans un cadre plus vaste ;

- gestion des appels avec différents services proposés, notamment le transfert d'appel, la mise en attente, la restriction d'appels en provenance de terminaux spécifiques ou bien à des horaires déterminés, entre autres possibilités ;
- historique des appels effectués, sauvegardés dans des journaux et ultérieurement exploitables ;
- génération de statistiques d'utilisation.

Ces possibilités peuvent être étendues, et il ne s'agit ici que d'exemples courants.

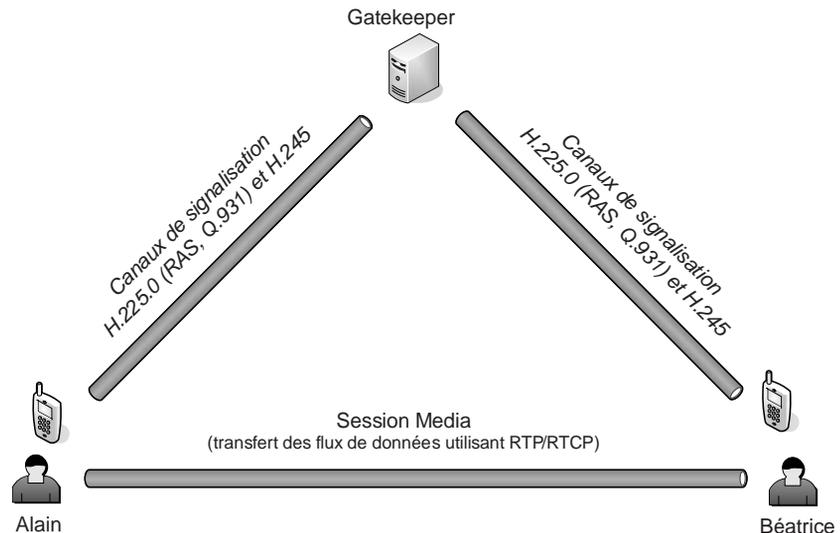
### Signalisation routée et directe

Pour mettre en relation deux utilisateurs, il est possible d'utiliser deux moyens différents pour faire transiter la signalisation dans le réseau :

- un mode indirect, ou routé, la signalisation entre les correspondants passant par le gatekeeper ;
- un mode direct, la signalisation entre les correspondants ne faisant intervenir que ces correspondants, sans entité intermédiaire.

Dans le mode indirect, toute la signalisation passe systématiquement par le gatekeeper. Ce dernier garde donc la supervision totale de la communication et peut intervenir ensuite lors des négociations entre les utilisateurs, en interdisant certains flux vidéo, par exemple, ou en sauvegardant les paramètres négociés lors de l'appel, ce qui peut être utilisé à des fins de facturation notamment.

**Figure 3.7**  
*Signalisation  
en mode indirect*

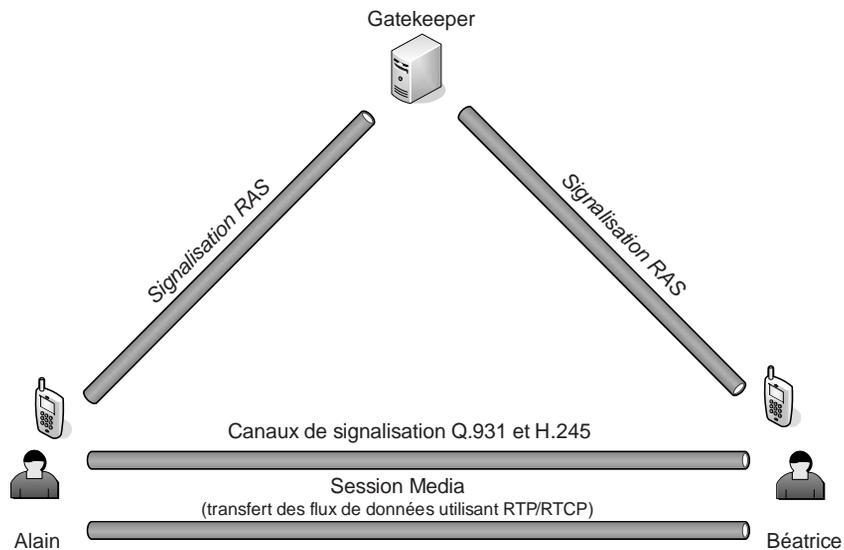


Comme l'illustre la figure 3.7, l'inconvénient immédiat de cette méthode est que le gatekeeper, déjà fortement sollicité dans une zone H.323, l'est davantage encore puisqu'il fait transiter l'ensemble des messages de signalisation.

Seule la partie signalisation de l'appel est concernée par cette redirection vers le gatekeeper, la transmission des flux multimédias eux-mêmes ne faisant pas intervenir le gatekeeper mais seulement les utilisateurs finals.

Dans le mode direct, les interlocuteurs s'échangent la signalisation entre eux, comme l'illustre la figure 3.8. Le gatekeeper joue cependant toujours son rôle, et les intervenants l'utilisent pour effectuer préalablement à l'appel la traduction d'adresse permettant de localiser le terminal appelé puis pour s'y authentifier et être soumis au contrôle d'admission ainsi qu'à toutes les fonctionnalités dont dispose le gatekeeper. Ce n'est qu'ensuite que les informations de signalisation sont envoyées uniquement entre les correspondants, exactement comme pour un appel dans un système H.323 ne faisant pas intervenir de gatekeeper.

**Figure 3.8**  
*Signalisation en mode direct*



### **La passerelle, pour joindre les réseaux ne fonctionnant pas en mode paquet**

Le protocole H.323 s'appuie nécessairement sur un réseau à commutation de paquets. Le réseau téléphonique classique, dit RTC (réseau téléphonique commuté), repose quant à lui sur une technologie à commutation de circuits, non compatible avec la commutation de paquets. Il est donc important que le protocole H.323 fournisse les moyens de communiquer avec les utilisateurs RTC.

À l'origine même de H.323, la recommandation H.320 visait à spécifier les caractéristiques des systèmes et équipements de vidéoconférence dans un réseau RNIS. Cette recommandation a ensuite été déclinée pour les autres types de réseaux (série de recommandations H.32x), donnant naissance à H.323. Il est donc naturel que la jonction entre

ces réseaux soit prévue. Cette fonctionnalité est fournie par un équipement dédié, la passerelle.

La passerelle assure l'interconnexion d'un réseau à commutation de paquets (typiquement les réseaux IP) avec les réseaux qui ne sont pas à commutation de paquets, incluant notamment les réseaux RTC, RNIS et ATM, afin de permettre à des utilisateurs du réseau IP de joindre des utilisateurs d'un autre type de réseau.

La figure 3.9 illustre le rôle d'une passerelle pour joindre un réseau tel que RTC.

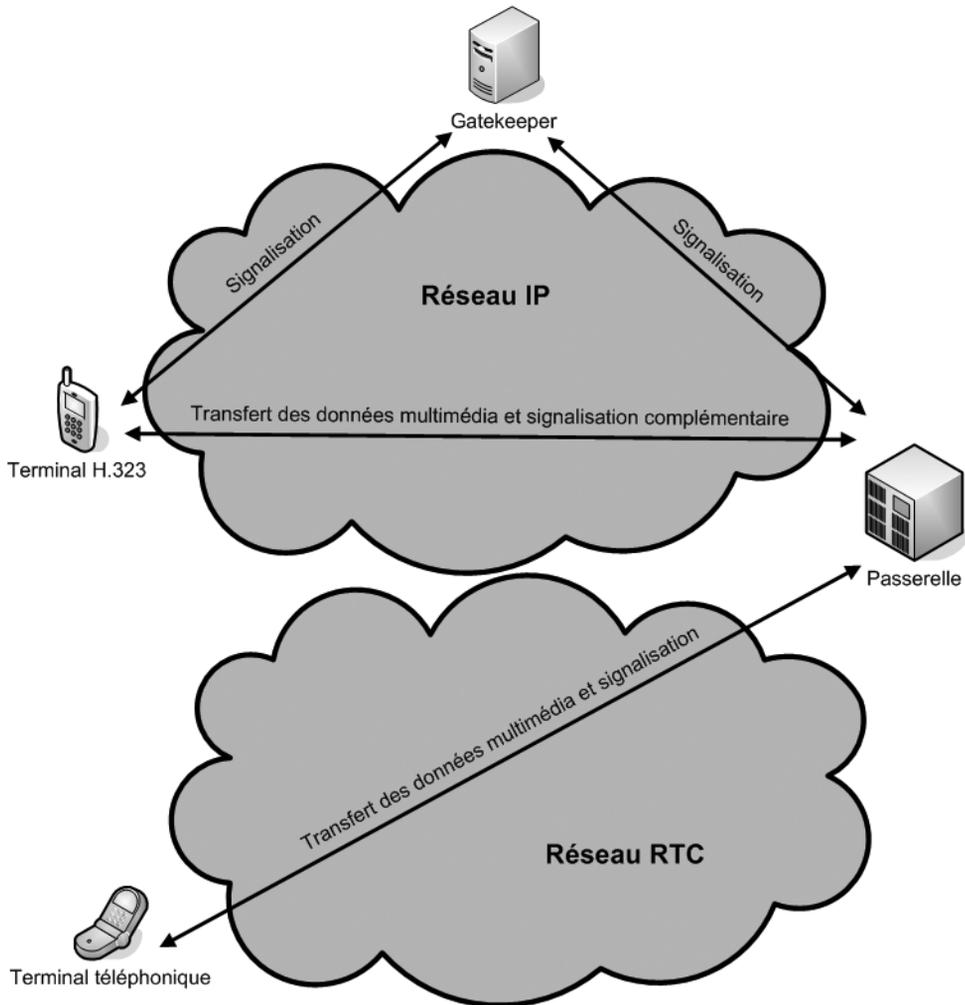


Figure 3.9

*Utilisation d'une passerelle pour joindre un réseau non IP*

Le concept de passerelle est très large et ne concerne pas seulement le protocole H.323. Une passerelle est en effet requise chaque fois que l'on souhaite joindre un réseau non-IP. Elle est indispensable, par exemple, pour joindre un téléphone conventionnel à partir d'un ordinateur relié à Internet. C'est pourquoi les fournisseurs d'accès ont systématiquement recours à des passerelles, qui ne sont pas forcément corrélées avec le protocole H.323, pour offrir leur service de téléphonie à leurs abonnés.

Les passerelles sont des équipements optionnels, dont la présence impose celle d'un gatekeeper, puisque la localisation de l'appelé ne peut être fournie que par ce dernier. Les utilisateurs d'un réseau non-IP ne disposant pas d'adresse IP, si un utilisateur du réseau IP dispose d'un numéro de téléphone pour joindre un contact, il doit demander au gatekeeper qui régit la zone, à quelle passerelle il doit s'adresser pour trouver son interlocuteur.

Comme illustré à la figure 3.10, les normes suivantes permettent de spécifier l'interconnexion entre les réseaux :

- H.320 pour les réseaux RNIS ;
- H.310 et H.321 pour les réseaux ATM ;
- H.322 pour les réseaux locaux offrant une qualité de service ;
- H.324 pour les réseaux à commutation de circuits (RTC).

Sans être exhaustive, cette liste est significative de la grande flexibilité offerte en termes d'adaptation et d'interfaçage aux réseaux ne fonctionnant pas en mode paquet, puisqu'on y retrouve les réseaux les plus répandus.

La complexité des passerelles a incité les équipementiers à implémenter des modifications propriétaires aux procédures décrites dans les normes protocolaires afin de simplifier leur utilisation. Dans ces cas, le respect des recommandations n'étant pas rigoureux, l'interopérabilité entre constructeurs n'est pas nécessairement assurée.

La passerelle est tenue d'assurer les deux fonctionnalités suivantes :

- Correspondance de signalisation. La signalisation utilisée dans un réseau IP étant différente de celle utilisée dans un réseau, par exemple, RTC, la passerelle se charge de convertir un signal de contrôle respectant la norme H.323 des réseaux IP en un signal de contrôle équivalent respectant la norme d'un réseau RTC. Pour joindre le réseau RTC, la passerelle adapte donc les messages H.323 en messages SS7.
- Adaptation des supports de communication. Cela permet d'assurer la cohésion entre les médias en garantissant notamment les opérations de multiplexage des données, de correspondance des débits et de transcoding audio. Ainsi, l'interlocuteur dans le réseau IP peut utiliser un certain codec, tandis que son correspondant en utilise un autre. Tous les flux transitent par la passerelle, qui se charge d'effectuer la correspondance de ces codecs et de transmettre à chaque intervenant le type de flux qu'il sait interpréter.

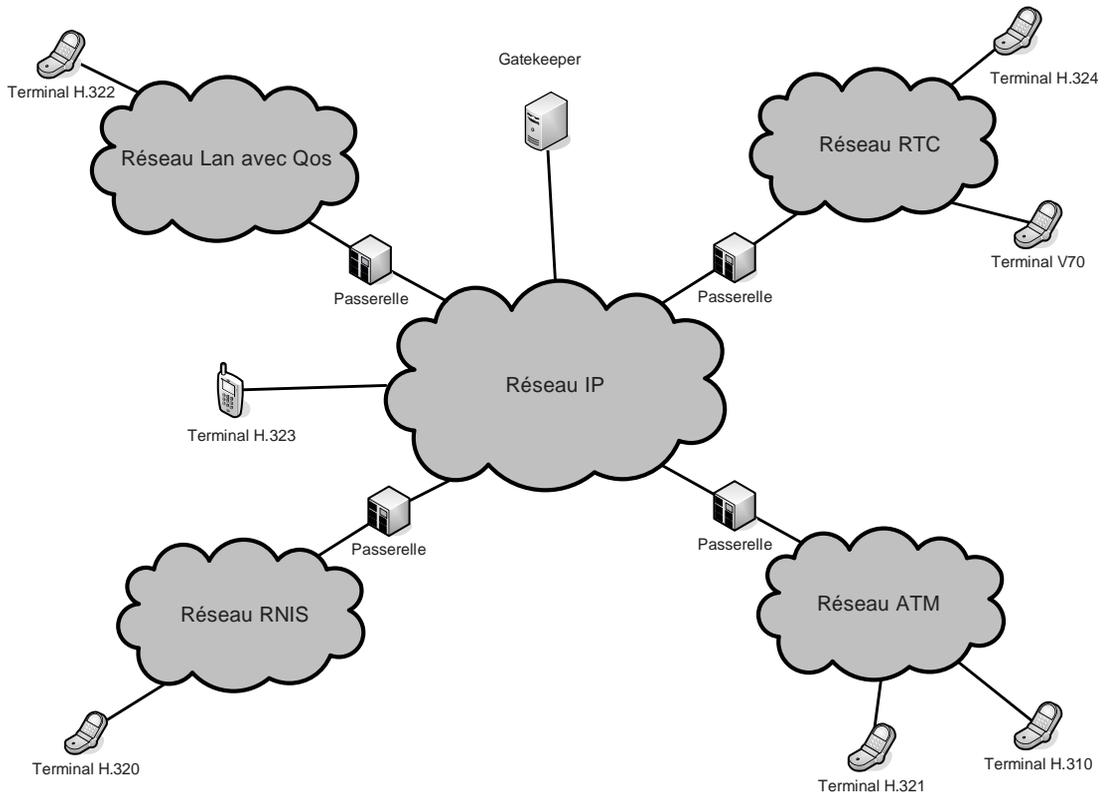


Figure 3.10

*Normes d'interconnexion de réseaux par le biais d'une passerelle*

### Architecture de la passerelle

Les fonctionnalités des passerelles en ont fait des équipements lourds, complexes, très sollicités et coûteux. C'est la raison pour laquelle la version 4 de la recommandation H.323 a épuré leur rôle.

Consciente de la charge importante assignée à la passerelle, l'UIT a travaillé de concert avec l'IETF afin de simplifier ses fonctionnalités en la décomposant en deux sous-entités compatibles avec le modèle de la recommandation H.248 : une entité de traitement, appelée MG (Media Gateway), ou passerelle multimédia, et une entité de contrôle, appelée MGC (Media Gateway Controller), ou contrôleur de passerelle multimédia.

Dans cette nouvelle architecture, l'ancienne passerelle devient une Media Gateway, qui a pour unique fonction d'effectuer le transcodage audio entre les différents réseaux. Toutes les MG sont elles-mêmes reliées et contrôlées par le MGC, lequel agit comme un contrôleur unique, centralisant l'ensemble des signalisations de contrôle entre les différents réseaux. Si les MG sont les entités qui appliquent les traitements, le MGC est l'entité de

gestion qui donne les ordres aux MG. Nous reviendrons sur ce modèle au chapitre 5, dédié au protocole MGCP.

Retenons pour l'heure que la passerelle se décompose en deux sous-parties, la passerelle multimédia et le contrôleur. Ces deux fonctionnalités logiques distinctes peuvent être regroupées au sein d'une même machine, notamment s'il n'y a qu'une seule passerelle.

### La MCU et les conférences

La MCU (Multipoint Control Unit) est utilisée pour mettre en place des conférences multimédias entre plusieurs utilisateurs, au moins deux. Comme l'illustre la figure 3.11, tous les utilisateurs désireux de participer à une conférence doivent se connecter à la MCU afin d'y définir et de négocier les paramètres de communication à utiliser.

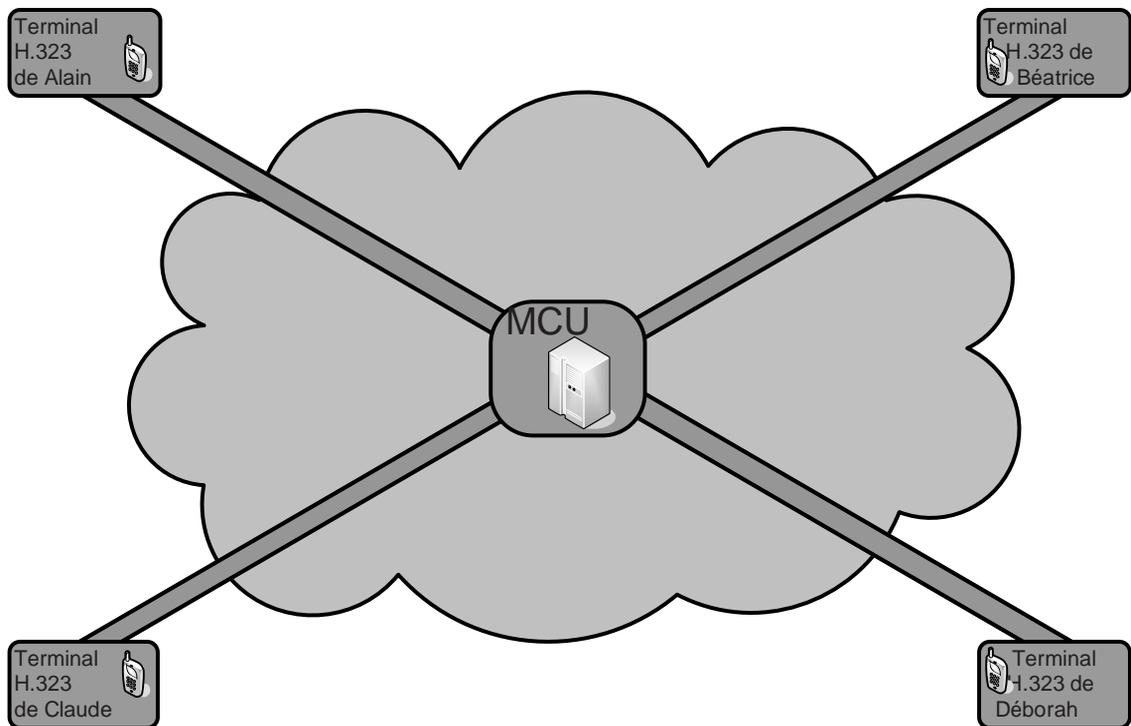


Figure 3.11

*Rôle de la MCU lors d'une conférence*

Contrairement aux autres types d'équipement, la MCU ne s'intéresse pas uniquement à la signalisation, mais aussi au transfert des données multimédias. Nous allons décrire les entités constituantes d'une MCU, avant d'examiner comment s'effectue une conférence multimédia.

## Le contrôleur et l'exécutant

La MCU désigne un équipement composé de deux entités, qui jouent un rôle complémentaire, le MC (Multipoint Controller) et le MP (Multipoint Processor).

### MC (Multipoint Controller)

Le Multipoint Controller est chargé de la négociation des paramètres. En préalable à la conférence, tous les utilisateurs négocient l'ensemble des paramètres de communication qu'ils désirent utiliser, selon les capacités de leur terminal et leur souhait. Ils conviennent notamment du mode d'adressage (unicast ou multicast), du type de flux souhaité (audio, vidéo ou les deux), du codec à utiliser et de la bande passante nécessaire.

C'est le MC qui centralise la demande de chacun des intervenants et leur fournit en réponse les paramètres effectifs à utiliser, en essayant d'optimiser les sollicitations faites par l'ensemble des terminaux.

Le MC n'intervient que pour les signaux de contrôle, à l'exclusion donc des données multimédias proprement dites, auxquelles il ne s'intéresse pas. Contrairement au MP, le MC n'est pas un intermédiaire de la communication. Il est toujours soit émetteur soit récepteur des messages et ne fait pas transiter les messages qu'il reçoit d'un poste vers un autre.

### MP (Multipoint Processor)

Le Multipoint Processor est un centre de traitement des flux multimédias.

Dans une conférence, chaque utilisateur peut disposer de paramètres spécifiques. L'un peut réclamer un codec audio de très bonne qualité, un autre un codec de moins bonne qualité, un troisième ajouter la vidéo en plus de l'audio. Pour satisfaire ces demandes, tous les utilisateurs se connectent auprès de l'entité MP, laquelle leur délivre à chacun, dans la limite de ses possibilités, les flux qu'ils sollicitent.

Concrètement, chaque intervenant adresse au MP l'ensemble de ses flux. Lorsqu'il les reçoit, le MP assure le multiplexage des paquets, ainsi que la synchronisation des données (régulation des diffusions et synchronisation des flux de voix avec les flux de vidéo) et l'adaptation des formats de codage et des débits. Les flux résultants sont envoyés aux destinataires.

Le MP permet de centraliser et d'adapter les échanges, mais ce n'est pas un composant obligatoire pour les conférences. De plus, comme il est fortement sollicité par les intervenants, il est possible d'associer à un même MC plusieurs MP.

Comme la MCU définit une fonctionnalité logique, n'importe quelle autre entité peut jouer son rôle, y compris les terminaux et le gatekeeper. Mais il est généralement préférable d'avoir un seul serveur dédié, disposant de la puissance de traitement suffisante pour tenir des charges importantes sans dégrader les performances des communications.

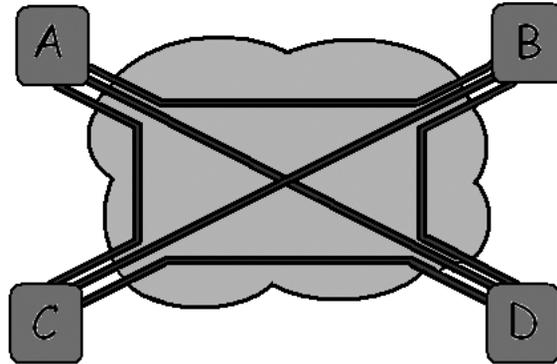
### Les trois modes de diffusion possibles

Pour offrir un service de conférence dans un réseau téléphonique, il est nécessaire de disposer de ressources réseau très importantes. En effet, chaque participant doit individuellement être mis en relation avec chacun des autres intervenants de la conférence, avec autant de liaisons à son extrémité qu'il y a d'intervenants.

La figure 3.12 illustre l'architecture d'une telle conférence avec quatre participants.

**Figure 3.12**

*Architecture d'une conférence à quatre participants en diffusion point-à-point*



Tous les participants de la communication étant mis en relation en point-à-point, cela réclame, pour chaque terminal, autant de liaisons qu'il y a d'intervenants. De plus, comme rien n'est centralisé, il est nécessaire de mettre en place des canaux dédiés.

Dans le monde IP, il est possible de réduire considérablement l'utilisation des ressources par trois méthodes : la centralisation des flux, la diffusion multicast et une combinaison de ces deux techniques.

#### Mode centralisé (unicast)

Une première méthode consiste à centraliser les flux vers un serveur unique. Ainsi, chaque utilisateur ne communique en unicast qu'avec ce serveur, lequel est en charge de la diffusion des flux à tous les abonnés inscrits à la conférence.

Dans ce cadre, la MCU fonctionne comme indiqué précédemment : les capacités MC et MP sont parfaitement adaptées pour respectivement recueillir les demandes de chacun des participants et leur fournir les flux adéquats.

L'avantage de cette méthode est que la MCU peut effectuer des traitements sur tous les flux qui transitent par elle. En outre, chaque intervenant ne maintient qu'un lien unique vers elle. Son inconvénient est la forte sensibilité aux pannes de l'architecture puisque la MCU est au centre des communications. En outre, l'utilisation de la bande passante reste importante, les paquets étant systématiquement envoyés vers la MCU avant d'aller vers les destinataires. Cela équivaut à deux routages de flux, alors qu'un cheminement de

l'émetteur vers les récepteurs n'impliquerait qu'un seul routage. C'est l'idée du mode décentralisé.

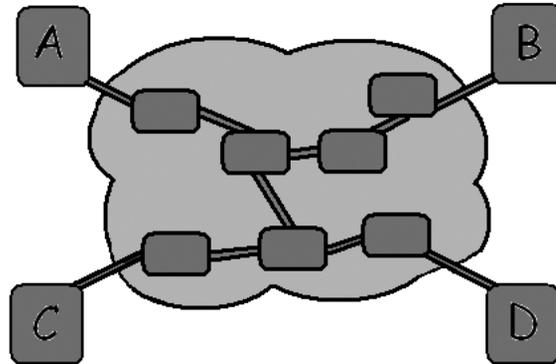
#### Mode décentralisé (multicast)

Une autre possibilité pour mettre en œuvre des conférences consiste à décentraliser les communications sans faire intervenir d'intermédiaire de transit. Les terminaux communiquent en ce cas directement entre eux.

Pour réduire l'utilisation de la bande passante, ils utilisent un adressage multicast, qui offre une diffusion unique des messages à l'ensemble du réseau. Ce scénario est illustré à la figure 3.13, dans laquelle quatre participants communiquent entre eux. Leurs messages transitent par des routeurs, qui sont capables de diffuser les flux simultanément sur plusieurs liens en même temps.

Figure 3.13

*MCU en mode décentralisé  
(multicast)*



La MCU n'est pas directement visible dans ce cadre, mais elle reste nécessaire pour que tous les intervenants conviennent des paramètres à mettre en place dans la communication. La MP s'avère totalement inutile ici, et la MCU se trouve réduite à sa seule fonctionnalité de MC, c'est-à-dire à sa partie contrôle.

Globalement, l'adressage multicast réduit considérablement les diffusions, puisque chaque message n'est diffusé qu'une fois dans le réseau. L'utilisation de la bande passante s'en trouve donc sensiblement diminuée. En outre, et contrairement à la solution précédente, les terminaux doivent effectuer eux-mêmes les traitements sur les flux, ce qui présente l'avantage de distribuer la charge du MP entre les terminaux. L'inconvénient majeur de cette méthode est que les composants du réseau doivent supporter le multicast. Cela vaut à la fois pour les routeurs, qui entrent dans le processus de routage des flux, et pour les pare-feu, qui doivent être disposés à laisser passer les flux multicast. Ces contraintes se révèlent particulièrement fortes dans la pratique.

### Mode hybride

Le mode de diffusion hybride utilise les deux méthodes précédentes de façon combinée. L'idée est de différencier les usages, soit par utilisateur, soit par type de flux, afin de fournir un service adapté. Dans le cadre d'une même conférence, le mode centralisé peut ainsi être dédié à l'audio et le mode décentralisé à la vidéo.

Le MP est de la sorte raisonnablement sollicité pour les besoins des flux audio uniquement et peut assurer les traitements pour les terminaux de faibles capacités. Quant aux autres terminaux, ils peuvent disposer de la vidéo en traitant ces types de flux. La distinction peut se faire suivant les contraintes de chaque terminal, en tenant compte notamment de ceux qui ne peuvent faire de multicast, du fait des équipements intermédiaires.

Ce mode permet au cas par cas de combiner les avantages et les inconvénients des modes précédents. Dans ce cadre, la MCU peut exploiter pleinement les capacités MC (pour tous les flux) et MP (pour les flux centralisés).

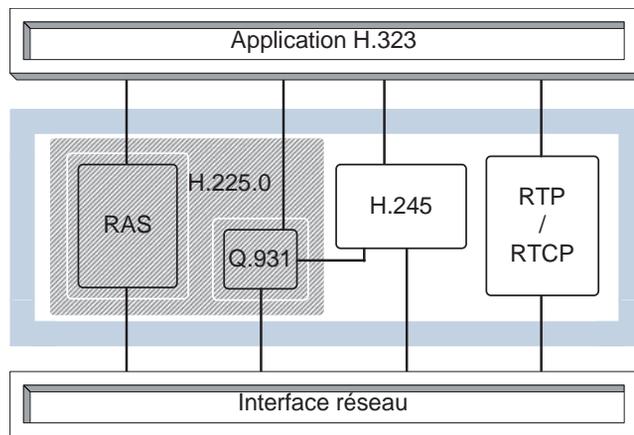
## Les messages H.323

Bien plus qu'un protocole, H.323 renvoie à une plate-forme complète décrivant comment des protocoles se combinent pour assurer la signalisation. Pour être fonctionnel, H.323 doit impérativement utiliser d'autres protocoles, qui forment son ossature. Les plus importants d'entre eux sont les standards fondamentaux H.225.0, qui exploite les protocoles RAS et Q.931, hérités du RNIS, et H.245.

Le protocole H.225.0 met en place un canal de signalisation d'appel et d'enregistrement afin d'assurer la mise en relation des interlocuteurs. Le protocole H.245 permet quant à lui de créer un canal de contrôle pour la négociation des paramètres de la communication (codeur utilisé, contrôle de flux, etc.).

Les couches protocolaires de ce modèle sont illustrées à la figure 3.14.

**Figure 3.14**  
*Couches protocolaires  
de H.323*



Initialement, les protocoles H.245 et Q.931 ne supportaient que le protocole de transport TCP, mais depuis la version 3 de H.323, ils supportent indifféremment TCP et UDP.

Les spécifications de ces protocoles étant particulièrement complexes, nous ne détaillons pas tous les mécanismes mis en œuvre lors d'une communication H.323 et nous nous contenterons d'observer les cas de figure les plus classiques des appels.

Dans les sections qui suivent, un endpoint est défini comme une entité source ou destinataire d'un message, qui peut correspondre aussi bien à un terminal qu'à une passerelle.

## ***Le protocole H.225.0, signalisation d'appel et d'enregistrement***

Le protocole H.225.0 est utilisé pour permettre la signalisation d'appel et la signalisation d'enregistrement (avec le contrôle d'admission). Ces deux types de signalisation sont assurés par les protocoles RAS (Registration Admission Status) et Q.931.

### **La signalisation d'appel avec Q.931**

La signalisation d'appel permet l'établissement d'un appel, la libération de la communication et la transmission des messages indiquant l'état d'un appel (occupation d'un poste, redirection, etc.). Elle regroupe les fonctionnalités de mise en paquet, de synchronisation, de multiplexage et de confidentialité.

C'est le protocole Q.931, également utilisé dans le cadre des réseaux RNIS, qui spécifie cette partie de la signalisation. Seul un sous-ensemble de ces messages Q.931 est applicable dans le protocole H.323.

Les cinq messages fondamentaux suivants doivent obligatoirement être supportés :

- SETUP : envoyé pour initier et établir une communication avec un terminal H.323.
- ALERTING : indique que le poste appelé est en train de sonner et que l'appelant se met en attente de sa réponse.
- CONNECT : indique que la communication peut débiter.
- RELEASE COMPLETE : envoyé pour initier la terminaison de l'appel.
- STATUS FACILITY : envoyé pour demander des services complémentaires.

Nous allons montrer comment s'effectuent l'ouverture et la fermeture d'un canal de signalisation d'appel, en restreignant le scénario aux seules étapes qui relèvent de la signalisation Q.931 (sans la localisation de l'appelé notamment).

#### **Exemple 1. Ouverture du canal de signalisation d'appel**

L'ouverture d'un canal de signalisation d'appel se fait généralement en trois étapes :

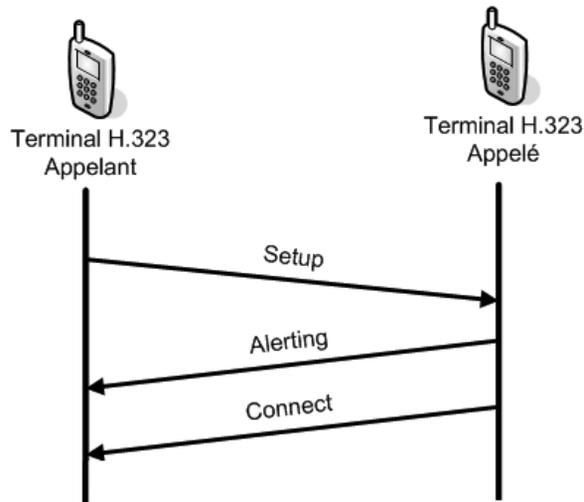
1. Message SETUP : l'appelant contacte son correspondant.

2. Message ALERTING : la sonnerie du terminal appelé retentit, et le terminal se met en attente de la réponse du correspondant.
3. Message CONNECT : dès que l'appelé a décroché, ce message prévient l'appelant de la disponibilité de son interlocuteur.

Ce scénario est illustré à la figure 3.15.

**Figure 3.15**

*Ouverture d'un canal de signalisation d'appel*



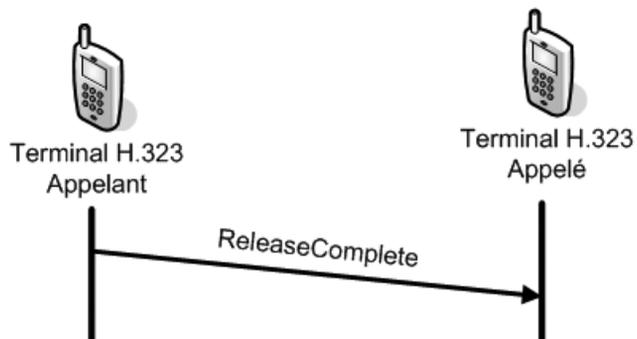
#### Exemple 2. Fermeture du canal de signalisation d'appel

La fermeture d'un canal de signalisation d'appel se fait à l'initiative de l'interlocuteur qui a raccroché son combiné, mettant fin à la conversation.

Un message RELEASE COMPLETE est envoyé pour fermer le canal de signalisation d'appel. Ce scénario est illustré à la figure 3.16.

**Figure 3.16**

*Fermeture d'un canal de signalisation d'appel*



## La signalisation d'enregistrement avec RAS

Le protocole RAS (Registration Admission Status) intervient pour les dialogues entre les terminaux et le gatekeeper, donc nécessairement dans une zone H.323.

Entre le terminal et le gatekeeper, on parle d'interface RAS pour indiquer que des messages RAS sont échangés entre ces deux entités. Le protocole RAS utilise UDP comme protocole de transport et les ports 1719 pour la diffusion d'un message à un seul destinataire (mode unicast) et 1718 pour les diffusions multiples (mode multicast). Généralement, tous les messages sont envoyés en unicast.

### Messages RAS génériques

Les messages RAS sont relativement simples et ressemblants. Chaque action possède généralement les trois déclinaisons de messages suivantes :

- xRQ : indique un message RAS de requête (REQUEST).
- xRJ : indique un message RAS de rejet de la requête (REJECT).
- xCF : indique que la requête a été correctement traitée (CONFIRM).

Le caractère x est ici générique et concerne n'importe quel message. Le tableau 3.1 récapitule les principales requêtes du protocole RAS. Chacune d'elles dispose des déclinaisons de réponse xRJ et xCF.

**Tableau 3.1 – Principales requêtes RAS**

Message RAS	Nom de référence	Description
GRQ	GATEKEEPER REQUEST	En envoyant ce message, le endpoint recherche un gatekeeper susceptible de le prendre en charge. Plusieurs gatekeepers peuvent répondre. L'adresse du gatekeeper peut être renseignée en statique sur le endpoint ou être déterminée par une résolution de type DNS (selon l'Annexe O).
RRQ	REGISTRATION REQUEST	Ce message permet au endpoint de s'enregistrer auprès du gatekeeper et de récupérer les services auxquels il a droit ( <i>voir section suivante</i> ). Dans le même temps, le gatekeeper affecte au endpoint un identifiant servant à ce dernier de référence pour tous les messages échangés avec le gatekeeper. Un paramètre « time to live » envoyé par le endpoint dans la requête indique la durée maximale au bout de laquelle le gatekeeper peut supprimer l'entrée associée au endpoint s'il n'a pas détecté d'activité de ce dernier. À tout moment, le endpoint peut rafraîchir son entrée (et donc son paramètre « time to live ») soit par un message complet RRQ, soit par un message plus court appelé LWRRQ.
LWRRQ	LIGHTWEIGHT REGISTRATION REQUEST	Ce message rafraîchit l'enregistrement d'un endpoint. Il ne peut que faire suite à un message RRQ et ne remplace pas ce dernier pour un premier enregistrement de endpoint. Plus court qu'un message RRQ, il est préférable pour rafraîchir une entrée.

Tableau 3.1 – Principales requêtes RAS (suite)

Message RAS	Nom de référence	Description
ARQ	ADMISSION REQUEST	Pour initier ou recevoir un appel, un endpoint doit être soumis à un contrôle d'admission auprès du gatekeeper. Le endpoint spécifie dans sa requête s'il est l'initiateur ou le récepteur de l'appel, ainsi que de quelle quantité de bande passante il a besoin et la valeur des deux paramètres suivants : le Call-Id (CALL IDENTIFIER) référençant un appel de manière unique et le CID (CONFERENCE IDENTIFIER) référençant une conférence en particulier (partagé par tous les conférenciers et à la valeur 0 si cet identifiant n'est pas déterminé). En retour, le gatekeeper autorise ou interdit l'appel selon des autorisations, droits d'accès et contraintes de qualité de service du réseau à respecter. Notamment, il peut décider de réduire la bande passante demandée.
LRQ	LOCATION REQUEST	Ce message est envoyé à un gatekeeper soit par un endpoint, soit par un gatekeeper pour demander la localisation d'un utilisateur, autrement dit pour résoudre un alias H.323 (par exemple, un numéro de téléphone) en une adresse IP. Généralement, c'est le gatekeeper qui se charge de la localisation.
BRQ	BANDWIDTH REQUEST	Ce message est utilisé par un endpoint pour demander au gatekeeper plus ou moins de bande passante qu'initialement sollicitée.
DRQ	DISENGAGE REQUEST	Ce message peut être envoyé par un endpoint vers un gatekeeper pour indiquer que la communication a pris fin ou par le gatekeeper vers un endpoint pour forcer ce dernier à terminer un appel. En principe, un gatekeeper ne doit pas envoyer de message de rejet DRJ à cette requête puisqu'il s'agit plutôt d'une information que d'une requête (utile, par exemple, pour connaître la durée de l'appel à des fins de facturation).

### Messages RAS particuliers

On observe des exceptions dans la déclinaison de messages de types xRQ/xRJ/XCF.

Les messages suivants sont également des messages RAS :

- **REQUEST IN PROGRESS (RIP)**. Le message RIP est une réponse temporaire qui acquitte la réception de la requête en indiquant que son exécution n'a pu être réalisée dans un délai normal mais qu'elle continue d'être traitée.
- **INFORMATION REQUEST (IRQ)/INFORMATION RESPONSE (IRR)**. La requête IRQ est envoyée par le gatekeeper vers un endpoint pour avoir des informations sur l'état d'une entité, par exemple pour connaître la disponibilité d'un terminal ou l'état d'un appel. La réponse est faite par un message IRR, qui décrit les informations demandées. Si un gatekeeper reçoit un message IRR alors qu'il n'a pas envoyé de requête IRR, il répond généralement par un message RAS d'acquiescement.
- **RESSOURCE AVAILABLE INDICATE (RAI)/RESSOURCE AVAILABLE CONFIRM (RAC)**. La requête RAI indique qu'un terminal a atteint ou est sur le point d'atteindre les capacités maximales disponibles. La confirmation par le gatekeeper de la réception du message RAI se fait par le message RAC.

- SERVICE CONTROL INDICATION (SCI)/SERVICE CONTROL RESPONSE (SCR). La requête SCI sollicite un service spécifique (auprès d'un terminal ou d'un gatekeeper), dont la réponse est envoyée par le message SCR.
- NON-STANDARD MESSAGE. Permet d'échanger des messages personnalisés (et donc non standards) entre le gatekeeper et un terminal.
- UNKNOWN MESSAGE RESPONSE. Utilisé pour indiquer que le message de requête n'a pas été reconnu (la requête est soit incorrecte, soit non prise en charge).

L'ensemble de ces messages reste cependant assez exceptionnel.

#### Exemple 1. Enregistrement d'un terminal auprès d'un gatekeeper

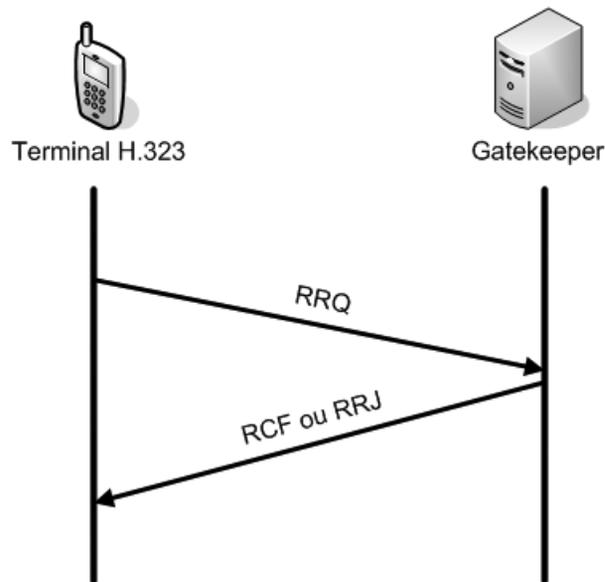
Lorsqu'un terminal se connecte dans une zone H.323, il doit s'enregistrer auprès du gatekeeper de la zone afin de lui indiquer sa présence dans le réseau, et donc sa disponibilité potentielle. Cela permet de recenser les terminaux pour ensuite fournir le service de localisation d'un utilisateur, soit à un terminal appelant, soit à un autre gatekeeper.

Le terminal fournit dans sa requête d'enregistrement son adresse IP associée à son identifiant H.323. De cette manière, tout utilisateur souhaitant joindre un terminal enregistré auprès du gatekeeper peut solliciter ce dernier en lui mentionnant l'identifiant H.323 du terminal à joindre. Dans le même temps, le terminal réclame au gatekeeper de disposer des services auxquels il a droit.

Ce scénario d'utilisation classique est illustré à la figure 3.17.

Figure 3.17

Requête d'enregistrement



Il se déroule de la façon suivante :

1. La requête RRQ (REGISTRATION REQUEST) est envoyée par le terminal au gatekeeper pour mentionner à ce dernier sa disponibilité dans le réseau. Les services sont demandés implicitement par cette requête. L'authentification peut être effectuée en même temps, de manière à proscrire l'usurpation d'identité entre utilisateurs.
2. En réponse, le gatekeeper retourne soit un message RCF (REGISTRATION CONFIRM) pour valider la demande d'enregistrement, soit un message RRJ (REGISTRATION REJECT) pour la refuser. Plusieurs raisons peuvent expliquer cet échec, notamment si la requête est incorrectement formulée ou si le gatekeeper a saturé sa base de données d'enregistrement ou encore si le gatekeeper est soumis à des restrictions sur les enregistrements et ne supporte qu'une liste d'abonnés fixes, par exemple.

### Exemple 2. Localisation d'un terminal

Le principe de la localisation d'un terminal a déjà été mentionné en présentation du rôle du gatekeeper. Nous mentionnons ici les messages relatifs à ce mécanisme.

La figure 3.18 illustre les différentes étapes de localisation d'un utilisateur. Nous supposons que les terminaux se sont préalablement enregistrés et associés chacun à un gatekeeper différent.

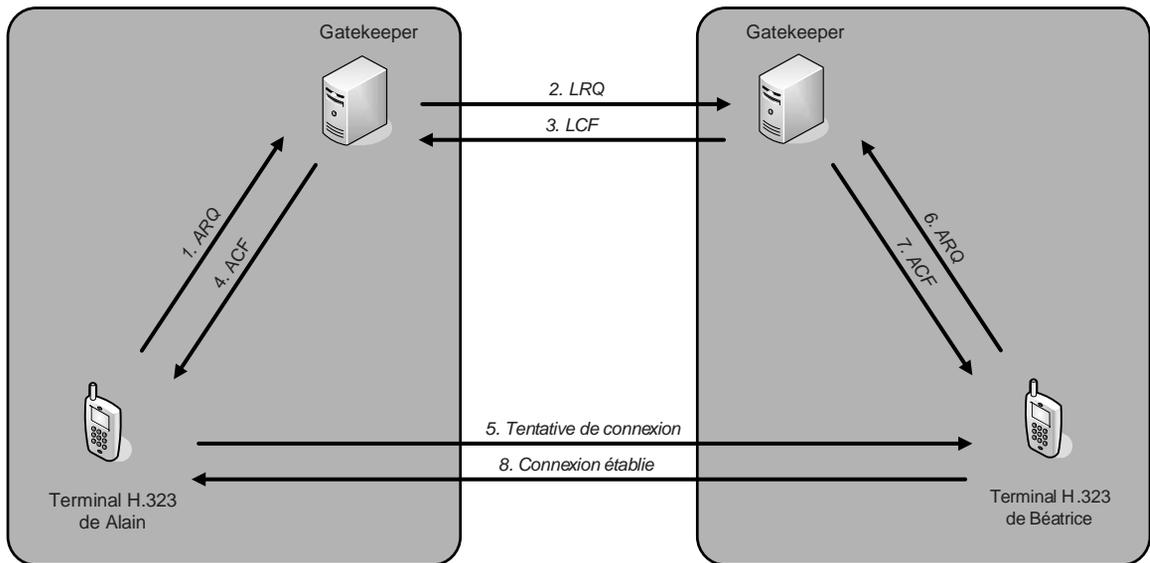


Figure 3.18

Étapes de localisation d'un utilisateur

On considère qu'Alain souhaite joindre Béatrice. Pour communiquer avec le terminal de Béatrice, il procède de la façon suivante :

1. Le terminal d'Alain envoie un message à son gatekeeper lui indiquant qu'il souhaite entrer en relation avec Béatrice.
2. Le gatekeeper d'Alain vérifie qu'il est autorisé à émettre cet appel. Si c'est le cas, il n'informe pas encore le terminal d'Alain de cette autorisation mais entreprend de localiser Béatrice. La réponse qui sera envoyée plus tard au terminal d'Alain contiendra à la fois l'autorisation d'appel et la localisation de l'appelé. Le gatekeeper d'Alain envoie un message de localisation LRQ vers le gatekeeper de Béatrice.
3. Le gatekeeper de Béatrice vérifie dans sa base de données qu'il a bien un enregistrement valable pour le terminal de Béatrice indiquant sa position courante (c'est-à-dire son adresse IP). Il retourne alors cette position au gatekeeper d'Alain par un message LCF.
4. Le gatekeeper d'Alain informe le terminal d'Alain de son autorisation à effectuer la communication sollicitée. Cela se fait par un message ACF incluant la position courante du terminal de Béatrice dans le réseau.
5. Le terminal d'Alain peut dès lors contacter le terminal de Béatrice (avec un message Q.931 SETUP que la figure ne mentionne pas afin de ne laisser que les messages RAS).
6. Le terminal de Béatrice envoie une demande ARQ à son gatekeeper pour demander l'autorisation de prendre cet appel, ainsi que l'allocation de bande passante nécessaire.
7. Le gatekeeper de Béatrice décide de valider la demande par un message ACF, éventuellement en réduisant la demande de bande passante si les contraintes réseau ou le profil de l'utilisateur l'exigent.
8. La communication est établie (en principe avec un message Q.931 ALERTING indiquant que le poste sonne puis un message CONNECT indiquant que le correspondant a répondu à l'appel).

### ***Le protocole H.245, la signalisation de contrôle de connexion***

Le protocole H.245 gère l'ouverture du canal de contrôle, l'établissement du canal de transmission, la négociation des paramètres (comme le codec utilisé) et le contrôle de flux ainsi que la fermeture du canal de contrôle. Comme pour le protocole Q.931, tous les messages H.245 ne sont pas exploitables dans le protocole H.323, qui n'en utilise qu'une faible proportion.

Initialement, les messages H.245 ne devaient être diffusés qu'après le message Q.931 SETUP. Pour optimiser les temps d'établissement d'une communication, les versions suivantes de H.323 ont fortement suggéré que les échanges H.245 s'établissent en parallèle ou même avant le message Q.931 SETUP.

### Le message TCS

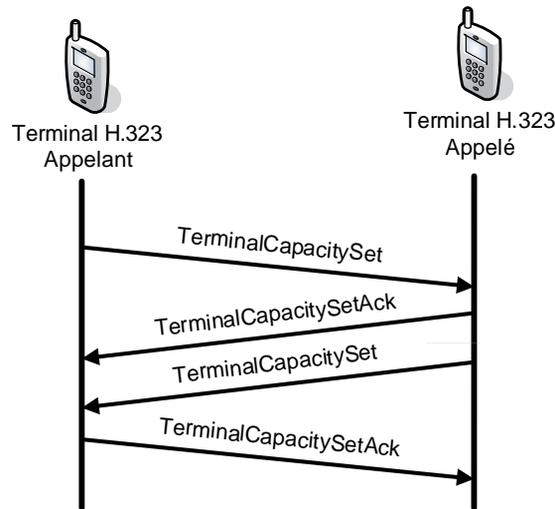
Le message TCS (TERMINALCAPABILITYSET) est obligatoirement le premier envoyé dans le canal H.245. Il indique les capacités du terminal qui émet ce message, notamment le type de média (audio, vidéo, données) et les codecs qu'il supporte.

Chaque terminal envoie la liste de ses capacités, généralement par ordre de préférence, dans un message TCS. À réception, un message d'acquiescement TERMINALCAPABILITYSETACK est retourné.

La figure 3.19 illustre les échanges de capacités entre deux terminaux.

**Figure 3.19**

*Échanges de capacités entre terminaux*



Le message TCS comporte une table, appelée CAPABILITY TABLE, mentionnant tous les paramètres supportés. Les combinaisons de paramètres possibles sont spécifiées dans un descripteur, appelé CAPABILITY DESCRIPTOR.

Par exemple, le terminal peut avoir les capacités soit pour la gestion d'une conversion audio avec un codec offrant une excellente qualité, soit pour disposer de l'audio et de la vidéo, le tout avec une qualité dégradée. Un seul CAPABILITY DESCRIPTOR est retenu par le correspondant pour communiquer. En confrontant les paramètres supportés par chacun des deux terminaux, des choix communs sont effectués.

### Le message MSD

Le protocole H.323 ne repose pas sur un modèle de type maître-esclave. Cependant, dans certaines situations conflictuelles entre deux entités équivalentes, il devient nécessaire de déterminer une entité qui impose ses choix à l'autre.

Par exemple, pour déterminer quelle MCU va être utilisée si les terminaux ont chacun un équipement différent pour jouer ce rôle, il est nécessaire d'arbitrer. Cela s'effectue au

moyen du message MSD (MASTERSLAVEDETERMINATION), qui doit être acquitté par le récepteur par un message MASTERSLAVEDETERMINATIONACK. En fait, la détermination du maître peut s'opérer préalablement, lors du message TCS, bien qu'il existe un message spécifique pour cela.

La détermination du maître et de l'esclave s'effectue au terme d'une négociation. Cela conforte le fait que les terminaux de l'architecture H.323 sont fondamentalement équivalents. Ils ne sont pas arbitrés par une entité supérieure, mais, dans certains cas, la désignation d'un terminal maître permet de trancher entre les directives à prendre et d'éviter ainsi des conflits entre terminaux.

### Le message OLC

Le message OLC (OPENLOGICALCHANNEL) permet d'ouvrir un canal de signalisation de contrôle (on dit aussi canal logique). Celui-ci indique le type de données multimédias transmis et les codecs utilisés.

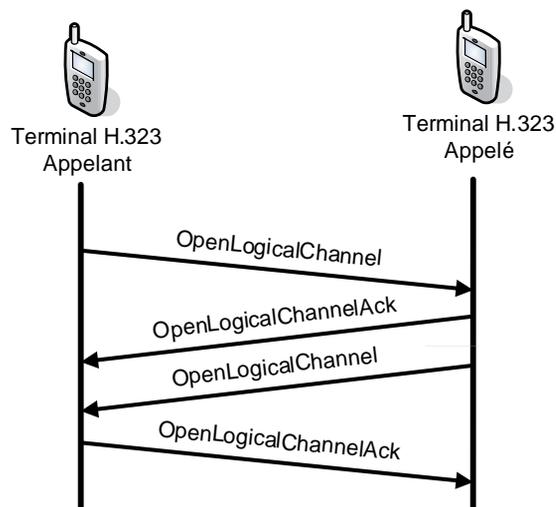
L'ouverture d'un canal logique se fait généralement sur un port TCP, mais peut tout aussi bien s'effectuer en UDP. Comme le canal n'est pas bidirectionnel, chaque terminal doit avoir un canal logique en utilisant le message OPENLOGICALCHANNEL. Ce message assigne un identifiant de session (session ID) à la communication. Par défaut, la session 1 est attribuée à une communication audio, la session 2 à une communication vidéo et la session 3 à une communication de données.

Le message doit en outre utiliser un codec sélectionné parmi ceux que le correspondant a préalablement mentionnés dans la requête TCS. Un message d'acquiescement OPENLOGICALCHANNELACK valide la requête.

La figure 3.20 illustre l'ouverture d'un canal de contrôle entre deux terminaux.

Figure 3.20

Ouverture d'un canal de contrôle



Cette étape étant optionnelle (voir la procédure de FASTCONNECT), il est possible de transporter les messages H.245 dans des messages H.225.0 (voir la procédure de H.245 tunneling).

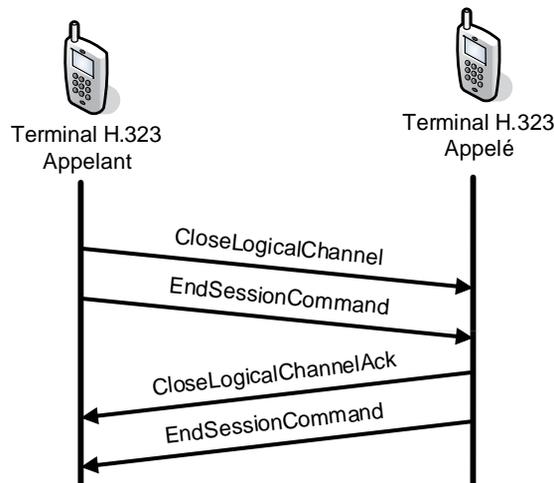
### Les messages CLC et ESC

Deux messages distincts sont nécessaires pour clôturer un canal de signalisation de contrôle : le message CLC (CLOSELOGICALCHANNEL), qui attend un acquittement CLOSELOGICALCHANNELACK, et le message ESC (ENDSESSIONCOMMAND), qui doit être émis par chaque intervenant.

La figure 3.21 illustre la fermeture d'un canal de contrôle entre deux terminaux.

**Figure 3.21**

*Fermeture d'un canal de contrôle*



Bien souvent, les constructeurs ne tiennent pas compte de la fermeture du canal de contrôle et se contentent d'un message de signalisation Q.931 pour indiquer la terminaison d'appel.

### Les autres protocoles

Bien d'autres protocoles sont utilisés dans la spécification H.323.

Le tableau 3.2 récapitule les principaux protocoles présents dans H.323.

**Tableau 3.2 – Principaux protocoles de H.323**

Protocole	Description
RTP (Real-time Transport Protocol)	Assure l'horodatage des paquets au niveau de l'émetteur pour permettre la synchronisation au niveau du récepteur.
RTCP (Real-time Transport Control Protocol)	Retourne des informations statistiques sur la qualité de la connexion du récepteur vers l'émetteur, afin que ce dernier puisse adapter ses envois en conséquence.

Tableau 3.2 – Principaux protocoles de H.323 (*suite*)

Protocole	Description
H.235.x	Les protocoles de sécurité à utiliser dans un système H.323 sont décrits dans les documents H.235 sous dix sections référencées de H.235.0 à H.235.9.
H.450.x	La série H.450.x définit un ensemble de protocoles pour la mise en œuvre de services supplémentaires. Alors que la spécification H.450.1 propose simplement un cadre générique, les suivantes spécifient la fourniture de services divers, comme le transfert d'appel (H.450.2), la mise en attente d'appel (H.450.4), l'indication d'un appel pendant un autre appel (H.450.6), la présentation de l'appelant (H.450.8), le renvoi d'appel (H.450.9), etc.
H.460.x	La série H.460.x définit un ensemble d'extensions qu'il est possible d'apporter au protocole de base. Par exemple, le document H.460.9 détaille comment un point de terminaison peut envoyer des informations de qualité de service pour permettre à ce dernier d'optimiser le routage des appels.
X.680	C'est le document de référence pour la syntaxe ASN.1 qui est utilisée dans le codage des données H.323.
X.691	Définit les règles d'encodage des paquets (Packet Encoding Rules) pour la transmission réseau.
T.120	Spécification pour l'échange de données lors des conférences, offrant la fiabilité des échanges et l'interopérabilité entre les constructeurs, tout en préservant une indépendance vis-à-vis du type de réseau utilisé.
T.38	Définit la manière de relayer les communications pour les fax.
V.150.1	Définit la manière de relayer les communications pour les modems.
H.26x	Ces documents détaillent les codecs normalisés pour les transmissions multimédias. Les deux plus utilisés sont H.261 pour le codage vidéo à débits multiples de 64 Kbit/s par seconde et H.263 pour le codage vidéo à faible débit.
H.510	Décrit un support pour la mobilité des utilisateurs en leur fournissant des services analogues quel que soit le terminal qu'ils utilisent.

### Exemple de scénario d'une communication complète

Une communication complète inclut l'ensemble des messages envoyés pour initier, établir et terminer une communication entre deux correspondants.

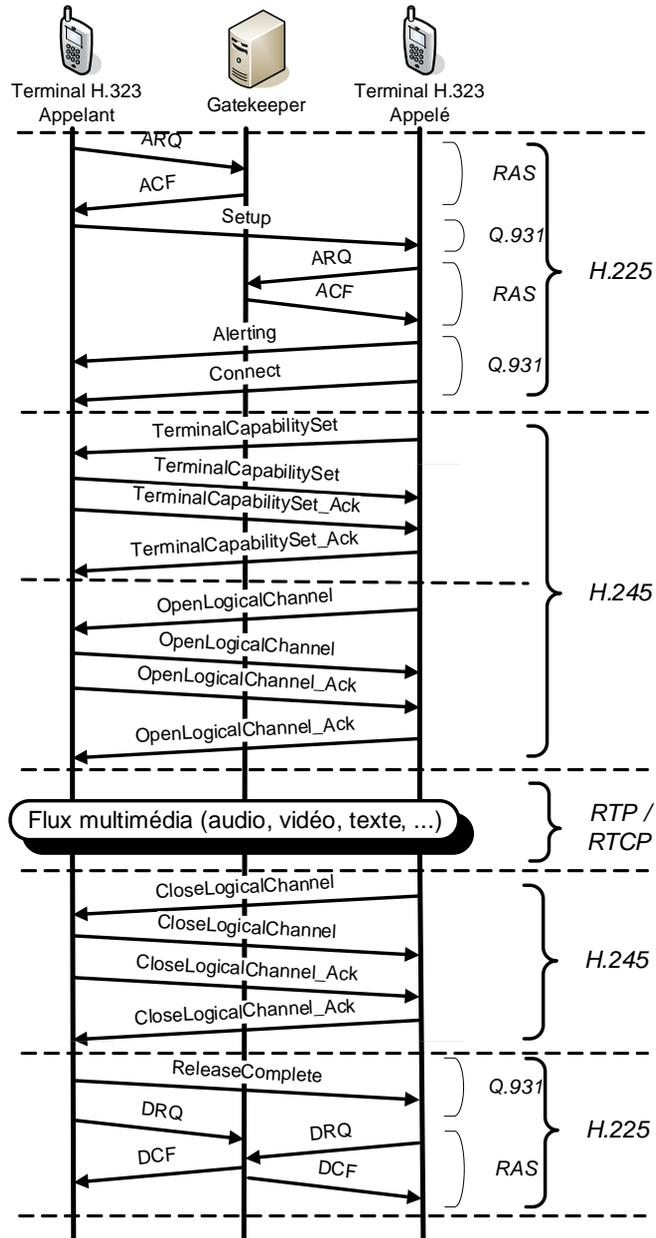
On considère une zone H.323 (il y a donc présence d'un gatekeeper pour le contrôle d'admission des terminaux). On suppose que la signalisation se fait en mode direct (seuls les messages de signalisation RAS sont routés vers le gatekeeper). On suppose également que ces terminaux se sont préalablement enregistrés auprès du gatekeeper et qu'ils dépendent tous deux d'un même gatekeeper (la localisation n'est donc pas à entreprendre).

Les optimisations que nous présentons plus loin dans ce chapitre ne sont pas prises en compte afin de simplifier la compréhension des différentes étapes. Nous verrons toutefois que, dans la pratique, l'implémentation de ces améliorations demeure indispensable.

La figure 3.22 illustre un exemple de communication complète entre deux terminaux H.323. Les étapes successives qui caractérisent cet échange sont les suivantes :

**Figure 3.22**

*Scénario complet d'une communication H.323*



1. Connexion (protocole H.225.0). Avant de contacter son correspondant, l'appelant s'assure d'être autorisé à émettre cet appel auprès du gatekeeper (RAS), puis envoie sa requête au terminal appelé (Q.931). Celui-ci s'assure à son tour auprès du gatekeeper d'avoir le droit d'effectuer cette communication (RAS). Pour les deux correspondants, on considère le cas où l'appel est autorisé pour pouvoir poursuivre. Dès que le terminal appelé reçoit la permission, il envoie des messages alertant l'appelant de sa disponibilité (Q.931) et lui confirmant que la connexion peut débuter (Q.931).
2. Négociation des paramètres et ouverture du canal de contrôle (protocole H.245.0). Commence un échange réciproque de messages pour déterminer les paramètres à mettre en place lors de la communication (H.245). À ce stade, il serait possible de négocier, par exemple, le choix des flux désirés (voix, vidéo ou les deux), le choix des codecs ainsi que les débits souhaités. Le canal de signalisation de contrôle est ainsi ouvert.
3. Début de la communication (RTP/RTCP). La communication audio et vidéo peut commencer en laissant transiter les médias.
4. Fermeture des canaux de signalisation. Lorsqu'un correspondant met fin à l'appel (en raccrochant son combiné, par exemple), la communication est clôturée par un message de signalisation fermant le canal Q.931. Ensuite, chacun des intervenants indique au gatekeeper la terminaison de l'appel, ce qui permet à ce dernier de déterminer la durée de l'appel à des fins de statistiques, journalisation ou facturation ou d'allouer la bande passante libérée à d'autres appels.

## Fonctionnalités avancées de H.323

Cette section détaille quelques fonctionnalités qui ont été ajoutées à la première recommandation de H.323 et sont devenues *de facto* des compléments non obligatoires mais fortement recommandés.

### La procédure Early H.245

En principe, le tunnel H.245 permettant la négociation d'appel n'est établi qu'après le message H.225.0/Q.931 SETUP. Ce message contient les paramètres permettant la mise en place du canal H.245.

L'idée à la base de ce choix est qu'il ne sert à rien de négocier des paramètres de communication tant que la personne appelée n'a pas accepté l'appel. Les négociations effectuées s'avèreraient totalement inutiles si la communication ne devait pas être établie. Si l'appel aboutit effectivement, les négociations de paramètres qui interviennent entre les intervenants retardent sensiblement le temps d'établissement de la communication. Par conséquent, anticiper ces actions se relève pratique.

Avec la procédure Early H.245, il est possible d'avancer l'établissement du canal H.245. Pour cela, il faut inclure les paramètres de mise en place du canal H.245 dans un message

précédant la requête SETUP. Typiquement, on utilise la requête ALERTING pour donner ces informations.

Dès que cette dernière est reçue, l'établissement du canal H.245 peut commencer parallèlement à la signalisation H.225.0. Globalement, cela accélère sensiblement l'étape de négociation des paramètres effectuée avec le protocole H.245, et l'appel est plus rapidement établi.

Cette optimisation particulièrement utile est souvent disponible en complément de la procédure FastConnect.

### ***La procédure FastConnect***

En principe, le canal H.245 permettant la négociation des paramètres d'appel est nécessaire pour l'établissement d'une communication. On constate cependant que l'ouverture d'un tunnel de contrôle H.245 est relativement longue et pénalisante pour l'établissement d'une communication (de l'ordre de 15 à 30 secondes dans l'implémentation de la première version de H.323, ce qui est excessivement long).

Tous les paramètres négociés par ce canal ne sont pas indispensables. Partant de ce constat, à partir de la version 2 de la recommandation H.323, l'ouverture du canal H.245 a été simplifiée et rendue optionnelle, tandis que d'autres moyens ont été proposés pour échanger les paramètres négociés par ce tunnel.

Avec la procédure FastConnect, il est possible d'inclure dans l'invitation d'appel des informations de canal de contrôle. De cette manière, l'établissement d'un appel peut se faire en seulement deux messages.

L'inconvénient de cette méthode est qu'elle ne permet pas la transmission d'informations DTMF. Il s'agit néanmoins d'une optimisation indispensable à l'établissement d'appels dans des délais raisonnables.

### ***La procédure H.245 tunneling***

Pour chaque appel, il est en principe nécessaire d'avoir deux connexions séparées : l'une pour les messages de signalisation d'appel Q.931, l'autre pour la mise en place du canal de contrôle H.245.

Avec la procédure H.245 tunneling, il est possible de n'avoir qu'un seul canal. Pour cela, les messages de contrôle H.245 doivent être encapsulés dans des messages H.225.0.

### ***La sécurité***

Absents de la première version du protocole, les mécanismes de sécurité ont été ajoutés dès la seconde avec la recommandation H.235.

En juin 2006, à l'occasion de la sixième version du protocole, la recommandation H.235 a été totalement remaniée et restructurée autour d'une série de documents numérotés de

H.235.0 à H.235.9. Ces documents détaillent les mécanismes de sécurité ajoutés à la norme en les répartissant en plusieurs volets, notamment les suivants :

- **Authentification.** Ce mécanisme permet de s'assurer de l'identité des correspondants. Tout utilisateur doit s'authentifier auprès du gatekeeper avant d'être autorisé à émettre ou recevoir un appel. C'est également indispensable pour garantir la bonne gestion des facturations d'appels.
- **Contrôle d'intégrité.** Ce mécanisme permet de s'assurer que les données sont transmises sans avoir été altérées ni corrompues pendant leur transfert.
- **Confidentialité.** Cet aspect couvre les méthodes de cryptage de données empêchant les écoutes clandestines d'une communication.
- **Non-répudiation.** Ce mécanisme permet de s'assurer de la provenance d'un message (empêchant l'émetteur de nier ultérieurement son envoi).

Le protocole H.323 supporte aussi la sécurisation des échanges multimédias *via* le protocole SRTP (Secure RTP).

### ***Gatekeeper alternatif et gatekeeper affecté***

Dans l'architecture H.323, les gatekeepers jouent un rôle prépondérant et sont relativement souvent sollicités par les autres équipements, ce qui en fait des entités particulièrement sensibles.

Deux mécanismes, appelés gatekeeper alternatif et gatekeeper affecté, ont été prévus pour améliorer la stabilité et la robustesse du réseau, même en cas de panne d'un gatekeeper.

#### **Gatekeeper alternatif**

Si un gatekeeper tombe en panne, un autre, le gatekeeper alternatif, peut prendre dynamiquement le relais et traiter les appels en cours. Cette possibilité a été introduite dans la version 2 du protocole H.323. Les terminaux qui la supportent peuvent basculer vers un nouveau gatekeeper dès qu'ils détectent une panne de leur gatekeeper initial.

Les gatekeepers ainsi placés en redondance procurent une meilleure stabilité au réseau. En outre, il devient possible d'effectuer de la répartition de charge afin qu'un gatekeeper trop sollicité puisse se faire relayer par un autre.

Pour les opérateurs, cette fonctionnalité assure une meilleure disponibilité et une continuité de services, même en cas de défaillances matérielles ou logicielles.

#### **Gatekeeper affecté**

Avec ce mécanisme, chaque terminal dispose par défaut d'un gatekeeper de référence, auprès duquel il cherche systématiquement et préférentiellement à s'enregistrer. Ce gatekeeper est appelé gatekeeper affecté. Si, à la suite d'une panne, d'une saturation ou pour

toute autre raison, la connexion avec celui-ci s'avère impossible, le terminal doit basculer vers un gatekeeper alternatif pour continuer à communiquer normalement.

Tout en étant sous le contrôle d'un autre gatekeeper que celui affecté, le terminal continue à surveiller périodiquement son gatekeeper affecté afin de vérifier sa disponibilité. En cas d'activité détectée, le terminal se réoriente vers lui pour s'y enregistrer. Cela permet de répartir au mieux les utilisateurs entre les gatekeepers et par voie de conséquence de disposer d'une meilleure gestion des ressources disponibles.

Cette fonctionnalité a été introduite dans la version 6 du protocole H.323.

## Conclusion

Le protocole H.323 a constitué un tournant dans l'histoire de la téléphonie sur IP. Symbole de l'unification des fonctionnalités de signalisation pour la téléphonie dans un réseau IP, il a été le premier standard proposé et adopté massivement par les industriels. Il a ensuite conquis des marchés considérables qui ont rendu toute concurrence difficile à soutenir.

Mais s'il propose une réponse à la signalisation, le protocole souffre d'inconvénients contraignants pour supporter le passage à l'échelle au niveau mondial. Son exploitation dans le cadre du réseau Internet se heurte à la superposition d'une architecture centralisée dans un modèle totalement distribué.

En outre, le protocole demeure complexe et lourd à mettre en place. Pour combler ces faiblesses, les implémentations de H.323 par les industriels et les éditeurs de logiciel ont parfois dû s'autoriser quelques écarts par rapport à la définition stricte donnée dans les recommandations de l'UIT. Malheureusement, ces optimisations propriétaires diffèrent selon les équipements, si bien qu'au lieu de jouer un rôle fédérateur, comme le protocole, elles ont fait perdre l'interopérabilité entre les plates-formes de constructeurs distincts.

Enfin, la compatibilité ascendante exigée par H.323 oblige tous les constructeurs et fournisseurs de services à implémenter l'ensemble des mécanismes pour être compatibles entre eux, y compris ceux proposés dans les versions précédentes du protocole. Par exemple, pour être compatible avec la version 6 du protocole, un équipement doit nécessairement savoir communiquer avec un équipement exploitant une implémentation de l'une des versions 1 à 5. Autrement dit, le protocole est constamment enrichi sans jamais être véritablement épuré.

Aujourd'hui, H.323 tend à disparaître et à se marginaliser. Bien souvent, sa présence n'est justifiée que pour des raisons historiques. Le protocole qui devrait s'imposer comme son remplaçant, SIP, a pour sa part été entièrement conçu selon la philosophie du monde IP.

# 4

## Le protocole SIP

---

La téléphonie sur IP se situe à la jonction de deux mondes : celui des télécoms et celui d'Internet. Le premier a inventé le service, le second cherche à se l'approprier. Il était donc naturel que des intervenants de ces deux mondes soient à l'origine de la conception du protocole de signalisation qui en permet la gestion.

Cependant, au lieu de travailler en commun, chaque acteur de la normalisation a tenté de faire valoir sa vision de la téléphonie au sein de son propre protocole. Côté télécoms, le protocole H.323 de l'UIT propose une architecture centralisée qui rappelle les origines de la téléphonie traditionnelle. Côté Internet, le protocole SIP de l'IETF propose des mécanismes très proches de ceux des protocoles en vigueur sur Internet.

Le chapitre précédent a présenté H.323. Le présent chapitre détaille le protocole SIP.

### La standardisation SIP (Session Initiation Protocol)

L'IETF s'intéresse à la téléphonie sur IP et travaille à la mise au point d'un protocole chargé de la gestion de sa signalisation depuis 1995.

En 1997, la première version de ce protocole, nommé SIP, est dévoilée au public. Entre-temps, l'UIT lui avait volé la vedette avec H.323, sorti en 1996, qui avait bénéficié de la faveur des industriels et dont les implémentations logicielles, notamment NetMeeting de Microsoft, assuraient la célébrité.

Pendant plusieurs années, l'IETF n'a pas été un acteur visible dans le domaine de la ToIP. Plus le protocole tardait à voir le jour, plus le handicap par rapport à son concurrent H.323 s'amplifiait. Si le protocole H.323 possède aujourd'hui la maturité que lui confèrent son avance et ses nombreuses expérimentations, sa gestion demeure laborieuse et

reste peu adaptée au monde Internet. Or c'est à ce niveau qu'intervient SIP, dont la force principale vient de son extrême simplicité, même à grande échelle.

Le protocole SIP a été conçu pour s'adapter à Internet, en particulier pour que le réseau supporte des montées en charge du nombre d'utilisateurs. Pour cela, l'architecture SIP repose sur plusieurs serveurs distincts, qui distribuent la charge du réseau en communiquant entre eux, un peu à la manière des serveurs DNS sur Internet. Lorsque le nombre d'utilisateurs croît, il suffit d'ajouter des serveurs disposant de fonctions dédiées pour collaborer avec ceux déjà en place.

Cette approche se révèle hautement évolutive et flexible puisque de nouvelles fonctionnalités peuvent à tout moment être déployées, sans avoir à modifier les composants existants.

## Historique

SIP (Session Initiation Protocol) a été normalisé par le groupe de travail WG MMUSIC (Work Group Multiparty Multimedia Session Control) de l'IETF. La version 1 est sortie en 1997, et une seconde version majeure a été proposée en mars 1999 (RFC 2543). Cette dernière a elle-même été largement revue, complétée et corrigée en juin 2002 (RFC 3261). Des compléments au protocole ont été définis dans les RFC 3262 à 3265.

SIP est au sens propre un protocole de signalisation hors bande pour l'établissement, le maintien, la modification, la gestion et la fermeture de sessions interactives entre utilisateurs pour la téléphonie et la vidéoconférence, et plus généralement pour toutes les communications multimédias.

Le protocole n'assure pas le transport des données utiles, mais a pour fonction d'établir la liaison entre les interlocuteurs. Autrement dit, il ne véhicule pas la voix, ni la vidéo, mais assure simplement la signalisation. Il se situe au niveau de la couche applicative du modèle de référence OSI et fonctionne selon une architecture client-serveur, le client émettant des requêtes et le serveur exécutant en réponse les actions sollicitées par le client.

SIP fournit des fonctions annexes évoluées, comme la redirection d'appel, la modification des paramètres associés à la session en cours ou l'invocation de services. En fait, SIP ne fournit pas l'implémentation des services, mais propose des primitives génériques permettant de les utiliser. De cette manière, l'implémentation des services est laissée libre, et seul le moyen d'accéder aux services est fourni.

## Compatibilité

L'un des grands atouts de SIP est sa capacité à s'intégrer à d'autres protocoles standards du monde IP. En tant que standard ouvert, il offre un service modulaire, prévu pour fonctionner avec différentes applications, telles que la téléphonie, la messagerie instantanée, la vidéoconférence, la réalité virtuelle ou même le jeu vidéo.

En fait, plus qu'une simple compatibilité, c'est la possibilité de l'utiliser en conjonction avec d'autres protocoles qui caractérise SIP. Le protocole s'insère comme une partie d'un ensemble plus générique, intitulé Internet Multimedia Conferencing Suite. À l'image de H.323, SIP est peu à peu devenu un protocole dit parapluie, qui encadre et rassemble plusieurs autres protocoles.

SIP peut notamment se déployer ou s'intégrer aux protocoles suivants :

- RTP (Real-time Transport Protocol), RFC 3550, qui se charge du transport des flux temps réel.
- RTCP (Real-time Transport Control Protocol), RFC 3550, qui fournit des informations dynamiques sur l'état du réseau.
- RTSP (Real-Time Streaming Protocol), RFC 2326, pour contrôler la diffusion de flux multimédias en temps réel.
- SDP (Session Description Protocol), RFC 2327, qui fournit la description d'une session, c'est-à-dire les paramètres utilisés dans une communication SIP.
- SAP (Session Advertisement Protocol), RFC 2974, pour les communications multi-cast, qui permet d'ajouter les spécifications d'une nouvelle session.
- MIME (Multipurpose Internet Mail Extension), RFC 2045, standard pour les descriptions de contenus, utilisé sur Internet.
- RSVP (Resource reSerVation Protocol), RFC 2205, pour obtenir des garanties de qualité de service et effectuer des réservations de ressources.
- HTTP (HyperText Transfer Protocol), RFC 2616, pour le traitement des pages Web sur Internet (on peut inclure des adresses SIP directement dans des pages Web).
- MGCP (Media Gateway Control Protocol), RFC 3435, pour le contrôle des passerelles assurant la connectivité entre un réseau IP et un réseau téléphonique.

Tous ces protocoles sont d'une nature différente de celle de SIP, et ils n'interfèrent pas avec la signalisation. Leur utilisation conjointe est possible, voire recommandée pour certains d'entre eux.

Cela dit, aucun d'eux n'est indispensable au bon fonctionnement de SIP, qui reste totalement indépendant à leur égard et autorise *a priori* n'importe quel autre protocole. Dans la pratique, nous verrons cependant que SIP est classiquement utilisé avec les mêmes protocoles.

## **Modularité**

Comme expliqué précédemment, le protocole SIP se veut modulaire. Son objectif est de constituer une brique de base pouvant se combiner avec le maximum d'autres protocoles. C'est la raison pour laquelle il a été conçu d'une manière indépendante de la couche de transport.

Les protocoles TCP et UDP sont donc tous deux supportés pour l'envoi de messages SIP. UDP est généralement préférable pour laisser à l'application le contrôle des retransmissions de messages, et donc l'enchaînement des messages. Pour sa part, TCP est préférable pour la traversée de pare-feu, dans la mesure où les ports utilisés avec SIP sont dynamiques et où la notion d'état de connexion n'existe pas avec UDP.

Mais il ne s'agit là que de recommandations. Aucune règle n'est fixée, et même avec UDP, il existe des moyens de contourner le filtrage des pare-feu.

## Simplicité

SIP affiche une grande simplicité, comme l'atteste la taille de la spécification du protocole, qui ne dépasse pas 153 pages dans sa première version (RFC 2543) et 269 pages dans la seconde (RFC 3261), ce qui reste nettement inférieur aux 763 pages de la spécification H.323.

SIP utilise un langage textuel très proche des protocoles HTTP et SMTP, ce qui facilite son intégration à Internet. Par comparaison, le protocole H.323 utilise ASN.1, qui est un langage compilé.

Les avantages d'un langage textuel sont les suivants :

- Les traitements de commandes ne nécessitent pas de compilation et sont par conséquent plus rapidement interprétés.
- L'implémentation de nouveaux services ne nécessite pas de compilateur pour interpréter les commandes.
- Il est facile pour les programmeurs d'interpréter à la volée les actions en cours, puisqu'elles circulent en clair. La maintenance des services et l'implémentation de nouveaux services en sont d'autant facilitées.

La simplicité de SIP en fait un protocole facile à embarquer et un candidat de choix pour les composants légers, dotés de capacités réduites, comme les téléphones mobiles. Son implémentation est peu gourmande en ressources de traitement.

La simplicité de SIP ne l'empêche nullement d'être véritablement performant. Sa souplesse d'utilisation est ainsi l'une de ses caractéristiques principales. Il est, par exemple, possible de modifier une session en cours. Un nouveau participant peut se joindre à une conférence dynamiquement et facilement. Les modifications de paramètres sont prises en compte à la volée lors de la communication. De la même manière, il est possible d'ajouter de la vidéo à une communication audio ou encore de changer de codec à tout moment, ce qui simplifie la mise en œuvre du protocole dans une infrastructure quelconque.

Par ailleurs, l'encodage UTF-8 (Unicode Transformation Format), décrit dans la RFC 2879, utilisé pour les messages SIP permet d'utiliser un jeu universel de caractères Unicode (jeu de caractères ISO 10646) pour toutes les langues, et donc de recourir à d'autres caractères que l'alphabet occidental.

## Architecture de SIP

Contrairement à H.323, largement fondé sur une architecture physique, le protocole SIP s'appuie sur une architecture purement logicielle.

L'architecture de SIP s'articule principalement autour des cinq entités suivantes :

- terminal utilisateur ;
- serveur d'enregistrement ;
- serveur de localisation ;
- serveur de redirection ;
- serveur proxy.

La figure 4.1 illustre de façon générique les communications entre ces éléments. Un seul terminal étant présent sur cette figure, aucune communication n'est possible. Nous nous intéressons en fait ici aux seuls échanges entre le terminal et les services que ce dernier est susceptible d'utiliser lors de ses communications.

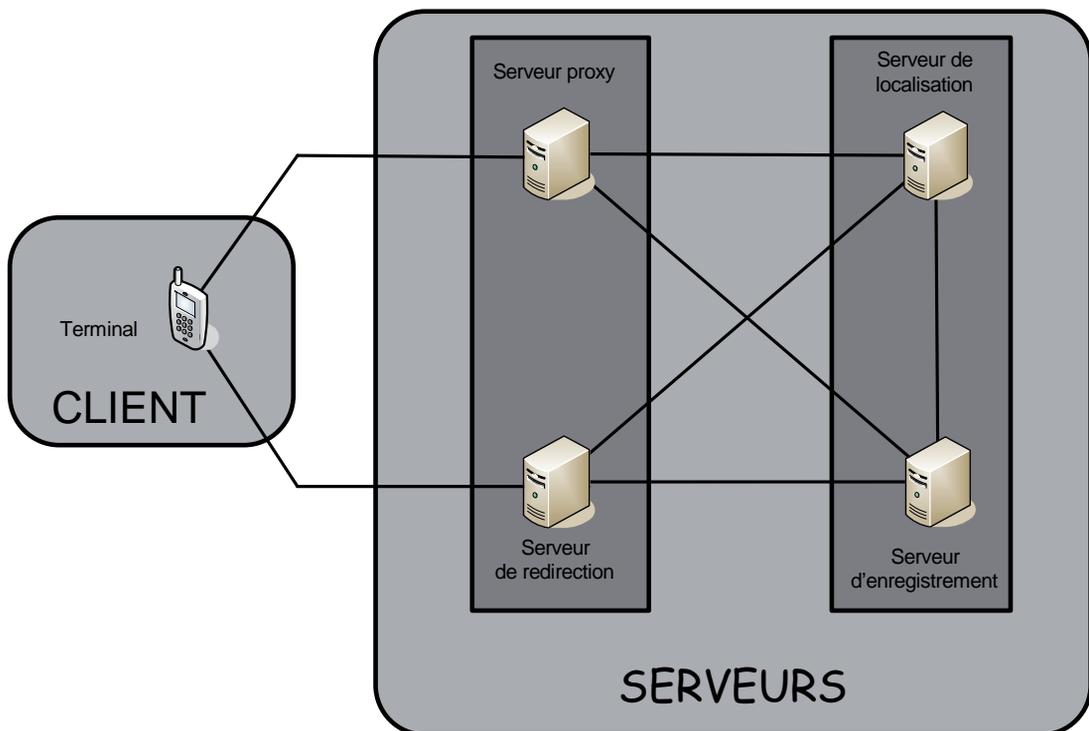


Figure 4.1

*Architecture de SIP*

On peut schématiquement observer qu'il existe deux catégories de services : l'un fourni au niveau de l'utilisateur (par le terminal), l'autre fourni au niveau des serveurs du réseau. Ces derniers sont répartis en deux classes : les serveurs de redirection et proxy, qui facilitent le routage des messages de signalisation et jouent le rôle d'intermédiaires, et les serveurs de localisation et d'enregistrement, qui ont pour fonction d'enregistrer ou de déterminer la localisation des abonnés du réseau.

### Terminal

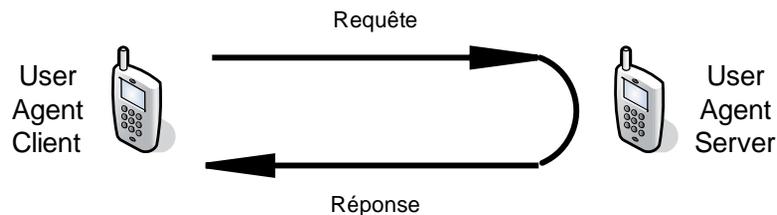
Le terminal est l'élément dont dispose l'utilisateur pour appeler et être appelé. Il doit donc permettre de composer des numéros de téléphone. Il peut se présenter sous la forme d'un composant matériel (un téléphone) ou d'un composant logiciel (un programme lancé à partir d'un ordinateur).

Le terminal est appelé UA (User Agent). Il est constitué de deux sous-entités, comme illustré à la figure 4.2 :

- Une partie cliente, appelée UAC (User Agent Client), chargée d'émettre les requêtes. C'est l'UAC qui initie un appel.
- Une partie serveur, appelée UAS (User Agent Server), qui est en écoute, reçoit et traite les requêtes. C'est l'UAS qui répond à un appel.

L'association des requêtes et des réponses entre deux entités de type UA constitue un dialogue.

Figure 4.2  
UAC et UAS



Par analogie, on peut remarquer que la même chose se produit avec le protocole HTTP dans une application Web : un utilisateur exploite son navigateur comme client pour envoyer des requêtes et contacter une machine serveur, laquelle répond aux requêtes du client. La différence essentielle par rapport aux applications standards utilisant HTTP est qu'en téléphonie un terminal doit être à la fois utilisé pour joindre un interlocuteur et pour appeler. Chaque terminal possède donc la double fonctionnalité de client et de serveur.

Lors de l'initialisation d'un appel, l'appelant exploite la fonctionnalité client de son terminal (UAC), tandis que celui qui reçoit la communication exploite sa fonctionnalité de serveur (UAS).

La communication peut être clôturée indifféremment par l'User Agent Client ou l'User Agent Server.

De nombreuses implémentations de clients SIP sont disponibles sur les plates-formes les plus courantes, Windows, Linux ou Mac. Elles sont le plus souvent gratuites, sous licence GPL.

Parmi les clients SIP les plus réputés, citons notamment les suivants :

- X-Lite Free
- Phone Gaim
- Wengo

Ces clients SIP disposent de diverses fonctionnalités améliorées. En choisir un est souvent affaire de goût, selon l'ergonomie du logiciel et les caractéristiques souhaitées (support d'un codec particulier, support de la messagerie instantanée, etc.).

### Serveur d'enregistrement

Deux terminaux peuvent communiquer entre eux sans passer par un serveur d'enregistrement, à la condition que l'appelant connaisse l'adresse IP de l'appelé. Cette contrainte est fastidieuse, car un utilisateur peut être mobile et donc ne pas avoir d'adresse IP fixe, par exemple s'il se déplace avec son terminal ou s'il se connecte avec la même identité à son travail et à son domicile. En outre, l'adresse IP peut être fournie de manière dynamique par un serveur DHCP.

Le serveur d'enregistrement (Registrar Server) offre un moyen de localiser un correspondant avec souplesse, tout en gérant la mobilité de l'utilisateur. Il peut en outre supporter l'authentification des abonnés.

Dans la pratique, lors de l'activation d'un terminal dans un réseau, la première action initiée par celui-ci consiste à transmettre une requête d'enregistrement auprès du serveur d'enregistrement afin de lui indiquer sa présence et sa position de localisation courante dans le réseau. C'est la requête REGISTER, que nous détaillons plus loin, que l'utilisateur envoie à destination du serveur d'enregistrement. Celui-ci sauvegarde cette position en l'enregistrant auprès du serveur de localisation.

L'enregistrement d'un utilisateur est constitué par l'association de son identifiant et de son adresse IP. Un utilisateur peut s'enregistrer sur plusieurs serveurs d'enregistrement en même temps. Dans ce cas, il est joignable simultanément sur l'ensemble des positions qu'il a renseignées.

### Serveur de localisation

Le serveur de localisation (Location Server) joue un rôle complémentaire par rapport au serveur d'enregistrement en permettant la localisation de l'abonné.

Ce serveur contient la base de données de l'ensemble des abonnés qu'il gère. Cette base est renseignée par le serveur d'enregistrement. Chaque fois qu'un utilisateur s'enregistre auprès du serveur d'enregistrement, ce dernier en informe le serveur de localisation.

Presque toujours, le serveur de localisation et le serveur d'enregistrement sont implémentés au sein d'une même entité. On parle alors souvent non pas de serveur de localisation, mais de service de localisation d'un serveur d'enregistrement, tant ces fonctionnalités sont proches et dépendantes.

Les serveurs de localisation peuvent être collaboratifs. Le fonctionnement d'un serveur d'enregistrement est analogue à celui d'un serveur DNS dans le monde Internet : pour joindre un site Internet dont on ne connaît que le nom, il faut utiliser un serveur DNS, qui effectue la conversion (on parle de résolution) du nom en adresse IP. Ce serveur a connaissance d'une multitude d'adresses, qu'il peut résoudre parce qu'elles appartiennent à son domaine ou qu'il a la capacité d'apprendre dynamiquement en fonction des échanges qu'il voit passer. Dès qu'un nom lui est inconnu, il fait appel à un autre DNS, plus important ou dont le domaine est plus adéquat. De la même manière, les serveurs de localisation prennent en charge un ou plusieurs domaines et se complètent les uns les autres.

### **Serveur de redirection**

Le serveur de redirection (Redirect Server) agit comme un intermédiaire entre le terminal client et le serveur de localisation. Il est sollicité par le terminal client pour contacter le serveur de localisation afin de déterminer la position courante d'un utilisateur.

L'appelant envoie une requête de localisation d'un correspondant (il s'agit en réalité d'un message d'invitation, qui est interprété comme une requête de localisation) au serveur de redirection. Celui-ci joint le serveur de localisation afin d'effectuer la requête de localisation du correspondant à joindre. Le serveur de localisation répond au serveur de redirection, lequel informe l'appelant en lui fournissant la localisation trouvée. Ainsi, l'utilisateur n'a pas besoin de connaître l'adresse du serveur de localisation.

### **Serveur proxy**

Le serveur proxy (parfois appelé serveur mandataire) permet d'initier une communication à la place de l'appelant. Il joue le rôle d'intermédiaire entre les terminaux des interlocuteurs et agit pour le compte de ces derniers.

Le serveur proxy remplit les différentes fonctions suivantes :

- localiser un correspondant ;
- réaliser éventuellement certains traitements sur les requêtes ;
- initier, maintenir et terminer une session vers un correspondant.

Lorsqu'un utilisateur demande à un serveur proxy de localiser un correspondant, ce dernier effectue la recherche, mais, au lieu de retourner le résultat au demandeur (comme le ferait un serveur de redirection), il utilise cette réponse pour effectuer lui-même l'initialisation de la communication en invitant le correspondant à ouvrir une session.

Bien que fournissant le même type de service de localisation qu'un serveur de redirection, un serveur proxy va donc plus loin que la simple localisation en initiant la mise en relation des correspondants de façon transparente pour le client. Il peut acheminer tous

les messages de signalisation des terminaux, de l'initialisation de la communication à sa terminaison, en passant par sa modification. En contrepartie, le serveur proxy est une entité beaucoup plus sollicitée que le serveur de redirection, et donc plus lourde.

Chaque terminal peut et devrait en principe disposer d'un tel serveur sur lequel se reposer pour interpréter, adapter et relayer les requêtes. En effet, le serveur proxy peut reformuler une requête du terminal UAC afin de la rendre compréhensible par le serveur auquel s'adresse l'UAC. Cela accroît la souplesse d'utilisation du terminal et simplifie son usage.

Les serveurs proxy jouent aussi un rôle collaboratif, puisque les requêtes qu'ils véhiculent peuvent transiter d'un serveur proxy à un autre, jusqu'à atteindre le destinataire. Notons que le serveur proxy ne fait jamais transiter de données multimédias et qu'il ne traite que les messages SIP.

Le proxy est une entité très souvent utilisée dans la pratique. Par analogie avec l'architecture illustrée à la figure 4.3, symbolisant l'organisation des communications, on parle souvent du *trapèze SIP* pour désigner l'ensemble formé par ces quatre entités.

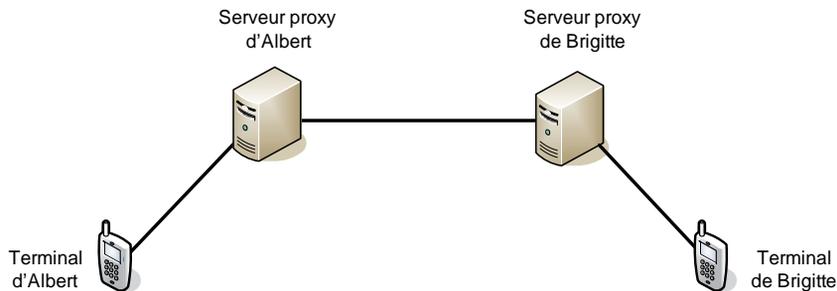


Figure 4.3

*Le trapèze SIP*

On distingue deux types de serveurs proxy :

- Proxy *statefull*, qui maintient pendant toute la durée des sessions l'état des connexions.
- Proxy *stateless*, qui achemine les messages indépendamment les uns des autres, sans sauvegarder l'état des connexions.

Les proxy *stateless* sont plus rapides et plus légers que les proxy *statefull*, mais ils ne disposent pas des mêmes capacités de traitement sur les sessions.

### Mise en place de serveurs SIP

Plusieurs logiciels implémentent les diverses fonctionnalités des serveurs SIP, notamment les suivants :

- SIP Express Router (<http://www.iptel.org/ser/>) ;
- Partysip SIP Proxy Server (<http://www.nongnu.org/partysip/partysip.html>).

Ces deux logiciels libres, respectivement sous licence GPL et LGPL, fonctionnent exclusivement sur plate-forme Linux.

Leur nom est en fait trompeur. Ces deux logiciels sont capables de fournir bien d'autres services que le routage ou la seule fonction de serveur proxy. Ils peuvent être utilisés simultanément comme serveurs de localisation, d'enregistrement, de redirection et de proxy. En outre, ils peuvent délivrer plusieurs services complémentaires, comme l'authentification (avec support de Diameter ou de Radius) ou la création de journaux d'activité (en les couplant à une base SQL ou PostgreSQL, par exemple).

L'installation de ces logiciels est modulaire. Les fonctionnalités qui enrichissent le serveur SIP de base sont disponibles sous forme de plug-in ou d'add-on. Il est donc nécessaire de lancer quelques recherches afin de récupérer les modules que l'on souhaite et les installer séparément. Si l'installation de ces plug-in peut se révéler délicate, le logiciel gagne en souplesse et évolutivité grâce à eux.

Une solution payante et propriétaire de Cisco, Cisco SIP Proxy Server (<http://www.cisco.com/univercd/cc/td/doc/product/voice/sipproxy/>), peut également être installée.

## Se connecter à des réseaux non-IP

SIP a été conçu initialement pour les réseaux à commutation de paquets de type IP, mais ses utilisateurs peuvent aussi joindre des terminaux connectés à des réseaux de nature différente.

Pour cela, il est nécessaire de mettre en place des passerelles (gateways), assurant la conversion des signaux d'un réseau à un autre. On se retrouve dans le cas de figure évoqué au chapitre précédent pour le protocole H.323, et nous verrons plus loin que le protocole MGCP propose une manière de gérer ces fonctionnalités.

L'appel dans l'autre sens, c'est-à-dire d'un réseau non-IP vers un réseau à commutation de paquets, est tout aussi envisageable, à la seule condition que le terminal appelant dispose de la capacité d'entrer l'adresse de son correspondant SIP.

Cette adresse n'est généralement pas constituée uniquement de numéros, alors que la majorité des téléphones traditionnels actuels sont dépourvus de clavier. Plusieurs possibilités permettent de contourner cette difficulté, notamment la reconnaissance audio, la saisie d'une adresse à la manière d'un SMS ou l'attribution de numéros aux correspondants SIP.

## L'adressage SIP

L'objectif de l'adressage est de localiser les utilisateurs dans un réseau. C'est une des étapes indispensables pour permettre à un utilisateur d'en joindre un autre.

Pour localiser les utilisateurs, il faut pouvoir les identifier de manière univoque. SIP propose des moyens très performants pour nommer les utilisateurs, grâce au concept d'URI, classique sur Internet, que nous allons détailler avant de voir son utilisation par SIP.

## URI (Universal Resource Identifier)

Un URI définit une syntaxe permettant de désigner de manière unique, formelle et normalisée une ressource, qu'il s'agisse d'un document textuel, audio, vidéo ou plus généralement d'une entité logique ou physique.

Une ressource décrite par un URI peut être déplacée ou même supprimée. L'URI correspondant n'en conserve pas moins de manière permanente la valeur descriptive de la ressource.

Considérons un exemple. Deux personnes portant le même nom de famille et le même prénom sont susceptibles d'être confondues si on les recherche dans un annuaire. En plus du nom de la personne, qui peut être partagé par d'autres, son âge, sa profession ou sa localisation sont des paramètres susceptibles d'évoluer et qui ne constituent donc pas des propriétés discriminantes. L'attribution d'un identifiant unique à chaque individu assure une identité unique et permet de le différencier des autres avec certitude.

C'est, par analogie, toute la vocation d'un numéro de Sécurité sociale. À la syntaxe près, un numéro de sécurité sociale est une forme d'URI. Une adresse e-mail est également une forme d'URI.

La figure 4.4 illustre quelques exemples d'attributs non discriminants et discriminants qui peuvent constituer ou non des URI.

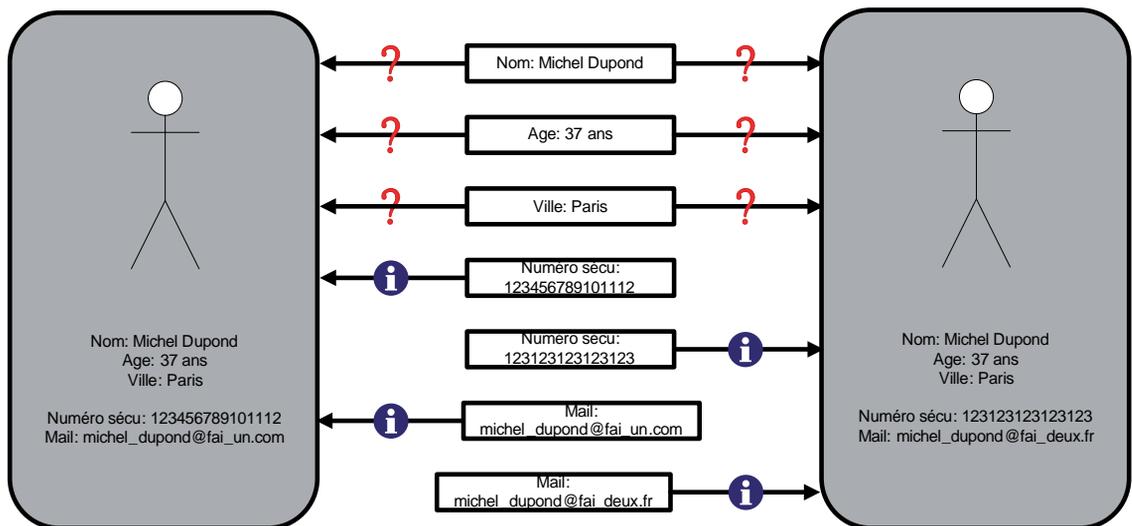


Figure 4.4

*Paramètres non discriminants et discriminants*

Un URI est formé d'une chaîne de caractères. Sa syntaxe a été définie au CERN (Centre européen pour la recherche nucléaire) de Genève, par Tim Berners-Lee dès 1989, dans le cadre du système d'hyperliens (liens hypertextes) qu'il proposait la même année. Cette syntaxe a été normalisée par l'IETF en août 1998 dans la RFC 2396 puis révisée de nombreuses fois, notamment dans la RFC 2396bis, et reprise en janvier 2005 dans la RFC 3986.

Tim Berners-Lee est considéré comme l'inventeur du World-Wide Web. C'est lui qui a conçu le premier serveur HTTP et le premier navigateur Web, qu'il baptisa « WorldWideWeb » après avoir pensé l'appeler « Information Mesh » ou encore « Information Mine ». Il a refusé de nombreuses propositions d'embauches émanant d'importants groupes industriels pour fonder en 1994 et diriger depuis le consortium W3C (World-Wide Web). Cette instance est chargée de superviser les nouvelles normes mises en œuvre dans Internet afin de permettre son évolution et de garantir son interopérabilité. Avec plus de 500 membres, il regroupe les plus gros éditeurs du secteur des hautes technologies, parmi lesquels Microsoft, Sun et IBM.

Bien qu'étant un consortium, le W3C n'a pas le pouvoir de normaliser des protocoles. Ses recommandations ont cependant un impact important et ont tendance à être suivies par l'ensemble des acteurs du marché.

Les URL (Uniform Resource Locator), que l'on manipule couramment dans l'adressage Web pour joindre un site Internet, constituent un sous-ensemble des URI. Elles ont pour fonction de spécifier une localisation relative à une ressource (par exemple *www.ietf.org*), ainsi que la méthode permettant d'y accéder (par exemple *http*, *ftp*, etc.).

À la différence d'un URI, une URL se contente d'apporter une localisation et non une définition de la ressource. Ainsi, un même document peut se trouver à deux emplacements différents, donc à deux URL différentes dans le réseau Internet, alors qu'il fait référence à une même ressource.

### **Format des adresses SIP**

Tout utilisateur SIP dispose d'un identifiant unique. Cet identifiant constitue l'adresse de l'utilisateur permettant de le localiser.

Le format d'une adresse SIP (ou URL SIP) respecte la RFC 3986 (nommée Uniform Resource Identifier: Generic Syntax) et se présente sous la forme illustrée à la figure 4.5.

**Figure 4.5**

*Syntaxe d'une adresse SIP*

```
sip : identifiant[:mot_de_passe]@serveur[?paramètres]
```

Les parties entre crochets sont optionnelles.

On distingue dans cette adresse plusieurs parties :

- Le mot-clé *sip* spécifie le protocole à utiliser pour la communication. Par analogie avec le Web (où une page est référencée par une adresse de type *http://monsie*), le mot-clé *sip* précise que ce qui va suivre est l'adresse d'un utilisateur.
- La partie *identifiant* définit le nom ou le numéro de l'utilisateur. Cet identifiant est nécessairement unique pour désigner l'utilisateur de manière non ambiguë.
- La partie *mot\_de\_passe* est facultative. Le mot de passe peut être utile pour s'authentifier auprès du serveur, notamment à des fins de facturation. C'est aussi un moyen pour joindre un utilisateur qui a souhaité s'enregistrer sur l'équivalent d'une liste rouge : sans la connaissance de ce mot de passe, le correspondant n'est pas joignable. De manière générale, cette possibilité offre le moyen de restreindre l'utilisation de certains services.
- La partie *serveur* spécifie le serveur chargé du compte SIP dont l'identifiant précède l'arobase. Le serveur est indiqué par son adresse IP ou par un nom qui sera résolu par DNS. Des paramètres URI peuvent être associés à ce nom. C'est ce serveur qui sera contacté pour joindre l'abonné correspondant. Un port peut être spécifié à la suite du serveur.
- La partie *paramètres* est facultative. Les paramètres permettent soit de modifier le comportement par défaut (par exemple, en modifiant les protocoles de transport ou les ports, ou encore le TTL par défaut), soit de spécifier des informations complémentaires (par exemple, l'objet d'un appel qui sera envoyé à l'appelé en même temps que l'indication d'appel, à la manière d'un e-mail précisant l'objet du message).

Le tableau 4.1 fournit quelques exemples d'adresses SIP commentées.

**Tableau 4.1 – Exemples d'adresses SIP**

Adresse SIP	Commentaire
<sip:guy.laurent@123.123.123.123>	C'est le format le plus commun. L'identifiant de l'utilisateur est spécifié par un nom ou un pseudonyme, <i>guy.laurent</i> . Après l'arobase est spécifiée l'adresse IP du serveur en charge de la gestion du compte de <i>guy.laurent</i> . Cette adresse IP étant fixe, il n'est pas nécessaire de la résoudre par un DNS, et il est possible de contacter directement ce serveur. L'IP fixe n'est généralement pas pratique, car une adresse fixe oblige le fournisseur d'accès à conserver ses mécanismes d'adressage ou à avertir ses utilisateurs de toute modification.
<sip:+33145555555:mon_pass123@ma_passerelle_rtc>	Le premier nombre (+33145555555) est le numéro de téléphone du correspondant. On peut supposer qu'il s'agit d'un numéro géographique et que le correspondant est actif dans le réseau RTC. Pour joindre ce réseau, il faut passer par une passerelle, donnée juste après l'arobase, dont le nom est <i>ma_passerelle_rtc</i> . L'utilisation d'un mot de passe ( <i>mon_pass123</i> ) permet à l'appelant de s'authentifier auprès du serveur <i>ma_passerelle_rtc</i> pour avoir le droit d'émettre l'appel (notamment pour la facturation).

Tableau 4.1 – Exemples d’adresses SIP (*suite*)

Adresse SIP	Commentaire
<sip:guy.laurent@sip_ietf.org:12345?subject=Confirmation_RendezVous;transport=tcp:54321>	<p>Cette adresse est semblable à la première, mais avec des paramètres supplémentaires. Elle comporte un nom d'utilisateur (toujours <i>guy.laurent</i>) et un serveur à contacter pour être mis en contact avec l'utilisateur. Le serveur étant nommé <i>sip_ietf.org</i>, son nom devra être résolu par un DNS afin de déterminer son adresse IP. Il sera contacté sur le port 12345.</p> <p>Cette adresse fournit les informations complémentaires suivantes :</p> <ul style="list-style-type: none"> <li>– Un sujet, qui indique le motif de l'appel : <i>Confirmation_RendezVous</i>. En même temps que le terminal appelé sonnera, le nom de l'appelant et le sujet de son appel pourront être affichés sur le terminal, sous réserve que ce dernier supporte ce service.</li> <li>– Un protocole de transport imposé : <i>tcp</i>. Par défaut, c'est le protocole UDP qui est utilisé dans les communications.</li> <li>– Un port à utiliser pour la communication : <i>54321</i>. Par défaut, c'est le port 5060 qui est utilisé.</li> </ul>

On retiendra deux avantages de l'adressage SIP :

- L'adressage est indépendant de la localisation géographique des abonnés. SIP est conçu pour assurer la mobilité de ses utilisateurs, et donc permettre de joindre quelqu'un avec une adresse SIP unique, quels que soient sa localisation et son terminal. Le réseau peut toutefois adopter un plan de numérotation selon n'importe quel critère, comme la localisation géographique, sans que cela soit gênant.
- Un utilisateur peut avoir plusieurs adresses SIP aboutissant toutes au même terminal. Par exemple, si quelqu'un souhaite différencier son adresse SIP professionnelle de son adresse SIP personnelle, il peut utiliser un même terminal référencé sur deux adresses distinctes. Il lui est alors possible d'activer la messagerie de son compte personnel pendant son travail et, le week-end, de rediriger les appels sur son adresse professionnelle vers un centre de permanence. Le tout en utilisant un terminal unique.

Ce mécanisme d'adressage particulièrement souple permet de supporter la mobilité des utilisateurs et le monde Internet.

### **Localisation et résolution d'une adresse SIP**

D'une manière générale, une adresse SIP spécifie un utilisateur et un nom de domaine. Pour localiser l'utilisateur, il faut d'abord contacter le serveur gérant le domaine puis solliciter ce serveur pour déterminer la position de l'utilisateur.

Si la partie indiquant le serveur de domaine contient une adresse IP, ce serveur est joint directement. À défaut, l'adresse IP du serveur sera déterminée après une résolution DNS. La demande de localisation de l'utilisateur s'effectue auprès du serveur de domaine ainsi contacté. La position de l'utilisateur peut être référencée de manière absolue ou relative.

Le cas simple consiste en une position absolue, spécifiant une adresse IP localisant l'utilisateur. Le cas plus complexe consiste en une position relative, spécifiant une autre adresse SIP. Dans ce cas, il faut répéter l'opération de résolution de cette nouvelle adresse SIP depuis le début : contacter le serveur SIP gérant le domaine puis lui demander la position de l'utilisateur.

Dans le cas illustré à la figure 4.6, il s'agit de localiser l'utilisateur *David* à partir de son adresse SIP *david.dad@dom\_A.fr*. Cet utilisateur possède une autre adresse SIP, qui est *dav@dom\_B.fr*. L'une correspond à une adresse professionnelle, l'autre à une adresse personnelle. Naturellement, l'utilisateur souhaite rester joignable sur ces deux adresses simultanément.

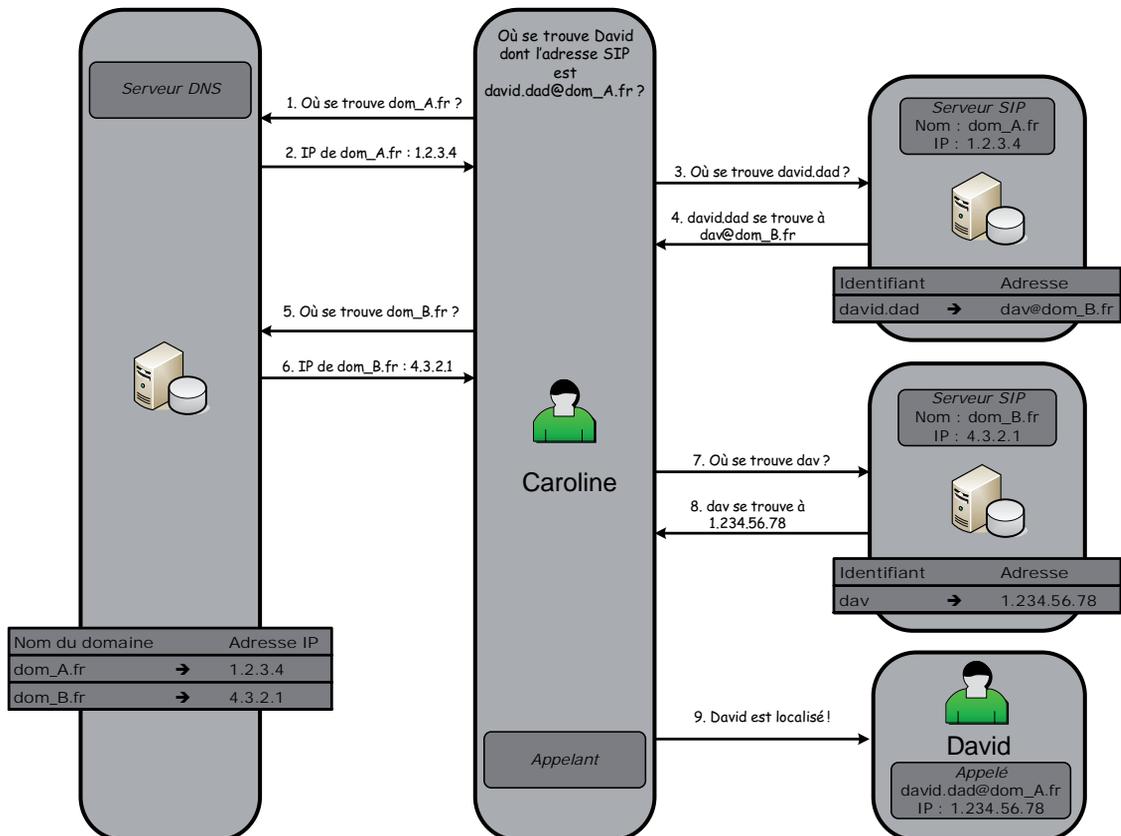


Figure 4.6

Principe de localisation à partir d'une adresse SIP

Pour que l'exemple soit valide, il faut considérer que, préalablement à la localisation, l'utilisateur *David* s'est enregistré auprès du serveur SIP gérant le domaine *dom\_A.fr*

en lui fournissant son identifiant (*david.dad*) et une adresse SIP associée (*dav@dom\_B.fr*).

Notons que le serveur indiqué dans la partie domaine d'une adresse SIP peut être un serveur SIP de type proxy ou une passerelle permettant de joindre des utilisateurs d'un réseau non-IP.

## Les messages SIP

Les messages SIP sont décrits dans la RFC 822, qui définit la syntaxe à la fois des requêtes et des réponses. On y trouve une très forte influence des autres protocoles de l'IETF, principalement HTTP et SMTP. Le format des requêtes et réponses est en effet similaire à celui utilisé dans le protocole HTTP, et les en-têtes s'apparentent à celles utilisées dans le protocole SMTP. On y retrouve par ailleurs le concept d'URL.

Les sections qui suivent détaillent le « langage SIP », afin de montrer comment s'effectuent les communications entre les interlocuteurs. L'objectif est d'apprendre à la fois à décrypter les échanges entre les intervenants et à écrire une application SIP.

### Notion de transaction

Une communication est constituée d'une succession de transactions par le biais d'échanges de messages, qui peuvent être soit des requêtes, soit des réponses à des requêtes.

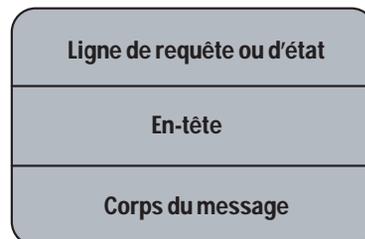
Une transaction SIP peut s'entendre en première approximation selon le sens commun : un émetteur formule une demande à un récepteur. Ce dernier étudie les conditions de la demande avant de répondre. Éventuellement, il peut être amené à envoyer des réponses temporaires, indiquant l'évolution du traitement de la requête.

Une transaction est donc constituée d'une requête et de sa ou ses réponses. Pour se mettre d'accord sur la nature de l'échange, chaque intervenant est susceptible de négocier les paramètres de la session au moyen de nouvelles transactions.

Les messages SIP sont codés en utilisant la syntaxe de message HTTP/1.1 (RFC 2068). Qu'il soit une requête ou une réponse à une requête, un message SIP doit respecter le format illustré à la figure 4.7.

**Figure 4.7**

*Format générique d'un message SIP*



La première partie est soit une ligne de requête, s'il s'agit d'une requête, soit une ligne d'état, s'il s'agit d'une réponse. La seconde partie rassemble les en-têtes du message. Enfin, vient le corps du message. Optionnel, celui-ci est précédé d'une balise de retour chariot et saut de ligne CR-LF (Carriage Return-Line Feed) afin d'indiquer le début du corps du message. Cette balise assure la séparation de l'en-tête et du corps du message, ce qui permet d'optimiser le temps de traitement des messages.

La séparation des champs réduit le temps de transit des messages dans le réseau. De cette manière, les serveurs intermédiaires peuvent rapidement discerner les informations utiles, sans avoir à manipuler les données du corps du message.

La spécification du protocole limite le nombre d'en-têtes possible, ce qui permet de borner le temps de lecture ou d'écriture des messages. En théorie, 36 en-têtes peuvent être utilisés au maximum. Dans la pratique, guère plus d'une dizaine est utilisée simultanément.

## Paramètres généraux pour les requêtes et les réponses

Certains paramètres généraux peuvent être partagés à la fois par les messages de requête et par les messages de réponse.

Les sections suivantes détaillent les caractéristiques spécifiques à la constitution des requêtes puis à celles des réponses.

### En-têtes d'un message

Un message de requête comme un message de réponse peut contenir des en-têtes.

Les en-têtes les plus couramment utilisés dans les messages SIP sont les suivants :

- En-têtes généraux, qui peuvent être utilisés indifféremment pour des messages de requête ou des messages de réponse.
- En-têtes de requête, exclusivement employés pour les messages de requête.
- En-têtes de réponse, exclusivement employés pour les messages de réponse.
- En-têtes d'entité, qui donnent des informations descriptives sur le corps du message.

Le tableau 4.2 récapitule les principaux champs d'en-têtes, classés selon ces quatre catégories. Toutes les méthodes SIP ne supportent pas forcément tous ces champs.

**Tableau 4.2 – Principaux champs d'en-têtes des messages SIP**

Champ d'en-tête	Commentaire
<b>En-têtes généraux</b>	
ACCEPT	Ce champ est utilisé dans les messages INVITE, OPTIONS et REGISTER afin de spécifier le format qui devra être supporté en réponse. Par exemple, <i>accept: application/sdp</i> signifie que le corps de la réponse à ce message devra être de l'applicatif codé avec le langage SDP (valeur par défaut de ce champ).
ACCEPT-ENCODING	Spécifie le type d'encodage textuel accepté dans le corps du message de la réponse du client. Par exemple, <i>Accept-Encoding:gzip</i> signifie que le récepteur pourra utiliser le format gzip pour compresser le corps de son message.

Tableau 4.2 – Principaux champs d'en-têtes des messages SIP (*suite*)

Champ d'en-tête	Commentaire
ACCEPT-LANGUAGE	Spécifie le langage accepté dans le corps du message de la réponse du client. Par exemple, <i>Accept-Language:fr</i> signifie que le corps du message de réponse pourra être exprimé en français.
CALL-ID	Indique le numéro d'identification d'un appel ( <i>voir plus loin</i> ). Par exemple, <i>Call-id:13d11a45d@ietf.org</i> identifie une session.
CSEQ	Indique le numéro d'une commande unique ( <i>voir plus loin</i> ). Par exemple, <i>CSeq:15197903 INVITE</i> identifie la commande INVITE avec un numéro de commande associé de valeur 15197903.
FROM	Indique l'adresse SIP de l'émetteur du message. Par exemple, <i>From:Nathalie&lt;sip:nathalie@domaine_ietf.org&gt;</i> identifie l'appelant Nathalie avec l'adresse fournie.
TO	Indique l'adresse SIP du récepteur du message. Par exemple, <i>To:Henri&lt;sip:henri@domaine_ietf.org&gt;</i> identifie l'appelé Henri avec l'adresse fournie.
VIA	Liste l'ensemble des nœuds parcourus par le message ( <i>voir plus loin</i> ). Par exemple : <i>VIA:SIP/2.0/UDP 142.132.227.111:5060</i> <i>VIA:SIP/2.0/UDP proxy-lambda:5060</i> <i>VIA:SIP/2.0/UDP émetteur:5060</i> permet de savoir que le message est passé par l'émetteur avant de traverser le nœud dont le nom est <i>proxy-lambda</i> puis le nœud dont l'adresse est <i>142.132.227.111</i> .
DATE	Spécifie la date et l'heure d'envoi du message. Cette donnée est exprimée au format GMT (Greenwich Mean Time) et est sensible à la casse. Par exemple, <i>Date:Thu, 15 Mar 2012 11:35:10 GMT</i> définit la date du jeudi 15 mars 2012, à 11 heures 35 minutes et 10 secondes, heure GMT.
EXPIRES	Spécifie la durée au bout de laquelle le message peut être effacé. Cette durée est indiquée en seconde. Par exemple, <i>Expires:3</i> définit une durée de 3 secondes avant que le message soit détruit.
<b>En-têtes de requête</b>	
SUBJECT	Indique l'objet de l'appel. Par exemple, <i>subject:anniversaire surprise de David demain !</i> définit un sujet indiquant le message spécifié à l'appelant lorsque son téléphone sonne.
PRIORITY	Indique la priorité d'une session, c'est-à-dire l'importance qui devrait être accordée à l'appel sollicité. On dénombre quatre priorités par ordre décroissant d'importance : EMERGENCY, URGENT, NORMAL, NON-URGENT. La priorité EMERGENCY est réservée à une situation de danger vitale et ne devrait pas être exploitée pour des messages d'une autre nature. Par exemple, <i>priority=urgent</i> caractérise un message d'importance classée urgente.
USER-AGENT	Fournit des informations sur le logiciel utilisé par le terminal UAC. Cette information peut cependant être une source de vulnérabilité si elle est divulguée à des personnes mal attentionnées. Les logiciels offrent généralement la possibilité de masquer cette information. Par exemple, <i>Server:Lite Sip Communicator 1.6</i> signifie que l'UAC utilise le logiciel Lite Sip Communicator 1.6 pour appeler.
<b>En-têtes de réponse</b>	
RETRY-AFTER	Est utilisé conjointement avec une réponse d'indisponibilité d'un correspondant pour spécifier le temps (exprimé en seconde) au bout duquel il convient de renouveler l'appel. Si aucune valeur n'est donnée, le serveur contacté est considéré comme indisponible pendant une durée indéterminée. Par exemple, <i>Retry-After:300</i> signifie qu'il faut rappeler 5 minutes après l'heure où le message a été envoyé. On peut également fournir un motif explicatif textuel à l'indisponibilité du contact, par exemple : <i>Retry-after:3600 (je suis allé déjeuner)</i> .
SERVER	Fournit des informations sur le logiciel utilisé par le terminal UAS. De même que pour l'en-tête USER-AGENT, il s'agit d'une donnée potentiellement sensible. Par exemple, <i>Server:Sip Phone 1.5.2</i> signifie que le terminal UAS utilise le logiciel Sip Phone 1.5.2 pour répondre à l'appel.

Tableau 4.2 – Principaux champs d’en-têtes des messages SIP (*suite*)

Champ d'en-tête	Commentaire
<b>En-têtes d'entité</b>	
CONTENT-TYPE	Indique le format utilisé dans le corps et le langage qui le décrit. Par exemple, <i>content-type:text/html</i> signifie que le corps du message contient du texte codé en langage HTML. De même, <i>content-type:application/sdp</i> indique que le corps du message contient de l'appliatif codé en langage SDP.
CONTENT-LENGTH	Indique la taille du corps du message exprimée en octets. Par exemple, <i>Content-Lenght:155</i> signifie que le corps du message fait 155 octets.
CONTENT-ENCODING	Indique le format de compression utilisé dans le corps du message. La compression est optionnelle. Par exemple, <i>Content-Encoding:gzip</i> signifie que le corps du message est compressé selon le format gzip.
CONTENT-LANGUAGE	Indique la langue utilisée dans le corps du message. Par exemple, <i>Content-Language:fr</i> signifie que le corps du message est en français.

### Le champ VIA pour détecter les boucles lors du routage

Le champ d'en-tête VIA permet d'indiquer le chemin parcouru par un message entre un émetteur et son récepteur.

Ce chemin désigne l'ensemble des nœuds intermédiaires par lesquels transite le message. Chaque nœud qui reçoit le message enrichit ce champ en y ajoutant son adresse réseau. Plus précisément, un nouveau champ VIA est ajouté pour chaque nœud traversé, contenant le type de protocole de transport utilisé, l'adresse IP (ou le nom à résoudre par DNS) du nœud et éventuellement le port sur lequel tourne l'application SIP. Le protocole de transport peut être choisi parmi UDP, TCP, TLS ou SCTP.

Un exemple de champ VIA peut être (les espaces sont tolérées) :

```
Via:SIP / 2.0 / TCP   nœud_de_transit:34567
```

Toutes les entités qui reçoivent ce message et sont chargées de le faire suivre jusqu'à son destinataire doivent ajouter une ligne similaire.

Grâce à ce champ VIA, les boucles peuvent facilement être détectées par les serveurs proxy traversés. Dans ce cas, en effet, le serveur qui marque la traversée du message en ajoutant un nouveau champ VIA trouve sa propre adresse dans un autre champ VIA. Il en déduit que le routage comporte une boucle et en avertit l'émetteur ou tente une autre route.

Une autre caractéristique de ce champ est de permettre au récepteur de retracer le chemin inverse lors de sa réponse, de manière que son message suive la même route. En effet, la réponse contient les mêmes champs VIA que la requête. Chaque serveur proxy qui reçoit le message de réponse retire sa propre adresse du champ VIA du message, puis envoie le message à l'adresse suivante. De proche en proche, la réponse suit la même route que la requête, et il n'est pas nécessaire de recourir à des serveurs DNS ou de gérer des états des liens de chaque connexion dans les serveurs proxy.

## Différence entre Call-Id et CSeq

Le Call-Id, ou identifiant d'appel, est un numéro identifiant une session particulière. Tous les messages liés à une même session portent obligatoirement un même identifiant d'appel. De cette manière, on peut décrire une session par son Call-Id associé.

Tous les utilisateurs d'une même session exploitent donc le même Call-Id, y compris lors d'une conférence. Cela vaut aussi pour les serveurs intervenant dans la signalisation des messages SIP, ces derniers distinguant les sessions auxquelles un message fait référence par leur identifiant d'appel.

Le Call-Id est généré lors de l'initialisation de l'appel par le terminal client. Par exemple, un Call-Id peut être : *Call-Id:a1-b2-c3-45@ietf.org*. Cet identifiant est reporté dans tous les messages SIP liés à la session, et ce quel que soit leur émetteur.

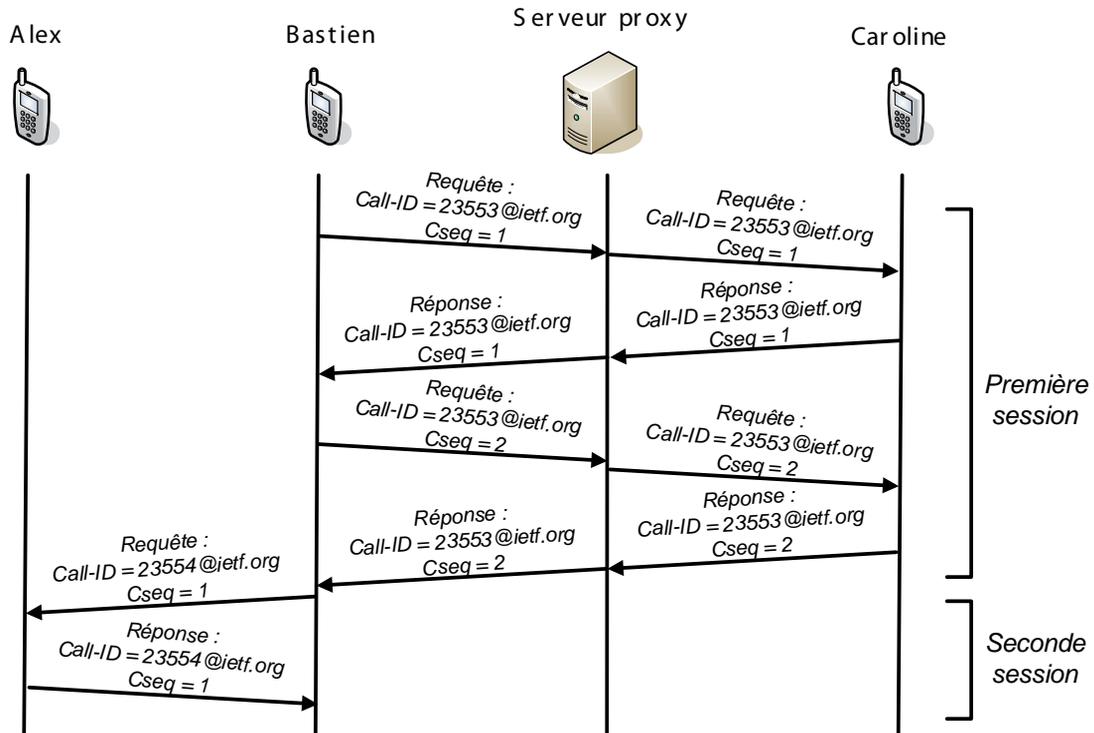


Figure 4.8

Différences entre Call-Id et CSeq

Une communication peut toutefois être répartie entre plusieurs sessions. Par exemple, une conférence peut mettre en place une session audio et une session vidéo distinctes. Dans ce cas, chacune des sessions dispose d'un identifiant d'appel différent.

Le CSeq (Command Sequence), ou ordre de commande, est constitué de l'association d'un numéro et de la requête correspondante. Il permet de différencier chaque message de requête. Lorsqu'un terminal en émet plusieurs au cours d'une même session, il distingue à quelle requête fait référence une réponse grâce à l'ordre de commande CSeq. Deux CSeq ne sont égaux que s'ils ont simultanément le même numéro et la même requête.

Un exemple de CSeq pourrait être : *CSeq:1 INVITE* pour le premier message d'initialisation d'une session (méthode INVITE). Ensuite, ce numéro est repris identiquement pour les messages qui relaient la requête et pour les messages qui fournissent une réponse. À la prochaine requête, l'ordre de commande est incrémenté d'une unité (passant donc à 2), et ainsi de suite.

La figure 4.8 illustre la différence entre Call-Id et CSeq. Dans cet exemple, Bastien initie une communication avec Caroline (en passant par un serveur proxy). Au départ, le Call-Id est arbitrairement fixé à *23553@ietf.org* et le CSeq débute à 1. Le CSeq ne sera incrémenté qu'avec une nouvelle requête. Quant au Call-Id, il ne changera que plus tard lorsque Bastien lancera une autre session vers Alex (en liaison directe). Par souci de simplification, les messages de signalisation et la communication multimédia entre les correspondants n'ont pas été représentés.

### **Abréviation des en-têtes de messages**

Les en-têtes SIP sont indiqués juste après la ligne de requête ou d'état. Aucun ordre n'est imposé. Les en-têtes se succèdent, simplement séparés par des points-virgules.

Chaque paramètre d'en-tête est précédé du champ qui lui correspond, en respectant la syntaxe générique suivante (les espaces sont tolérées) :

*entete1:valeur1; entete2:valeur2; entete3:valeur3; ...*

Par exemple, pour spécifier que le Call-Id d'un message a pour valeur *170451@ietf.org*, on écrit : *call-id:170451@ietf.org*, et ainsi de suite pour chaque paramètre, indépendamment de leur ordre d'apparition.

Cette succession de paramètres alourdit cependant la taille du message, qui peut rapidement devenir très importante. Cela pose d'autant plus problème si les paquets doivent respecter une taille limitée par une MTU (Maximum Transmission Unit). La MTU impose en effet une valeur au-delà de laquelle le paquet doit être segmenté.

Pour éviter cette segmentation et réduire globalement la taille des envois, le protocole SIP permet d'utiliser des abréviations, dites formes compactes, pour représenter les en-têtes les plus courants.

Une forme compacte s'écrit en une lettre seulement. Pour la même information de Call-Id que précédemment, on peut écrire : *i:170451@ietf.org*, la lettre *i* se substituant à *call-id*. Notons toutefois que tous les en-têtes ne disposent pas d'une forme compacte, qui reste l'apanage des plus usuels d'entre eux.

Le tableau 4.3 récapitule quelques-unes de ces abréviations. Il est possible de mélanger dans un même message les formats longs avec les formats compacts des en-têtes. Toutes les entités SIP doivent savoir interpréter les deux formes.

**Tableau 4.3 – Abréviations de certains champs d'en-têtes**

Format complet	Format compact
CALL-ID	i
FROM	f
TO	t
VIA	v
SUBJECT	s
CONTENT-TYPE	c
CONTENT-LENGTH	l
CONTENT-ENCODING	e

### Corps d'un message

Le corps d'un message SIP contient le descriptif complet des paramètres de la session concernée.

Typiquement, une description de la session à ouvrir comporte les informations suivantes :

- informations générales sur la session (nom de la session, date de la session, objet de la session, etc.) ;
- informations sur l'émetteur du message (nom, e-mail, numéro de téléphone, etc.) ;
- informations réseau (ressources nécessaires, protocole et port utilisés pour le transport des données multimédias, etc.) ;
- liste des flux multimédias utilisés (audio, vidéo, texte) ;
- liste des codages supportés (G.711, G.729, H.216, MPEG, etc.) ;
- informations de sécurité (type de cryptage utilisé).

Ces informations sont fournies soit pour être négociées avec le correspondant, soit pour être fixées au terme de la négociation. Elles permettent de s'accorder préalablement à l'échange sur la compatibilité des terminaux.

C'est l'en-tête qui précise dans quel langage sont spécifiées les informations données dans le corps du message (avec le champ CONTENT-TYPE). Le plus couramment, ces spécifications s'effectuent soit sous forme textuelle, en utilisant le langage HTML, soit sous forme applicative, en utilisant le langage protocolaire SDP.

## SDP (Session Description Protocol)

Décrit dans la RFC 2327, le protocole SDP a été conçu par l'IETF dans le cadre de MMUSIC, le groupe de travail à l'origine du protocole SIP.

Son rôle est de décrire l'ensemble des paramètres constituant une session. Cela inclut la spécification des médias utilisés, des protocoles de transport, des ports, des codages, des ressources nécessaires et des dates d'activité de la session.

Avec SDP, la syntaxe des spécifications du corps respecte le modèle suivant :

*paramètre = valeur*

Les paramètres peuvent appartenir à deux catégories distinctes :

- Globale : le paramètre s'applique à l'ensemble de la session (paramètre de média).
- Locale : le paramètre ne s'applique qu'à un média particulier (paramètre de média).

Certains champs de paramètres sont propres à l'une ou l'autre de ces catégories, tandis que d'autres peuvent être employés indifféremment dans les deux catégories. Seul l'ordre d'apparition permet de distinguer le champ d'application des paramètres.

Chaque paramètre est caractérisé par une lettre.

Le tableau 4.4 récapitule quelques-uns des champs possibles définis avec SDP.

**Tableau 4.4 – Champs SDP les plus courants**

Champ SDP	Correspondance	Type d'information	Descriptif	Présence du champ
v	PROTOCOL VERSION	Description de session	Version du protocole	Requise
o	OWNER / CREATOR AND SESSION IDENTIFIER	Description de session	Nom du créateur de la session et identification de la session	Requise
s	SESSION NAME	Description de session	Nom de la session	Requise
i	MEDIA TITLE	Description de session et de média	Information sur la session	Optionnelle
u	URI	Description de session	URI de description de la session	Optionnelle
e	EMAIL ADDRESS	Description de session	E-mail du créateur de la session	Optionnelle
p	PHONE NUMBER	Description de session	Numéro de téléphone du créateur de la session	Optionnelle
c	CONNECTION INFORMATION	Description de session et de média	Adresse réseau avec laquelle s'effectue la connexion.	Requise soit dans la description de la session, soit dans toutes les descriptions de média
b	BANDWIDTH INFORMATION	Description de session et de média	Débit nécessaire	Optionnelle
t	TIME	Description temporelle	Date d'activité de la session	Requise

Tableau 4.4 – Champs SDP les plus courants (*suite*)

Champ SDP	Correspondance	Type d'information	Descriptif	Présence du champ
r	REPEAT TIMES	Description temporelle	Répétition de la session. Cette durée commence à partir de la valeur de début de session indiquée dans le champ t.	Optionnelle
z	TIME ZONE ADJUSTMENTS	Description de session	Information horaire	Optionnelle
k	ENCRYPTION KEY	Description de session et de média	Clé de cryptage utilisée	Optionnelle
a	SESSION ATTRIBUTE	Description de session et de média	Attributs de média	Optionnelle
m	MEDIA NAME AND TRANSPORT ADDRESS	Description de média	Type de média utilisé et adresse de transport	Requise

Les paramètres doivent impérativement respecter l'ordre donné dans le tableau. On trouve d'abord les paramètres généraux (caractérisant la session), puis les paramètres particuliers (qui s'appliquent seulement à un média, lequel est défini par le champ *m*).

Les paramètres particuliers décrivant un média sont mentionnés après le champ *m*. Tout paramètre indiqué avant le champ *m* sera considéré comme décrivant une session tout entière. Autrement dit, les champs SDP qui peuvent décrire à la fois une session et un média sont interprétés selon leur emplacement dans le corps du message. Avant le champ *m*, ils s'appliquent à une session. À la suite du champ *m*, ils ne s'appliquent qu'au média décrit par le champ *m*. À la suite de ce champ *m*, on peut donc reprendre les paramètres du tableau ayant pour type la description d'une session et qui s'appliqueront cette fois pour décrire uniquement le média mentionné par le champ *m*.

Un exemple de corps de message SDP peut être :

```
v=0
o=lenzo 2890844526 2890842807 IN IP4 132.227.60.4
s=Message test
i=Ce message permet d'analyser quelques champs SDP courants
u=http://www.monsitesurlavoip.com/docs/protocole_sdp.doc
e=Laurent.guy@monsitesurlavoip.com (Laurent Guy)
p=+33-061-770-5555
c=IN IP4 132.227.60.12/125
t=2873397496 2873404696
a=sendrecv
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31 32
m=application 32416 udp wb
a=orient:portrait
```

Le tableau 4.5 reprend cet exemple de message SDP en le commentant.

**Tableau 4.5 – Exemple de message SIP complet commenté**

Ligne de message	Commentaire
<code>v=0</code>	Indique que c'est la version 0 du protocole SDP qui est utilisée dans ce message.
<code>o=lenzo 2890844526 2890842807 IN IP4 132.227.60.4</code>	<p>Cette ligne respecte le format <code>&lt;nom d'utilisateur&gt; &lt;identifiant de session&gt; &lt;version de l'annonce&gt; &lt;type de réseau&gt; &lt;format de l'adresse réseau&gt;&lt;adresse réseau&gt;</code>, avec les caractéristiques suivantes :</p> <ul style="list-style-type: none"> <li>– Le nom d'utilisateur indique l'identifiant de la personne à l'origine du message. Le tiret (caractère -) peut être utilisé pour ne pas préciser d'identifiant.</li> <li>– L'identifiant de session est une chaîne de caractères numériques définie de manière que, hormis le paramètre de version, les cinq autres éléments constituant le champ <code>o</code> (c'est-à-dire nom d'utilisateur, identifiant de session, type de réseau, format de l'adresse réseau et adresse réseau) forment une identification unique. Un utilisateur devra modifier la valeur de ce paramètre lors d'une nouvelle session, tandis qu'un autre utilisateur pourra utiliser le même identifiant de session. Généralement, mais pas obligatoirement, la valeur de ce champ est donnée par le protocole NTP, qui fournit l'heure suivant différents formats, ici en seconde.</li> <li>– La version de l'annonce est attribuée afin de déterminer l'antériorité des messages. Ainsi, si une entité reçoit plusieurs messages d'annonce affectés à une même session, elle considérera le plus récent d'entre eux, c'est-à-dire celui qui possède le numéro de version le plus grand. Là encore, il est recommandé, sans que ce soit obligatoire, d'utiliser le protocole NTP pour donner une valeur à ce champ.</li> <li>– Le format de l'adresse qui va suivre est spécifié par <code>IP4</code>, qui indique que c'est la version 4 du protocole IP qui est utilisée (le code <code>IP6</code> renvoie à IPv6).</li> <li>– Le type de réseau utilisé est généralement Internet, qui est mentionné par le code <code>IN</code>.</li> <li>– L'adresse IP qui suit est celle du terminal à l'origine de l'annonce.</li> </ul>
<code>s=Message test</code>	Spécifie le nom donné à la session.
<code>i=Ce message permet d'analyser quelques champs SDP courants</code>	Message informatif décrivant la session
<code>u=http://www.monsitesurlavoip.com/docs/protocole_sdp.doc</code>	L'URI indique un pointeur sur lequel des informations concernant la session courante peuvent être trouvées.
<code>e=Laurent.guy@monsitesurlavoip.com (Laurent Guy)</code>	L'e-mail de la personne à l'origine du message peut être mentionné indifféremment sous la forme <code>e=mail (nom_de_la_personne)</code> ou <code>e=nom_de_la_personne &lt;email&gt;</code> ou encore simplement <code>e=mail</code> .
<code>p+=33-061-770-5555</code>	Le numéro de téléphone doit être indiqué selon le format international. Les tirets peuvent être remplacés par des espaces.
<code>c=IN IP4 132.227.60.12/125</code>	<p>Le champ de connexion possède le format <code>&lt;type de réseau&gt; &lt;format de l'adresse réseau&gt; &lt;adresse réseau de connexion&gt;/&lt;ttl&gt;/&lt;nombre d'adresse de diffusion&gt;</code>. Alors que le type de réseau et le format de l'adresse réseau sont semblables à ce qui a été indiqué précédemment pour le champ <code>o</code>, l'adresse réseau de connexion spécifie l'adresse réseau (ou le nom à résoudre par DNS) de l'entité avec laquelle la connexion va s'effectuer.</p> <p>Avec ce champ <code>c</code>, il est possible d'indiquer autant de lignes que de destinataires du flux mentionné.</p> <p>La durée de vie des messages envoyés, ou TTL (Time to Live), en seconde est indiquée à la suite et prend une valeur comprise entre 0 et 255.</p>

Tableau 4.5 – Exemple de message SIP complet commenté (suite)

<p><i>c=IN IP4 132.227.60.12/125 (suite)</i></p>	<p>Si ce champ se trouve avant la description des médias (champ <i>m</i>), il s'applique aux médias sans distinction (tous les correspondants auront les mêmes types de flux). S'il suit une description de média, il est limité au média correspondant (par exemple, pour mettre en place une conférence avec une communication audio et vidéo avec un correspondant et une communication audio seulement avec un autre).</p> <p>Dans le cas où plusieurs terminaux sont concernés par le flux, il est nécessaire de répéter cette ligne pour chacun des destinataires. Une autre manière de faire, si l'adressage s'y prête, consiste à utiliser un adressage multicast en mentionnant la base réseau de multicast et le nombre de terminaux concernés.</p> <p>Ainsi, la ligne suivante :</p> <p><i>c=IN IP4 132.227.60.1/125/3</i></p> <p>est équivalente aux lignes :</p> <p><i>c=IN IP4 132.227.60.1/125</i></p> <p><i>c=IN IP4 132.227.60.2/123</i></p> <p><i>c=IN IP4 132.227.60.3/125</i></p> <p>Par défaut, le nombre d'adresses de diffusion est 1, et cette valeur peut être omise.</p>
<p><i>t=2873397496 2873404696</i></p>	<p>Une conférence peut être bornée dans le temps par l'attribution d'une valeur temporelle de début et de fin dans le champ <i>t</i>. Celui-ci donne respectivement un horaire auquel débute la conférence et un autre auquel elle s'achève (défini au format décimal de NTP, en seconde). La valeur 0 comme horaire de fin n'indique pas de fin pour l'appel. De même, la valeur 0 comme horaire de début indique une session permanente.</p>
<p><i>a=sendrecv</i></p>	<p>Il existe de nombreuses possibilités d'attributs. Leur format doit respecter l'une des deux syntaxes suivantes :</p> <ul style="list-style-type: none"> <li>– <i>a=&lt; nom_attribut &gt;</i> : permet de positionner un attribut binaire. Par exemple, <i>a=sendrecv</i> indique que les flux multimédias peuvent être initiés en réception comme en émission (SEND RECEIVE).</li> <li>– <i>a=&lt; nom_attribut &gt;:&lt;valeur&gt;</i> permet de donner une valeur à un attribut. Par exemple, si l'on partage un tableau blanc, pour indiquer qu'on souhaite l'orienter en mode portrait, on indique <i>a=orient:landscape</i>. De même, si l'on souhaite indiquer le nom du programme ayant permis de constituer le message SDP, on utilise <i>a=tool:monprogrammesdp 1.3</i>.</li> </ul>
<p><i>m=audio 49170 RTP/AVP 0</i>  <i>m=video 51327 RTP/AVP 31 32</i>  <i>m=application 32416 UDP wb</i>  <i>a=orient:portrait</i></p>	<p>Le format est mentionné selon la syntaxe suivante :</p> <p><i>&lt;media&gt; &lt;port&gt;/&lt;nombre de port&gt; &lt;transport&gt; &lt;format&gt;</i></p> <p>Le type de média est défini d'abord. Il peut avoir les valeurs prédéfinies suivantes : 'audio', 'video', 'application', 'data' et 'control'. Éventuellement, des valeurs supplémentaires peuvent être proposées.</p> <p>Le port concerné par la diffusion de ce type de média est indiqué à la suite. Dans le cas où les numéros de ports se suivent, il est possible de mentionner le nombre de ports concernés. Par exemple, en mentionnant comme port 1234/2, les ports 1234 et 1235 seront utilisés pour ce flux. Par défaut, c'est la valeur non obligatoire 1 qui définit le nombre de port.</p> <p>Le transport mentionne le nom du protocole de transport utilisé. On distingue les deux valeurs suivantes :</p> <ul style="list-style-type: none"> <li>– <i>RTP/AVP</i> : transport avec le protocole RTP en utilisant les paramètres audio-vidéo (<i>Audio/Video profile</i>). Implicitement, c'est le protocole UDP qui est utilisé pour les flux RTP.</li> <li>– <i>UDP</i> : seul le protocole UDP est utilisé.</li> </ul>

Tableau 4.5 – Exemple de message SIP complet commenté (*suite*)*(suite)*

```

m=audio 49170 RTP/AVP 0
m=video 51327 RTP/AVP 31 32
m=application 32416 UDP wb
a=orient:portrait

```

Il est possible d'étendre ces valeurs.

Le format est indiqué à la suite. Il correspond à un type définissant le codage des données. Par exemple, la valeur 0 correspond à un codage de type G.711 loi mu, utilisant un seul canal audio avec une fréquence de 8 kHz. De même, avec un seul canal et une fréquence de 8 kHz, la valeur 8 est utilisée pour un codage G.711 loi A, la valeur 2 pour un codage G.721, la valeur 3 pour un codage GSM et la valeur 9 pour un codage G.722. La RFC 1890 liste les différentes valeurs de codage possibles.

Plusieurs formats de codage peuvent être indiqués dans cette liste pour indiquer que tous ces formats sont susceptibles d'être utilisés au cours de la session, bien que, par défaut, ce soit le premier mentionné dans la liste qui est utilisé (c'est le cas pour le média vidéo, pour lequel deux formats sont donnés, 31 pour un codage H.261 et 32 pour un codage MPV). De plus, certaines valeurs peuvent être remplacées par des paramètres. Par exemple, le paramètre *wb* (pour whitepaper) définit le format du tableau blanc.

Dans l'exemple, trois médias différents sont définis (audio, vidéo et application). Le champ *a* étant mentionné après le champ *m* est un champ de média qui s'applique uniquement au dernier média mentionné (en l'occurrence le média application).

Le protocole SDP fournit un langage descriptif des sessions relativement simple, mais limité, puisqu'il n'offre pas de logique permettant d'exprimer des solutions de configuration différentes. En outre, il n'est pas extensible, ce qui interdit d'ajouter de nouveaux paramètres descriptifs.

D'une manière générale, le corps d'un message est rempli (ou non) selon le type du message. En ce qui concerne les requêtes, une description de la session est nécessaire dans le corps pour les messages d'initiation d'appel (requête INVITE), d'acquiescement (requête ACK) ou d'informations sur les caractéristiques d'un serveur (requête OPTIONS). Pour les messages d'enregistrement (requête REGISTER), d'annulation de requête (requête CANCEL) ou de terminaison d'appel (requête BYE), le corps peut être laissé vide.

Pour leur part, les réponses, hormis quelques rares exceptions (par exemple les réponses de type 2xx pour valider l'annulation d'une requête ou la terminaison d'une session), comportent le plus souvent dans le corps du message des informations complémentaires. Ces dernières peuvent consister en une description des paramètres que le terminal appelé doit supporter (afin d'alléger les transmissions, en particulier en multicast, il est par exemple possible de n'insérer que les paramètres qui varient par rapport à ceux proposés par l'appelant dans sa requête) ou des services accessibles sur le serveur sollicité. Le cas échéant, le corps décrit le type d'erreur rencontré lorsque le serveur a tenté d'exécuter la requête.

### Version du protocole

La version du protocole SIP utilisée est toujours indiquée dans la ligne de requête ou d'état du message. Autrement dit, elle est présente dans tous les messages SIP. Ainsi, les entités qui traitent les messages SIP peuvent rapidement savoir s'ils sont compatibles et

donc s'ils sont susceptibles de communiquer par SIP avec l'émetteur du message reçu. La compatibilité n'est effective que si la version est strictement identique.

Actuellement, c'est la version 2.0 qui est couramment utilisée. L'indication de la version doit se faire en spécifiant le mot-clé *SIP* suivi d'une espace puis du numéro de la version (*SIP 2.0*).

## Les requêtes SIP

Comme indiqué précédemment, une requête est composée de trois parties : une ligne de requête, les champs d'en-tête du message et le corps.

Les en-têtes et le corps ayant été détaillés précédemment, nous nous intéressons ici à la ligne de requête spécifique aux messages de requêtes.

La partie définissant la ligne de requête se compose des trois champs suivants illustrés à la figure 4.9 :

- *méthode*, qui indique l'action sollicitée.
- *URI*, qui précise le destinataire de la requête.
- *version*, qui spécifie le numéro de la version du protocole SIP utilisée.

Chacun de ces champs est séparé par un séparateur, ou SP (SeParator).

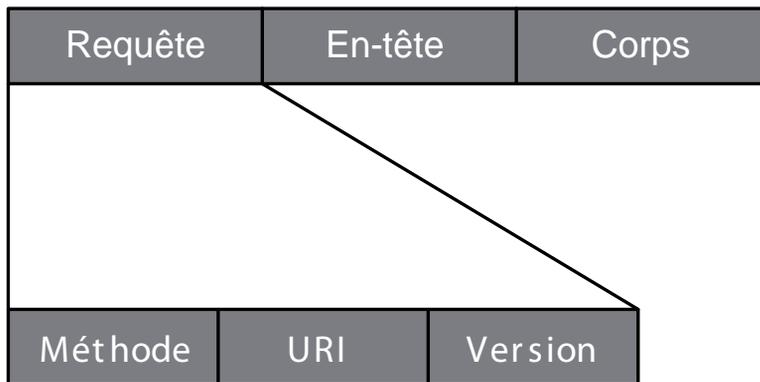


Figure 4.9

*Format d'une requête SIP*

### Les six méthodes d'une requête SIP

SIP n'utilise que six méthodes fondamentales pour formuler ses requêtes. Cela indique très nettement la volonté de simplicité de ses concepteurs.

Ces méthodes sont détaillées dans la RFC 3261. Elles doivent être supportées par tous les terminaux et serveurs sollicités. À ces méthodes fondamentales, ont été ajoutées des

méthodes supplémentaires, destinées à diverses améliorations, qui ne sont pas forcément implémentées par tous les terminaux.

Notons que l'établissement d'un appel peut ne faire intervenir que trois de ces méthodes fondamentales (INVITE, ACK et BYE).

#### Initier une session avec *INVITE*

La méthode INVITE permet d'initier une communication en invitant un correspondant à y participer. Elle peut aussi être utilisée pour une conférence afin d'inviter plusieurs interlocuteurs à communiquer au sein d'une même session.

Le corps du message de cette méthode fournit à l'appelé les paramètres de session souhaités et supportés par l'appelant. Ce dernier spécifiera, par exemple, le codec souhaité et le type de flux requis (voix, vidéo, etc.). En règle générale, l'appelant ne se contente pas d'une seule proposition, mais offre de multiples possibilités de paramètres. Ses préférences sont définies par l'ordre dans lequel apparaissent ses propositions.

Notons qu'une autre utilisation classique de cette méthode consiste à renégocier dynamiquement de nouveaux paramètres de session. Par exemple, si, durant une session déjà établie, l'un des interlocuteurs souhaite enrichir la voix par la vidéo, il fait sa demande par une requête d'invitation.

#### Confirmer les paramètres de session avec *ACK*

La méthode ACK correspond à un acquittement de l'appelant. Elle peut être utilisée dans les deux cas de figure suivants :

- Elle fait suite à l'acceptation d'un appel par l'appelé. Avec la méthode d'invitation, l'émetteur a fait connaître au récepteur les paramètres qu'il supporte et ses préférences. En réponse, le récepteur en a fait autant. Au final, l'émetteur compare les paramètres supportés par les deux terminaux et indique par la méthode ACK ceux qui seront utilisés. Dans le corps de ce message, la méthode décrit l'ensemble des paramètres de session à adopter au cours de la session. Si le corps de la méthode d'acquiescement est vide, les paramètres proposés par l'appelant pourront être adoptés pour la session.
- Elle fait suite à une réponse de localisation fournie par un serveur de redirection. Une fois la détermination de la position de l'appelé effectuée, le serveur de redirection retourne le résultat à l'UAC (User Agent Client). Celui-ci valide la réception de ce résultat par la méthode ACK.

#### S'informer sur le serveur avec *OPTIONS*

La méthode OPTIONS permet d'interroger un serveur SIP, y compris l'entité UAS (User Agent Server) sur différentes informations. Elle comporte globalement deux volets : l'état du serveur et ses capacités.

Par exemple, cette méthode offre la possibilité de savoir si un utilisateur que l'on souhaite appeler est présent, c'est-à-dire disponible pour initier une communication. Cette réponse est pratique en ce qu'elle donne des informations sur un serveur, sans avoir à initier une communication pour autant. Autrement dit, la requête ne fait pas sonner le

poste du récepteur, puisqu'elle n'établit pas d'appel, mais agit comme un indicateur de présence.

Avec cette méthode, il est possible d'obtenir des informations sur un serveur relativement aux types de médias supportés (audio, vidéo, données), aux codecs supportés, aux méthodes supportées et aux options d'appels, si le serveur en a diffusé. Le serveur qui reçoit cette requête répond par son état et ses capacités.

#### Terminer une session avec *BYE*

La méthode *BYE* permet de libérer une communication.

Cette requête peut être émise indifféremment par l'appelant ou par l'appelé. Elle n'attend pas d'acquiescement, puisqu'une terminaison d'appel peut être décidée unilatéralement.

#### Abandonner une demande avec *CANCEL*

Cette méthode annule une requête dont la réponse n'est pas encore parvenue au demandeur.

Elle ne permet pas d'interrompre une session, mais indique que la réponse n'est plus attendue et qu'il n'est donc pas nécessaire de traiter la requête.

Par exemple, une demande de recherche d'un utilisateur peut solliciter plusieurs serveurs de localisation en parallèle, qui vont tous rechercher la présence de l'abonné dans leur base de données et retourner le résultat de la recherche. Dès qu'un serveur a trouvé l'abonné, les autres serveurs n'ont plus besoin de poursuivre leur recherche. Un message d'annulation leur est donc envoyé.

Autre exemple, si un utilisateur envoie un message d'invitation *INVITE* et qu'il attende la réponse de l'appelé, il peut à tout moment émettre un message *CANCEL* afin d'annuler l'invitation sans avoir à attendre la réponse de l'appelant.

La méthode *CANCEL* est nécessairement acquiescée par un message *ACK* pour signifier que l'annulation est prise en compte.

#### Enregistrer sa localisation avec *REGISTER*

Cette méthode permet d'enregistrer son adresse IP auprès d'un serveur d'enregistrement. Elle permet donc d'assurer le service de localisation.

L'information enregistrée correspond à une entrée dans la base spécifiant la correspondance d'une adresse SIP avec une adresse IP.

Cette entrée a une durée de vie limitée. Passé ce délai, le serveur de localisation la supprime de sa base de données. Par défaut, la durée est d'une heure. Périodiquement, chaque terminal doit rafraîchir cette entrée pour la conserver en base ou, s'il est mobile, la modifier le cas échéant.

Une autre manière de gérer la durée de vie de l'enregistrement est d'utiliser le champ *expire* de la méthode afin d'imposer une durée de validité fixe, qu'il n'est pas nécessaire de mettre à jour. Le risque dans ce cas est que si l'abonné se déconnecte du réseau sans

l'indiquer au serveur d'enregistrement, il reste considéré comme actif dans le réseau (une tentative infructueuse de communication est initiée).

## **Méthodes d'extension du protocole SIP**

Plusieurs méthodes complémentaires ont été définies comme des extensions aux méthodes précédentes afin d'enrichir les communications SIP.

Leur implémentation n'est toutefois pas systématique, et certains terminaux peuvent ne pas en supporter certaines. Néanmoins, ces méthodes apportent des fonctionnalités pratiques.

Les huit méthodes d'extension classiquement utilisées sont les suivantes :

- **INFO.** Décrite dans la RFC 2976, cette méthode donne des informations complémentaires sur la session en cours (puissance de réception, qualité du son, codecs utilisés, etc.). Seules des données informatives sont envoyées en réponse à cette commande, ce qui ne modifie aucunement la session en cours.
- **REFER.** Décrite dans la RFC 3515, cette méthode permet d'indiquer au récepteur du message une ressource. Elle autorise de la sorte différents services de relais, en particulier la redirection d'appel.
- **UPDATE.** Décrite dans la RFC 3311, cette méthode met à jour les paramètres d'une session multimédia.
- **PRACK.** Décrite dans la RFC 3262, cette méthode fournit une sécurisation des réponses provisoires. PRACK est l'acronyme de Provisional Reliable ACK.
- **MESSAGE.** Décrite dans la RFC 3428, cette méthode est utilisée pour l'envoi de messages instantanés. Le contenu de cette méthode peut être fait en MIME, ce qui offre une liberté pour le transport des différents types de contenus.
- **SUBSCRIBE.** Décrite dans la RFC 3265, cette méthode permet de demander une notification d'événement. Un client envoyant au serveur cette requête réclame d'être informé lorsque certains événements surviennent. Par exemple, un utilisateur peut souhaiter être averti lorsqu'un de ses contacts termine sa communication en cours. Il interroge alors le serveur pour lui demander d'être prévenu lorsque le contact devient disponible. Si le serveur accepte cette demande, il notifie les événements correspondant à l'abonnement de l'utilisateur pendant toute la durée spécifiée dans le champ EXPIRES inséré dans l'en-tête du message. Au-delà de cette durée, la notification n'est plus active. La notification est effectuée par le serveur au moyen de la commande NOTIFY.
- **NOTIFY.** Décrite dans la RFC 3265, cette méthode permet de recevoir une notification pour les événements auxquels on a souscrit. Lorsqu'un client a envoyé une demande de notification d'événement par la méthode SUBSCRIBE, le serveur concerné lui envoie les notifications réclamées par la méthode NOTIFY.
- **PUBLISH.** Décrite dans la RFC 3903, cette méthode permet d'afficher l'état d'une requête à laquelle on a souscrit par la méthode SUBSCRIBE.

Un message de requête complet serait de la forme suivante :

```

INVITE          sip :laurent@reseau_lab0.fr          SIP/2.0

From :          Guy <sip : guy@reseau_lab0.fr>
To :           Laurent <sip : laurent@reseau_lab0.fr>
Call-ID :      3210-zad-323-101@ reseau_lab0.fr
CSEQ :         123 INVITE
Subject :      Réunion d'équipe demain.
VIA :          SIP/2.0/UDP 192.168.1.3@reseau_lab0.fr
Expire :       7200
Content-Type:  application/SDP
Content Lenght : 13232

v= 0
o= IN IP4 132.227.60.2
s= Test de ToIP
c= IN IP4 prox11AB.1ip6.fr
m= audio 4142 RTP
  
```

## Les réponses SIP

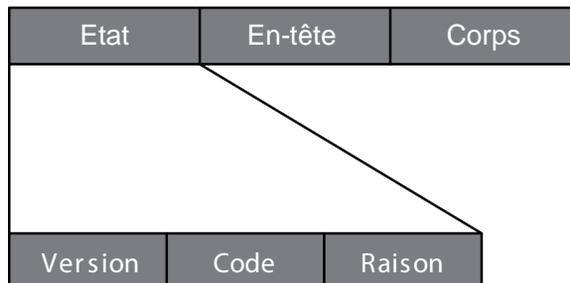
Quelle que soit la méthode utilisée dans une requête, le récepteur final doit y apporter au moins une réponse en retour, ne serait-ce qu'une réponse temporaire pour informer l'émetteur que sa requête est prise en compte et en train d'être traitée et qu'elle sera suivie d'une réponse finale dès que possible.

Les réponses sont classées en catégories, suivant leur type. Elles peuvent au besoin être complétées par un message plus explicite.

Les réponses SIP doivent respecter le format illustré à la figure 4.10.

**Figure 4.10**

Format des réponses SIP



Les réponses aux requêtes SIP débutent par une ligne d'état (Status Line), laquelle comporte les trois champs suivants :

- *Version* : c'est la version du protocole SIP utilisée.
- *Code d'état (Status Code)* : code numérique à trois chiffres spécifiant la réponse donnée à la requête. Cet entier est codé sur trois bits.
- *Raison (Reason Phrase)* : message textuel expliquant brièvement le code d'état de la réponse. Fonctionnellement parlant, il s'agit d'un élément redondant par rapport au code d'état. Le protocole offre néanmoins ainsi un niveau de clarté plus accessible, qui favorise la compréhension des messages protocolaires par les utilisateurs comme par les programmeurs. Son utilité n'est pas protocolaire mais informative.

Il existe six classes de réponse dans lesquelles sont répertoriés tous les messages de retour possibles. Le premier chiffre de chaque code spécifie la catégorie à laquelle appartient le code.

Le tableau 4.6 récapitule les principaux codes d'état et leur raison respective, utilisés dans les messages de réponse SIP par les entités SIP (incluant les UAS comme n'importe quel serveur SIP dédié).

**Tableau 4.6 – Principaux codes d'état et raisons d'une réponse SIP**

Code d'état	Raison	Commentaire
<i>1xx – Messages d'information</i>		
La requête est en cours de traitement. C'est une réponse temporaire, qui est purement informative. Une réponse définitive (ou finale, correspondant à toute autre catégorie que 1xx) devra être émise au plus tard dans 200 ms.		
100	TRYING	Tentative d'appel en cours
180	RINGING	Le poste de l'appelé est en train de sonner.
181	CALL IS BEING FORWARDED	L'appel est en train d'être redirigé vers la position actuelle de l'appelé.
182	QUEUED	L'appelant n'est pas disponible. L'appel n'est pas rejeté pour autant mais simplement mis en attente.
<i>2xx – Messages de succès</i>		
La requête a été reçue, comprise et acceptée par le serveur.		
200	ok	La requête a été exécutée avec succès.
<i>3xx - Messages de redirection</i>		
Décrit une autre action à effectuer avant de finaliser la requête. Cela peut être une nouvelle position de l'utilisateur qui est retournée ou bien une information sur un autre service imposé ou proposé pour satisfaire l'appel.		
300	multiple choices	La résolution d'adresse de l'appelé a conduit à plusieurs localisations possibles, toutes retournées à l'appelant afin qu'il en choisisse une.
301	moved permanently	L'appelé n'est plus disponible à la localisation demandée. Une nouvelle localisation est envoyée pour inviter l'appelant à le contacter à cette adresse.
302	moved temporarily	L'appelé est temporairement indisponible à l'adresse spécifiée. Une nouvelle localisation est spécifiée à l'appelant.
305	use proxy server	L'appelant doit impérativement utiliser un proxy pour pouvoir contacter l'appelé.
380	alternative service	L'appel n'a pas abouti, mais d'autres services sont disponibles pour l'appelant s'il le souhaite. Ces services sont listés dans le corps du message.

Tableau 4.6 – Principaux codes d'état et raisons d'une réponse SIP (*suite*)

Code d'état	Raison	Commentaire
<i>4xx – Messages d'erreurs : erreurs du client ou requêtes ne pouvant être prises en charge</i>		
La syntaxe est erronée ou la requête ne peut être prise en charge. Dans ce cas, l'appelant ne doit renvoyer la même requête que s'il a fait une modification ou passé un certain délai ou encore sollicité un autre serveur.		
400	BAD REQUEST	Le format de la requête est incorrect et ne peut être compris. Le message devrait inclure des informations sur la partie de la requête qui pose problème.
401	UNAUTHORIZED	Une authentification est requise pour pouvoir effectuer l'exécution de cette requête. Cette réponse est émise soit par un serveur d'enregistrement, soit par un UAS.
402	PAYMENT REQUIRED	Un paiement est nécessaire pour effectuer l'exécution de cette requête (destiné à un usage futur).
403	FORBIDDEN	Le service demandé est reçu et compris, mais son exécution est interdite.
404	NOT FOUND	L'appelé n'a pas été trouvé à l'URI spécifié.
405	METHOD NOT ALLOWED	La méthode sollicitée n'est pas supportée. Une liste des méthodes qui le sont est fournie dans le corps du message de réponse à la requête.
406	NOT ACCEPTABLE	Le champ d'en-tête ACCEPT de la requête impose un format qui ne peut être respecté dans la réponse.
407	PROXY SERVER AUTHENTICATION REQUIRED	Une authentification est requise pour effectuer l'exécution de cette requête. Cette réponse est émise par les serveurs proxy exclusivement (contrairement au code 401).
408	REQUEST TIMEOUT	La requête nécessite un temps de traitement trop long (par exemple, la localisation d'un abonné).
410	GONE	L'appelé n'est définitivement plus disponible à l'URI spécifié. Si le caractère permanent est incertain, une erreur de code 404 est utilisée.
413	REQUEST ENTITY TOO LARGE	Le serveur ne peut prendre en charge une requête dont le corps est aussi gros.
414	REQUEST-URI TOO LONG	L'URI est trop long pour être interprété par le serveur.
415	UNSUPPORTED MEDIA TYPE	Le format du corps du message de requête n'est pas pris en charge par le serveur pour cette requête. Le serveur doit retourner la liste des formats qu'il supporte dans les champs ACCEPT, ACCEPT-ENCODING ou ACCEPT-LANGUAGE.
416	UNSUPPORTED URI SCHEME	Le serveur ne supporte pas la forme de l'URI indiqué dans la requête.
420	BAD EXTENSION	Dans l'en-tête REQUIRE ou PROXY-REQUIRE une extension d'un protocole inconnu est spécifiée. Le serveur doit indiquer dans sa réponse les extensions qu'il ne supporte pas.
421	EXTENSION REQUIRED	La spécification d'une extension est indispensable pour l'exécution de la requête.
423	INTERVAL TOO BRIEF	La durée d'expiration spécifiée dans le message est trop courte pour être prise en compte. Typiquement, un enregistrement dans le serveur de localisation qui a une durée de validité trop courte peut être refusé par le serveur.
480	TEMPORARILY NOT AVAILABLE	L'appelé a été contacté, mais il est temporairement indisponible et ne peut prendre la communication.
481	CALL LEG/TRANSACTION DOES NOT EXIST	La requête fait référence à une autre requête dont l'identifiant est inconnu. Typiquement, une requête de type BYE ou CANCEL dont le Call-Id n'existe pas est invalide.

Tableau 4.6 – Principaux codes d'état et raisons d'une réponse SIP (*suite*)

Code d'état	Raison	Commentaire
482	LOOP DETECTED	Une boucle a été détectée dans le routage de la requête. Concrètement, le serveur qui réceptionne une requête retrouve sa propre adresse dans le chemin parcouru (champ VIA), ce qui indique qu'il l'a déjà traitée.
483	TOO MANY HOPS	Trop de nœuds intermédiaires sont requis pour le traitement de la requête. C'est le champ MAX FORWARDS qui spécifie le nombre de nœuds de transit maximal des requêtes.
484	ADDRESS INCOMPLETE	L'adresse de localisation spécifiée pour la personne appelée est incomplète.
485	AMBIGUOUS	La localisation de l'appelé présente une ambiguïté qui ne peut être résolue. Les différentes interprétations de cette localisation sont fournies quand c'est possible.
486	BUSY HERE	L'appelé a été contacté, mais il est occupé et ne peut prendre la communication.
487	REQUEST TERMINATED	La requête a été terminée par la réception d'un message BYE ou CANCEL.
488	NOT ACCEPTABLE HERE	L'appelé a été joint, mais certains paramètres de session ne peuvent être mis en œuvre et doivent être changés. Identique au message 606, si ce n'est qu'il ne s'applique qu'à la ressource spécifiée dans l'URI.
<i>5xx - Messages d'erreurs : erreur du serveur</i>		
La requête est correcte, mais le serveur n'a pas réussi à la traiter, car le service n'est plus disponible sur le serveur sollicité.		
500	INTERNAL SERVER ERROR	Une erreur de nature non prévisible est survenue et empêche la réalisation de l'exécution.
501	NOT IMPLEMENTED	La requête n'est pas implémentée sur le serveur qui la reçoit.
502	BAD GATEWAY	Le serveur, agissant en tant que passerelle ou proxy, a reçu une réponse invalide en essayant de relayer la requête pour la traiter.
503	SERVICE UNAVAILABLE	Le service n'est pas disponible en ce moment, peut-être par surcharge du serveur, maintenance ou dysfonctionnement.
504	GATEWAY TIMEOUT	La requête a fait appel à une demande dont la réponse n'est pas parvenue dans le délai attendu. Typiquement, un serveur proxy qui interroge un serveur de localisation peut retourner ce code s'il n'a pas reçu de réponse passé un temps déterminé.
505	SIP VERSION NOT SUPPORTED	La requête spécifie une version du protocole SIP qui n'est pas prise en charge par le serveur chargé de son exécution.
<i>6xx - Messages d'erreurs : échec général</i>		
Aucun serveur ne peut traiter cette requête, car ils sont occupés, inaccessibles ou refusent l'appel. La syntaxe de la requête n'est pas en cause.		
600	BUSY EVERYWHERE	L'appelé a été joint, mais il est occupé sur tous les postes et ne peut prendre la communication.
603	DECLINE	L'appelé a été joint, mais il a refusé la communication.
604	DOES NOT EXIST ANYWHERE	Le serveur qui a autorité sur la gestion du compte de l'appelé informe l'appelant que l'appelé n'existe pas dans le réseau.
606	not acceptable	L'appelé a été joint, mais certains paramètres de session ne peuvent être mis en œuvre, quelle que soit la localisation, et doivent être changés.

On remarque que ces codes d'erreur sont semblables à ceux utilisés par le protocole HTTP version 1.1 (RFC 2616), soulignant un effort de généralisation délibéré. Néanmoins, le protocole SIP ne s'approprie pas tous les codes de HTTP 1.1, certains étant inappropriés. Inversement, il étend les réponses possibles avec HTTP 1.1, avec, d'une part, des messages spécifiques (tous les codes supérieurs ou égaux à  $x80$ , avec  $x$  prenant une valeur comprise entre 1 et 5, sont propres à SIP) et, d'autre part, une nouvelle classe de message (la classe 6 est spécifique à SIP). Tout code d'état non référencé et reçu par un terminal est interprété comme étant équivalent au code  $x00$ .

Un message de réponse complet serait de la forme suivante :

```
SIP/2.0                                180                                Ringing

VIA :                                  SIP/2.0/UDP 192.168.1.18@reseau_labo.fr
From :                                  Guy <sip : guy@reseau_labo.fr>
To :                                    Laurent <sip : laurent@reseau_labo.fr>
Call-ID :                               3210-zad-323-1o1@ reseau_labo.fr
CSEQ :                                  123 INVITE
Expires :                               7200
```

## Scénarios de communication

Nous allons illustrer la succession chronologique des messages de requêtes et de réponses dans les six scénarios classiques suivants :

1. Initialisation d'une communication directe.
2. Enregistrement d'un terminal.
3. Initialisation d'une communication avec un serveur proxy.
4. Localisation par un serveur de redirection et initialisation d'appel directe.
5. Modification dynamique d'une communication SIP.
6. Terminaison d'une communication.

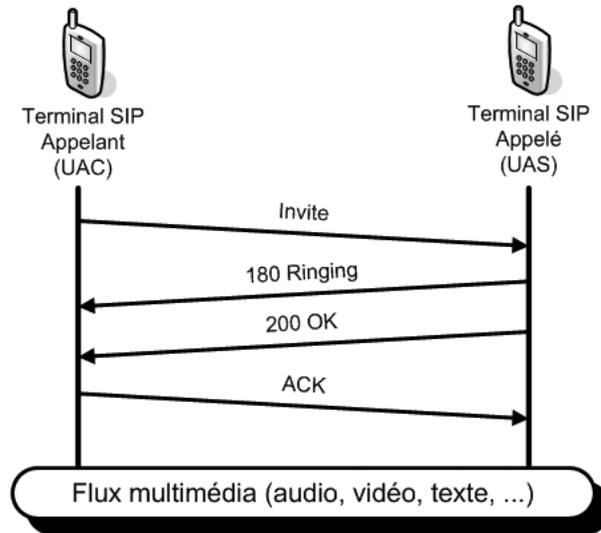
### 1. Initialisation d'une communication directe

Une communication peut s'effectuer directement entre deux correspondants, sans faire intervenir d'autre entité.

Dans ce cas, l'appelant doit connaître la localisation (sous forme d'adresse IP) de la personne qu'il souhaite contacter.

La figure 4.11 illustre ce scénario.

**Figure 4.11**  
*Initiation d'une communication directe*



Cette communication reflète la simplicité d'utilisation du protocole SIP.

Quatre étapes seulement suffisent à mettre en relation les deux utilisateurs :

1. L'appelant (UAC) envoie un message (requête INVITE) proposant à son correspondant (UAS) d'initier une communication. Ce message contient les paramètres désirés pour établir la communication.
2. Dès que l'UAS reçoit le message, il en informe l'utilisateur appelant (le téléphone sonne, avec indication de l'appelant et du motif de son appel s'il a renseigné ce champ, ainsi que des services disponibles). Dans le même temps, il indique à l'appelant (par une réponse provisoire 180 RINGING) que l'appelé est en train d'être averti de l'appel.
3. Dès que l'appelé accepte l'appel (en décrochant), l'UAS informe l'appelant (par une réponse définitive 200 OK) que l'appel peut débuter. Ce message contient les paramètres que l'UAS supporte pour la session.
4. L'UAC retourne à l'UAS un message d'acquittement (requête ACK) lui indiquant qu'il a pris note que l'appel peut débuter. Ce message comporte les paramètres fixés pour la session, qui tiennent compte de ces possibilités et de celles de l'UAS.

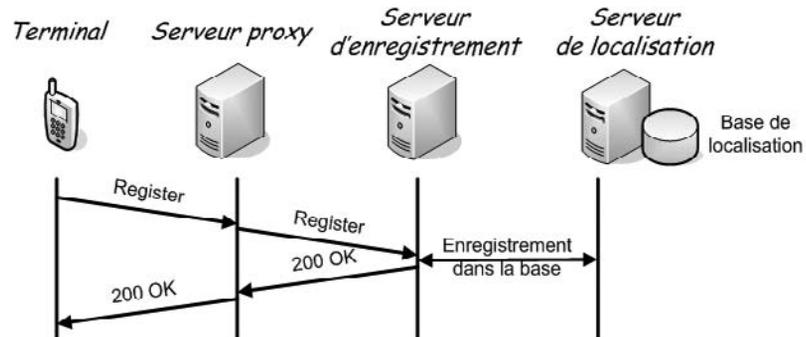
Les intervenants sont ensuite mis en relation et peuvent communiquer.

## 2. Enregistrement d'un terminal

Lorsqu'un terminal est activé dans un réseau, sa première action consiste à se déclarer auprès d'un serveur d'enregistrement, de manière à être disponible si un appelant souhaite le joindre.

Ce scénario est illustré à la figure 4.12.

**Figure 4.12**  
Enregistrement  
d'un terminal SIP



Le serveur de localisation maintient dans sa base de données une entrée associant l'identifiant d'un utilisateur avec sa position dans le réseau (adresse IP du terminal de l'utilisateur, port utilisé par l'application SIP et identifiant de l'utilisateur sur ce poste).

Cette entrée sera du type indiqué au tableau 4.7.

**Tableau 4.7** Entrée dans le serveur de localisation permettant de localiser un utilisateur

Utilisateur	Localisation	Délai d'expiration
<code>sip:albert@mon_domaine.fr</code>	<code>sip:albert@132.227.155.155:12345</code>	48 minutes

Notons la présence d'un délai d'expiration, ici arbitrairement fixé à 48 minutes (par défaut, une entrée est valable pendant une heure). Périodiquement, le terminal doit rafraîchir son entrée à l'aide de la requête REGISTER afin de manifester sa présence. À défaut, l'entrée est effacée.

## 3. Initialisation d'une communication SIP avec un serveur proxy

Les étapes et messages envoyés pour initier une session entre deux correspondants dans le cas où un proxy est utilisé sont illustrés à la figure 4.13.

Dans cet exemple, Anne souhaite ouvrir une session avec Brigitte. Comme elle ne connaît pas la localisation de cette dernière, elle sollicite son proxy afin de la déterminer.

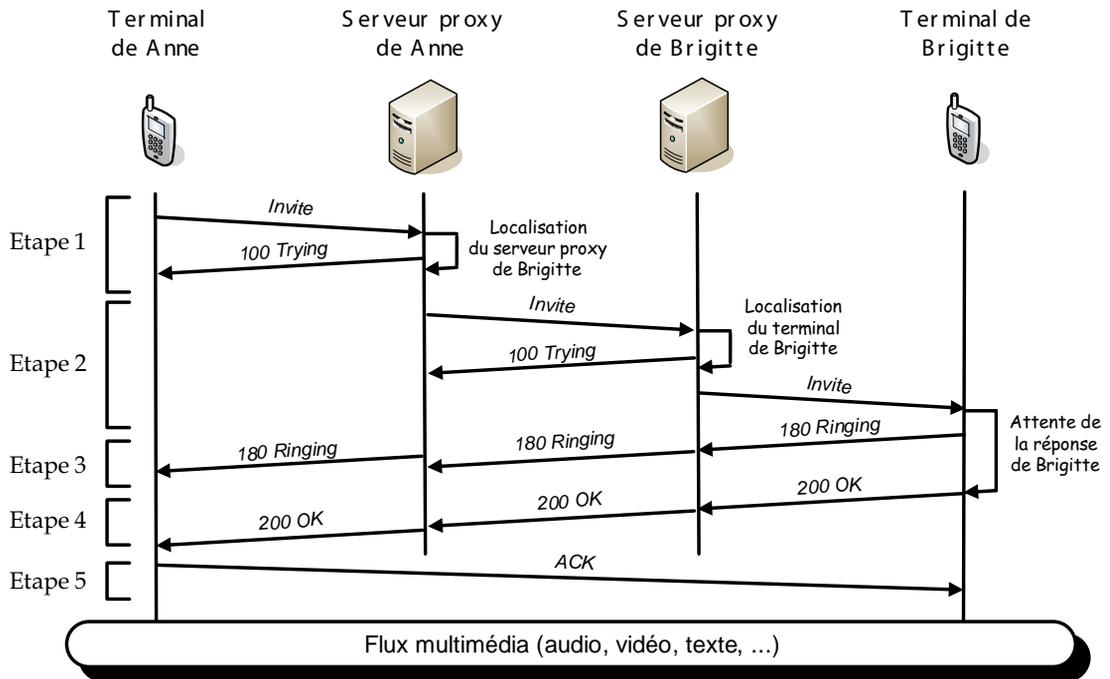


Figure 4.13

Initialisation d'un appel avec un proxy

Les étapes suivantes sont nécessaires :

1. Anne compose sur son terminal l'adresse SIP de Brigitte. Cette dernière n'est pas nécessairement une adresse IP et peut être un identifiant qu'il faut résoudre. Un message d'invitation (requête INVITE) est envoyé de l'UAC d'Anne vers son serveur proxy SIP. L'adresse du proxy d'Anne peut être configurée sur son terminal ou être automatiquement distribuée, par DHCP par exemple. À réception de ce message, le serveur proxy d'Anne utilise la partie domaine de l'adresse SIP de Brigitte pour déterminer le serveur en charge de la gestion du compte de Brigitte (c'est-à-dire en charge du domaine de Brigitte). À cette fin, un serveur DNS peut être sollicité pour localiser le serveur proxy de Brigitte. En parallèle, le serveur proxy informe Anne qu'il prend en charge la requête et tente de la mettre en relation. La réponse temporaire *100 TRYING* indique à cette dernière que le message a été reçu et qu'il est en cours de traitement.
2. Routage du message d'invitation. Le serveur proxy d'Anne transmet l'invitation au serveur proxy de Brigitte après l'avoir localisé. C'est le message d'invitation original qui est intégralement relayé du proxy d'Anne vers celui de Brigitte. La seule modification apportée au message par le premier serveur proxy concerne le champ *VIA*, qui liste l'ensemble des machines parcourues lors de l'acheminement du paquet, et

auquel il ajoute sa propre adresse réseau (en plus de celle d'Anne, qui y figure initialement). Le serveur proxy de Brigitte informe le serveur proxy d'Anne (par un message de réponse temporaire *100 TRYING*) de la réception de la requête et de la tentative d'initialisation. Parallèlement, il recherche la localisation du terminal de Brigitte en utilisant le service de localisation. Une fois la position du terminal dans le réseau trouvée, il lui transmet l'invitation d'Anne. À nouveau, ce message est conforme à l'original, et seul le champ *VIA* a été enrichi de l'adresse du serveur proxy de Brigitte.

3. Le terminal de Brigitte sonne. Le téléphone de Brigitte (éventuellement un soft-phone) reçoit l'invitation et la fait connaître à l'utilisateur Brigitte, le plus souvent par une sonnerie. En parallèle, il indique à son proxy (par un message *180 RINGING*) que l'appel est en train d'être notifié à Brigitte et que la communication est en attente de son acceptation. Ce message informatif est relayé jusqu'à l'émettrice Anne, qui reçoit généralement un retour audio ou visuel (une tonalité de sonnerie particulière le plus souvent). L'utilisation du champ d'en-tête *VIA* permet de remonter de proche en proche jusqu'à Anne selon le même chemin.
4. Brigitte répond au téléphone. On suppose le cas où Brigitte a choisi de répondre à l'appel. À l'instant où elle décroche, l'UAS retourne à l'UAC un message *200 OK* pour l'informer que l'appel est accepté. Ce message est relayé par les différents proxy. À ce stade, la communication n'a pas encore débuté, et aucun son n'est transmis.
5. Le terminal d'Anne confirme les paramètres d'appel. En tenant compte des capacités prises en charge par les correspondants, le terminal d'Anne envoie un message d'acquiescement *ACK* qui spécifie les paramètres définitifs à utiliser lors de cette session. Notons que le message d'acquiescement peut passer directement d'un interlocuteur à l'autre, sans transiter par les serveurs proxy. À ce stade, chacun des utilisateurs a pu apprendre la localisation exacte de son interlocuteur, et il n'est donc plus nécessaire de recourir aux serveurs proxy. Toutes les transactions qui suivent sont effectuées directement, de poste utilisateur à poste utilisateur. Ainsi, les serveurs proxy sont sollicités au minimum. De la même manière, pour ne pas saturer les serveurs proxy inutilement, les flux de données multimédias ne transitent jamais par eux.

À réception de ce message, la communication entre les interlocuteurs peut débuter. Tous ces échanges n'ont réclamé que quelques millisecondes, imperceptibles pour les intervenants.

Globalement, on retrouve dans cet appel, les trois phases fondamentales de l'appel direct entre les correspondants :

1. Requête *INVITE* : invitation de l'appelant.
2. Réponse *200 OK* : acceptation par l'appelé.
3. Acquiescement *ACK* : confirmation par l'appelant.

Il s'agit des trois messages nécessaires à la modification dynamique d'une communication SIP. Les autres messages concernent essentiellement la localisation ou sont à titre informatif.

#### 4. Localisation par un serveur de redirection et initialisation d'appel directe

La figure 4.14 illustre le scénario où un serveur de redirection est utilisé par le terminal appelant afin de localiser son correspondant et pour l'échange qui s'ensuit. L'objectif est toujours de mettre en relation le terminal d'Anne avec celui de Brigitte, mais par un autre moyen.

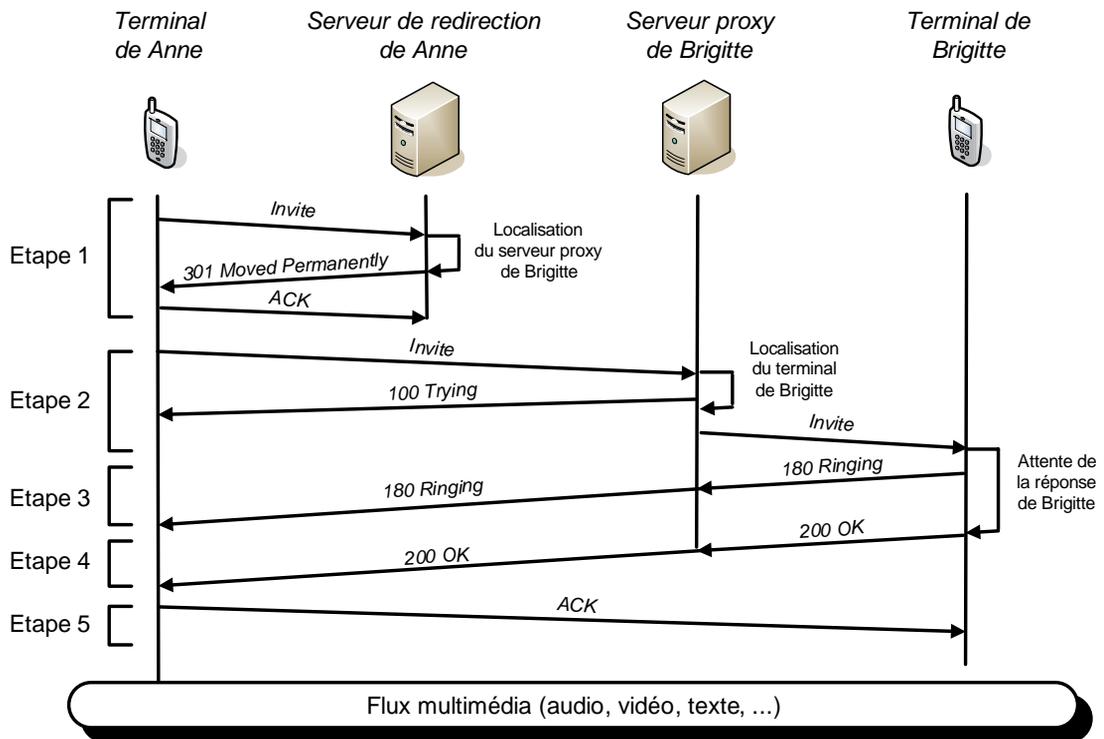


Figure 4.14

Localisation avec un serveur de redirection et initialisation d'appel

Dans la première étape, le terminal d'Anne sollicite le serveur de redirection pour déterminer la localisation du terminal d'Anne. Une fois cette recherche effectuée, la réponse est envoyée directement au terminal d'Anne, lequel initie l'appel lui-même, en contactant le serveur proxy de Brigitte.

Les étapes qui suivent sont identiques à celles du scénario précédent avec l'initialisation d'appel par un serveur proxy, si ce n'est que ce dernier n'intervient pas dans les échanges intermédiaires.

## 5. Modification d'une communication SIP

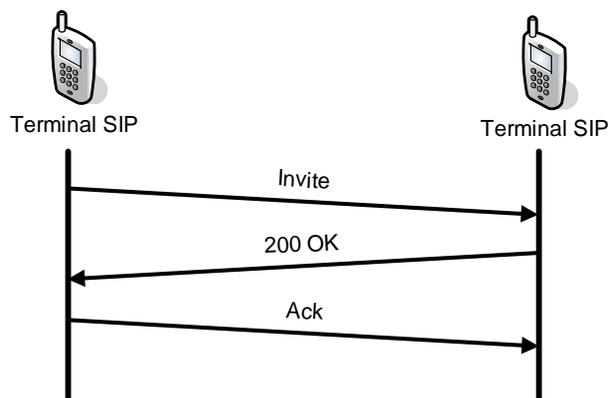
Lorsqu'un utilisateur est en communication, il peut arriver qu'il souhaite modifier les paramètres de cette communication tout en la conservant active. Par exemple, s'il commence un téléchargement et que son débit risque de diminuer en conséquence, il peut souhaiter utiliser un codec moins gourmand. Dans un autre cas, l'utilisateur peut vouloir enrichir la communication audio avec une diffusion vidéo. Ou encore, il peut souhaiter inviter à une conférence un nouveau correspondant, qui ne supporte pas le codec utilisé par les autres conférenciers.

Ces cas sont parfaitement envisageables avec le protocole SIP, qui offre, rappelons-le, une très grande souplesse. À tout moment, l'appelant ou l'appelé peut envoyer un nouveau message d'invitation, avec la requête INVITE, afin de renégocier les paramètres de la communication. Bien sûr, dans ce contexte, le message n'a pas pour objectif d'inviter à une nouvelle session, mais d'utiliser de nouveaux paramètres.

C'est pour cette raison qu'on nomme RE-INVITE ce type de requête d'invitation. Du reste, la communication en cours n'est pas interrompue par la réception de cette requête. S'il accepte les modifications sollicitées dans la requête d'invitation, le récepteur confirme son accord par l'envoi d'une réponse 200 OK, qui sera ensuite acquittée par le demandeur, comme pour l'initiation d'une communication (voir figure 4.15). Dans ce contexte, cette requête ne fait pas sonner le poste de l'interlocuteur puisque la communication est déjà en cours.

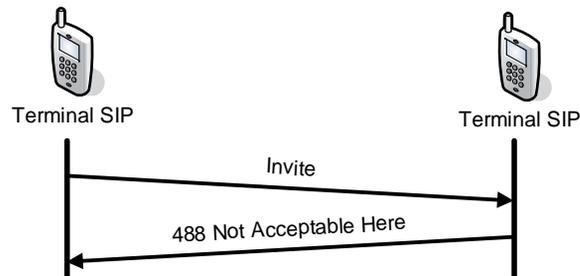
Figure 4.15

Requête re-invite acceptée



Dans le cas contraire, où le récepteur ne supporte pas ou ne souhaite pas accepter la modification de la session en cours, il reste libre de le faire, sans pour autant mettre fin à la communication, en envoyant un message de réponse 488 NOT ACCEPTABLE HERE, comme l'illustre la figure 4.16.

Figure 4.16

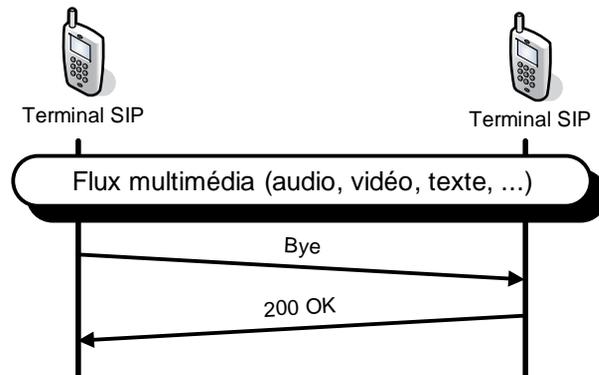
*Requête re-invite refusée*

Le demandeur qui en prend connaissance ne peut effectuer la modification désirée et doit soit se contenter des paramètres de la session actuelle, soit faire une nouvelle offre, en suggérant l'utilisation d'autres paramètres.

## 6. Terminaison d'une communication SIP

La figure 4.17 illustre la terminaison d'une session à l'initiative de n'importe quelle entité souhaitant mettre fin à l'appel.

Figure 4.17

*Terminaison d'une communication*

Cette opération ne comporte que les deux étapes très simples suivantes :

1. Un message (requête BYE) est envoyé pour indiquer au correspondant que la session va être clôturée.
2. Le correspondant répond à cette requête en validant la prise en compte de cette demande par une réponse 200 OK.

La communication entre les intervenants est alors rompue.

## Conclusion

Indiscutablement, le protocole H.323 possède une avance historique par rapport au protocole SIP. Son interaction avec les réseaux téléphoniques RTC est parfaitement maîtrisée, alors qu'elle n'est pas totalement spécifiée avec le protocole SIP. Globalement, H.323 est plus riche en termes de fonctionnalités que SIP.

Les deux protocoles de signalisation disposent de mécanismes d'extensibilité : avec H.323 les paramètres NONSTANDARDPARAM (défini en ASN.1) peuvent contenir des codes personnalisés pour y développer des extensions. De même, SIP hérite du mécanisme PEP (Protocol Extension Protocol) de HTTP permettant de définir des pointeurs documentant des caractéristiques complémentaires. La compatibilité avec H.323 est complète, quoique souvent laborieuse, puisqu'elle impose de respecter des fonctionnalités lourdes qui ont été améliorées mais doivent toujours être supportées. Au contraire, la compatibilité avec SIP n'est pas explicitement requise, ce qui allège les implémentations du protocole tout en restreignant son cadre d'application.

Mais SIP possède des arguments solides qui plaident en sa faveur : plus souple, rapide et modulaire que ne l'est H.323, le protocole SIP bénéficie d'un héritage protocolaire issu du monde Internet. Il s'intègre simplement dans un réseau IP et profite d'une architecture distribuée pour s'adapter facilement à la montée en charge d'utilisateurs au sein d'un réseau.

SIP étend même ses possibilités par de nouveaux protocoles qui s'appuient sur les fonctionnalités de SIP et ses capacités pour de nouvelles applications. C'est le cas du protocole SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) qui utilise SIP pour gérer la messagerie instantanée et la fonctionnalité de présence. SIMPLE est étudié au sein du groupe de travail IMPP (Instant Messaging and Presence Protocol) de l'IETF, et est soutenu par d'importants industriels, tels que Microsoft ou Lotus.

En outre, la mobilité est une notion centrale dans SIP : un même utilisateur peut être joint par différents moyens, par exemple à son domicile ou à son lieu de travail, ou encore sur son téléphone fixe, son téléphone portable ou son assistant personnel. En conservant une même identité, le protocole SIP détermine la localisation réelle de l'utilisateur : c'est la notion de *personal mobility*.

Pour toutes ces raisons, SIP a aujourd'hui les faveurs des industriels et s'impose progressivement auprès des acteurs de la ToIP, tandis que H.323 se marginalise dans les nouveaux produits et installations. Ainsi, des fournisseurs d'accès ADSL, tels que Free et Neuf Telecom, l'ont choisi pour assurer la signalisation de leur service de téléphonie IP. De même, Microsoft utilise SIP dans son serveur unifié de communications multimédias LCS (Live Communications Server).

Le protocole SIP constitue donc un solide concurrent de H.323 : il repose sur des bases saines et solides, que devront conforter d'importants développements pour compléter les fonctionnalités de SIP attendues dans le futur.

# 5

## Le protocole MGCP

---

L'une des raisons qui expliquent l'émergence et le succès du protocole H.323 est le besoin de regrouper les différents acteurs du multimédia, qu'ils soient équipementiers ou fournisseur de services, autour de normes communes. La concurrence engendrée par le protocole SIP a réduit cet effet d'universalité puisque les réseaux IP ont désormais une solide solution de rechange à H.323.

Dès lors, se pose la question de la communication entre des réseaux de nature différente (ATM, RNIS, RTC ou IP) ou bien de même nature mais exploitant des protocoles de signalisation différents (H.323, SIP ou autre).

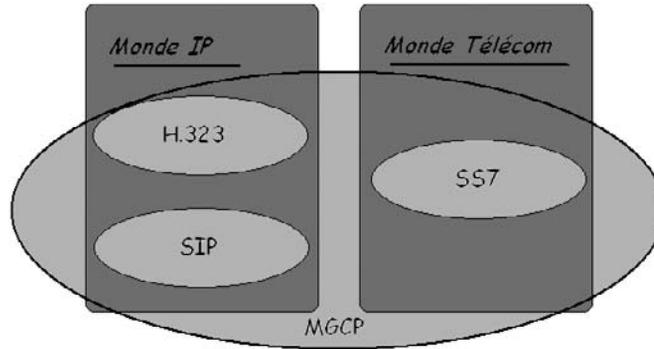
Si les passerelles ont déjà été introduites avec le protocole H.323, elles demeurent des entités complexes, onéreuses et non administrables, qui plus est fortement sollicitées alors qu'elles tiennent difficilement la montée en charge. Pour combler ce nouveau besoin, le protocole MGCP (Media Gateway Control Protocol), ou protocole de contrôle des passerelles multimédias, a été proposé.

Il fonctionne au niveau applicatif et permet d'offrir une couverture plus large en fédérant toutes les signalisations, qu'elles soient de type IP ou RTC entre autres. C'est le maître d'œuvre de l'interopérabilité entre tous les protocoles de signalisation et tous les réseaux, de quelque nature qu'ils soient.

Comme l'illustre la figure 5.1, qu'il s'agisse de la signalisation SS7, utilisée dans un réseau commuté, H.323 ou SIP, le protocole MGCP est conçu pour relier et faire communiquer l'ensemble de ces réseaux.

Figure 5.1

Rôle fédérateur  
du protocole MGCP



MGCP est aujourd'hui massivement utilisé par les fournisseurs d'accès Internet pour assurer le contrôle et l'administration à distance des boîtiers (*box*) mis à disposition de leurs abonnés.

Après un bref rappel historique, ce chapitre se penche sur les caractéristiques essentielles de MGCP, qui en font un protocole de choix pour les opérateurs.

Nous y détaillons successivement ses principales fonctionnalités, son architecture, ainsi que les échanges protocolaires utilisés dans ses communications.

## Historique

MGCP puise sa source dans une problématique de convergence des réseaux, que différents acteurs ont voulu résoudre indépendamment les uns des autres. Une fois encore, ces tentatives individuelles des constructeurs et équipementiers n'ont fait qu'accroître les incompatibilités entre entités réseau.

Fruit de réflexions communes, le protocole MGCP a synthétisé des concepts énoncés par différents groupes.

Les travaux ont débuté après que le protocole H.323 eut été proposé par l'UIT. Il est vite apparu que l'initiative H.323 était insatisfaisante pour relier à grande échelle des réseaux de natures différentes. Les passerelles (*gateways*) proposées dans l'architecture H.323 sont des éléments complexes et coûteux, ce qui pose problème dans le monde IP. Plus le nombre de réseaux à traverser pour établir une communication est important, plus l'est aussi celui des passerelles sollicitées.

Progressivement, un certain nombre d'initiatives ont visé à disposer d'un réseau dont toutes les passerelles multimédias soient des composants simples. L'idée-force était de

relier ces passerelles à un contrôleur maître concentrant toute l'intelligence du réseau et centralisant les décisions selon le modèle maître-esclave.

L'architecture générale conceptualisant les entités de contrôleur et de passerelle multi-média, et plus généralement le modèle maître-esclave, fut initialement proposée dans le projet TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks) de l'ETSI. La première tentative de protocole de communication entre ces entités fut SGCP (Simple Gateway Control Protocol), en 1997. Proposée par Telcordia (anciennement BellCoRe, acronyme de Bell Communications Research), elle reçut le soutien de Cisco. TIPHON est le protocole précurseur de MGCP.

En 1998, l'opérateur de télécommunications Level 3 Communications (groupe XCOM Technologies) a mis en place un comité consultatif technique, ou TAC (Technical Advisory Council), réunissant une douzaine d'industriels renommés, comme Ericsson, Lucent, Nortel, Alcatel, 3Com et Cisco. Avec ces membres fondateurs, le TAC sera à l'origine de la conception d'un protocole de contrôle des entités réseau sur Internet, nommé IDCP (Internet Device Control Protocol). Ses spécifications seront présentées à l'UIT, à l'IETF et à l'ETSI.

En octobre 1998, avec le soutien d'importants constructeurs, tels que Cisco et Alcatel, le protocole MGCP a été standardisé à l'IETF par le groupe de travail MeGaCo (Media Gateway Control). Celui-ci réalisait la fusion des initiatives SGCP et IDCP. MGCP s'inscrit dans la droite ligne de la version 1.1 du protocole SGCP, qui servira de socle principal à MGCP.

En plus de dériver de SGCP, MGCP s'est enrichi de nombreuses fonctionnalités proposées dans IDCP. En octobre 1999, la RFC 2705 présentait la première version de MGCP. Elle sera rendue obsolète en janvier 2003 par la RFC 3435, qui sera complétée par la RFC 3661 en décembre 2003.

## H.248/MeGaCoP

Parallèlement à MGCP, Lucent Technologies proposait, en novembre 1998, MDPCP (Media Device Control Protocol), un protocole analogue, dont la conception présentait des avantages notables, que MGCP ne pouvait prendre en compte sans repenser son modèle.

Pour synthétiser les efforts des protocoles MGCP et MDPCP, le groupe de travail MeGaCo de l'IETF et le groupe d'étude 16 (*Study Group 16*) de l'UIT-T décidèrent de travailler de concert et de fédérer les différentes propositions émergentes au sein d'un protocole commun. Les travaux reprenaient ceux menés sur MGCP et y associaient un ensemble de nouveaux paramètres, de codes d'erreurs et de procédures inspirés de MDPCP.

L'UIT identifiera ce protocole sous la recommandation H.GCP (Gateway Control Protocol), devenue ensuite H.248, tandis que l'IETF référençait ce même protocole sous l'appellation MeGaCoP (MeGaCo Protocol). En août 2000, l'IETF publiait la RFC 2885, corrigée le même mois par la RFC 2886, qui présentait le protocole MeGaCoP version 0.8. En novembre 2000, la version 1 de MeGaCoP était publiée dans la

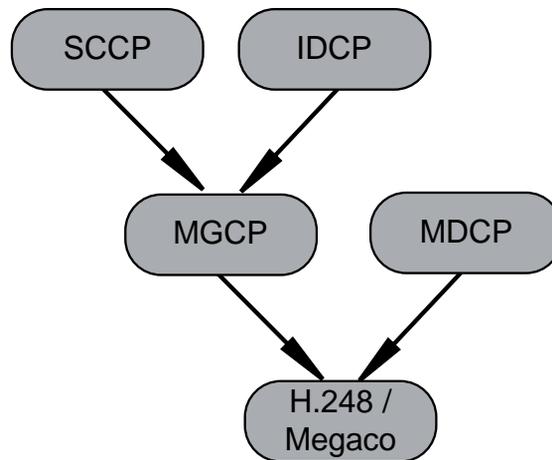
RFC 3015, rendant obsolètes les RFC 2885 et 2886. Depuis l'acceptation de cette RFC, le groupe de travail MeGaCo s'est dissout.

MeGaCoP reste assez remarquable dans sa tentative de conceptualisation et de nomenclature des principales notions abordées dans MGCP. Ainsi, de nouvelles terminologies sont apparues, certaines pour évoquer de nouvelles notions, d'autres uniquement pour renommer l'existant en harmonisant l'ensemble des concepts afin de les généraliser. C'est ainsi, notamment, que l'entité de contrôle des passerelles fut appelée Call Agent dans MGCP puis MGC (Media Gateway Controller) dans la terminologie MeGaCo.

La figure 5.2 illustre les héritages protocolaires entre ces différentes propositions.

**Figure 5.2**

*Héritages protocolaires des différents standards autour de MGCP*



Aujourd'hui, MGCP tient parfaitement son rôle, en dépit de l'apparition du protocole H.248.

## Architecture et fonctionnement

Pour communiquer entre deux réseaux de nature différente, il est nécessaire d'utiliser une passerelle. Cette entité prend en charge à la fois la signalisation pour l'établissement, la gestion et la terminaison de la communication, mais aussi la conversion des signaux pour l'adaptation des flux d'un réseau vers un autre. MGCP sépare ces deux aspects en entités distinctes, l'une pour contrôler les appels, l'autre pour appliquer le contrôle ordonné par la première entité.

MGCP fonctionne selon une architecture centralisée permettant de faire communiquer et de contrôler différentes entités appartenant à des réseaux distincts. Il se fonde sur l'hypothèse que les terminaux des utilisateurs peuvent être des composants de base, peu coûteux et sans aucune intelligence, réduits à des fonctionnalités élémentaires.

Les passerelles sont également des entités simples. En fournissant un service simple et générique, elles restent indépendantes de leur constructeur. Pour leur donner des directives permettant le traitement des services, ces passerelles multimédias sont reliées à une entité centrale. Le protocole MGCP assure le contrôle et l'échange de messages de signalisation entre ces passerelles, réparties dans un réseau IP, et le contrôleur de passerelles, chargé de l'administration et de la gestion dynamique des passerelles.

MGCP fait éclater le modèle architectural proposé avec H.323 en décomposant le rôle des passerelles et en externalisant toute leur intelligence sur une entité centrale.

Pour réaliser cette distinction, MGCP définit les entités suivantes :

- Le Call Agent, qui sert à piloter et administrer les passerelles de manière centralisée.
- Les passerelles, qui maintiennent la connectivité entre réseaux de nature différente.

## ***Le Call Agent***

Le Call Agent, également appelé contrôleur de passerelles multimédias ou encore SoftSwitch, selon une terminologie non officielle mais courante, a pour fonction de contrôler les passerelles et de concentrer toute l'intelligence ainsi que la prise de décision dans le réseau.

Entité logique, pouvant être localisée n'importe où dans le réseau, le Call Agent est spécifiquement responsable de l'établissement, de la maintenance et de la terminaison des appels établis entre des terminaux appartenant à des réseaux de nature différente.

Comme ces opérations sont initiées au niveau des passerelles multimédias, le Call Agent intervient pour contrôler l'activité de ces dernières et leur donner les directives de traitement de ces opérations. Il est en quelque sorte le maître d'œuvre et d'opération des communications entre les réseaux.

Dans la mesure où il contribue à centraliser le réseau autour de lui, le Call Agent est une entité fortement sollicitée. De ce fait, il devient un élément sensible dans le réseau, particulièrement en cas de panne. Néanmoins, cette centralisation n'intervient que pour arbitrer et assurer la maintenance et la gestion des échanges de signalisation. Elle n'entre pas en jeu dans les communications intra-réseau. En outre, pour gérer les pannes, il est plus simple de mettre en place des doublures, sous forme de Call Agents redondants, que de rendre toutes les passerelles multimédias redondantes.

Il est possible d'avoir plusieurs Call Agents, chacun ayant en charge un parc de passerelles multimédias. Par exemple, chaque opérateur peut gérer ses propres passerelles par un Call Agent propriétaire. Le protocole MGCP ne définissant pas de mécanisme de synchronisation entre Call Agents, on doit considérer indépendamment chaque Call Agent et les passerelles qu'il contrôle.

Fondamentalement, MGCP repose sur un modèle maître-esclave, et il n'est pas dans son objectif de fournir des mécanismes de communication entre les agents de contrôle, qui sont des entités de même nature, auxquelles le modèle maître-esclave ne convient pas.

Pour faire communiquer entre eux plusieurs Call Agents, un protocole tel que SIP peut être utilisé afin de négocier les paramètres de communication.

### **Les passerelles multimédias**

Selon le protocole MGCP, la notion de passerelle est assez floue et couvre un vaste ensemble de définitions, notamment les suivantes :

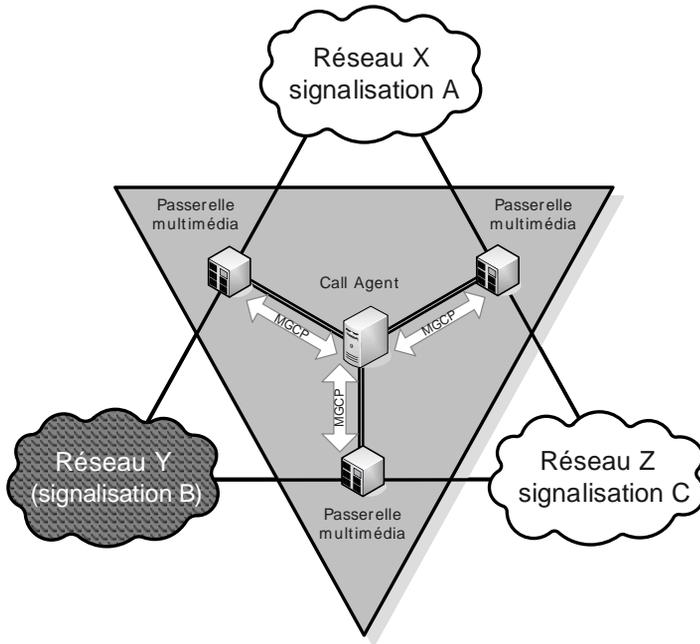
- Passerelle d'opérateur téléphonique, pour faire le lien entre un réseau téléphonique et un réseau IP. Les opérateurs de téléphonie alternatifs, par exemple, utilisent souvent le réseau RTC de l'abonné comme réseau de base puis basculent les flux de l'abonné vers un réseau IP (lequel présente l'avantage d'être à commutation de paquets, donc sans réservation de ressources, et ainsi moins coûteux qu'un réseau à commutation de circuits) sur de longues distances internationales, avant de basculer à nouveau les flux de l'appelant vers le réseau RTC auquel le terminal du destinataire est connecté.
- Passerelle résidentielle de type box (boîtier exploitant le modem, le câble ou les technologies xDSL), généralement mise à disposition par le FAI. Ce boîtier fait la liaison entre le réseau IP des utilisateurs et le réseau d'accès téléphonique de l'opérateur. Nous en verrons une illustration un peu plus loin dans ce chapitre.
- PBX d'entreprise faisant la liaison entre le réseau IP de l'entreprise et le réseau téléphonique RTC de l'opérateur. Au sein de l'entreprise, des téléphones IP ou des softphones peuvent être utilisés en interne pour exploiter les services complémentaires qu'offre le réseau IP et permettre la convergence des flux sur un support unique. Comme le réseau de l'entreprise est connecté à une liaison RTC, une passerelle est toutefois nécessaire.

Par rapport aux passerelles initialement prévues dans le protocole H.323, les passerelles multimédias sont simplement dépourvues de la fonctionnalité de traitement des appels. Elles s'en remettent pour cela au Call Agent. Néanmoins, elles conservent intact leur emplacement physique, à la frontière entre les deux réseaux de nature distincte, alors que le Call Agent peut être situé n'importe où, comme l'illustre la figure 5.3.

X, Y ou Z représentent des réseaux quelconques (RNIS, ATM, IP, RTC, etc.). Sur chacun de ces réseaux, le protocole de signalisation intra-réseau de son choix peut être utilisé, par exemple SIP ou H.323 dans un réseau IP, ou SS7 dans un réseau RTC. MGCP ne peut s'appliquer au sein de ces réseaux, mais seulement à leur périphérie afin d'assurer la gestion et le traitement des communications interréseau.

On observe deux niveaux de communications : l'un qui fait intervenir les réseaux et les passerelles multimédias et l'autre les passerelles multimédias et le Call Agent. Le protocole MGCP s'applique exclusivement à transmettre de la signalisation entre le Call Agent et les passerelles. Les flux de données multimédias (voix, vidéo, données) entre deux terminaux appartenant à des réseaux différents ne transitent jamais par le Call Agent. Une fois que le Call Agent en a donné l'autorisation, ces flux sont véhiculés de poste terminal à poste terminal, en passant uniquement par la passerelle.

**Figure 5.3**  
*L'architecture MGCP*



Le rôle de la passerelle multimédia est donc réduit à l'acheminement cohérent des données, ce qui implique qu'elle accomplisse les tâches suivantes :

- conversion du signal ;
- adaptation au support ;
- compression des données ;
- conversion de la signalisation ;
- multiplexage ;
- mise en paquets.

Les passerelles multimédias se retrouvent ainsi réduites à leur fonctionnalité première et fondamentale de transmission : elles travaillent au niveau du média lui-même et assurent le traitement des données, sans la logique de traitement. Toutefois, ces actions ne sont réalisables qu'en accord avec le Call Agent, dont les passerelles sont les exécutants.

Globalement, le mode de fonctionnement des passerelles est donc allégé par rapport à H.323 et le réseau devient constitué d'éléments simples et configurables.

Les communications MGCP passent systématiquement par le protocole UDP, choisi pour optimiser les délais de traitement des envois.

S'il n'est pas mentionné, le Call Agent utilise par défaut le port 2727 pour ses communications, tandis que les passerelles utilisent le port 2427.

## **Raisons d'être d'un nouveau protocole**

On peut s'interroger sur la nécessité d'adopter un nouveau protocole dans un contexte où le protocole SIP, réputé plus simple que H.323, a du mal à s'imposer compte tenu de la forte pénétration de ce dernier.

Cette section introduit différents aspects qui expliquent pourquoi une simple extension de SIP ou de H.323 n'était pas envisageable pour jouer le rôle de MGCP.

### **Centralisation et asymétrie**

Une extension du protocole SIP pour assurer les fonctionnalités de communication inter-domaine serait étrangère à la logique de fonctionnement de SIP. SIP est un protocole point-à-point, donc décentralisé.

MGCP veut fédérer les communications au niveau d'une entité maîtresse concentrant toute l'intelligence du réseau et imposant ses directives à l'ensemble des autres entités du réseau. L'architecture requise par ce modèle est donc celle d'un maître et d'esclaves. Pour cette raison, on dit que MGCP est un protocole asymétrique. Une seule entité a la vision de l'ensemble du réseau et la possibilité de l'administrer dans sa globalité. SIP est symétrique, et ses échanges sont effectués sur le modèle client-serveur. La vision de MGCP et celle de SIP sont donc intrinsèquement différentes.

H.323 est beaucoup plus proche du modèle centralisé. Mais cette centralisation est la cause des lourdeurs du protocole, souvent considéré pour cette raison comme non adapté au monde Internet. Une centralisation encore accrue, si les fonctionnalités de MGCP étaient portées dans H.323, ne serait guère possible. Le réseau deviendrait dépendant d'éléments architecturaux très sollicités, riches en fonctionnalités et coûteux. Il serait en outre complexe à gérer, maintenir et entretenir.

La vision MGCP consiste au contraire à proposer un réseau aussi simple que possible en reportant l'intelligence des passerelles sur un élément unique, le Call Agent, seule pièce maîtresse du réseau considéré dans son ensemble. Ainsi, si une passerelle tombe en panne, il est simple, rapide et peu onéreux de la remplacer. Les équipements de type passerelle sont facilement remplaçables, et la compatibilité est assurée à un niveau supérieur par le Call Agent. On est proche du modèle Internet, où le contrôle n'est pas effectué par le réseau lui-même, mais est relégué aux extrémités.

### **Indépendance et compatibilité**

MGCP a l'avantage d'être indépendant de tout autre protocole, puisqu'il est censé superviser n'importe lequel d'entre eux.

En choisissant un protocole additionnel, chaque acteur de la normalisation peut se concentrer sur son propre protocole en admettant la possibilité de communiquer avec le protocole concurrent afin de satisfaire le plus grand nombre de cas de figure possible.

Du reste, dans un contexte où plusieurs protocoles de signalisation existent au sein d'un réseau, l'association de l'UIT et de l'IETF est assez significative de la

volonté d'offrir un modèle générique sans imposer de protocole de signalisation sous-jacent.

MGCP permet de fédérer les signalisations. Par nature, il est donc pleinement compatible avec SIP et H.323. Non seulement, il n'entre pas en concurrence avec ces derniers, mais il ne se substitue aucunement à eux, puisqu'il prend en charge la signalisation entre réseaux exploitant différents protocoles de signalisation, et non la signalisation au sein de chacun d'eux.

Dans la version 5 du protocole H.323, les passerelles sont décomposées conformément au modèle proposé par MGCP. H.323 est ainsi parfaitement compatible avec MGCP.

Par bien des aspects, MGCP s'inspire du modèle défini par SIP. En particulier, tous deux utilisent un langage textuel, décrivent leur connexion par le protocole SDP et utilisent un jeu de codes de réponse comparable. De nombreux spécialistes estiment que MGCP et SIP sont complémentaires et forment un choix cohérent pour mettre en œuvre une architecture de ToIP, évinçant par cette simple considération l'association de MGCP avec H.323, que pourtant rien n'interdit.

### ***Exemple d'utilisation de MGCP chez les FAI***

Les FAI proposent des offres dites Triple-Play, incluant Internet, la télévision et la téléphonie. L'abonné est connecté au réseau téléphonique commuté RTC. C'est ce que l'on appelle la boucle locale, qui relie la prise téléphonique de l'abonné au local technique de l'opérateur historique. C'est dans ce local technique que les flux réseau sont déviés de l'opérateur historique vers le FAI de l'abonné.

Chez lui, l'abonné peut mettre en place un réseau IP domestique afin de faire communiquer ses différents terminaux (ordinateur, PDA, téléphone IP) et de partager une connexion Internet. Pour assurer la connectivité entre le réseau IP de l'abonné et le réseau RTC de la boucle locale, le FAI met à disposition de l'abonné une Internet Box. Celle-ci agit comme une passerelle entre le réseau de l'abonné et celui de l'opérateur en transcodant les flux IP en des flux RTC, et réciproquement.

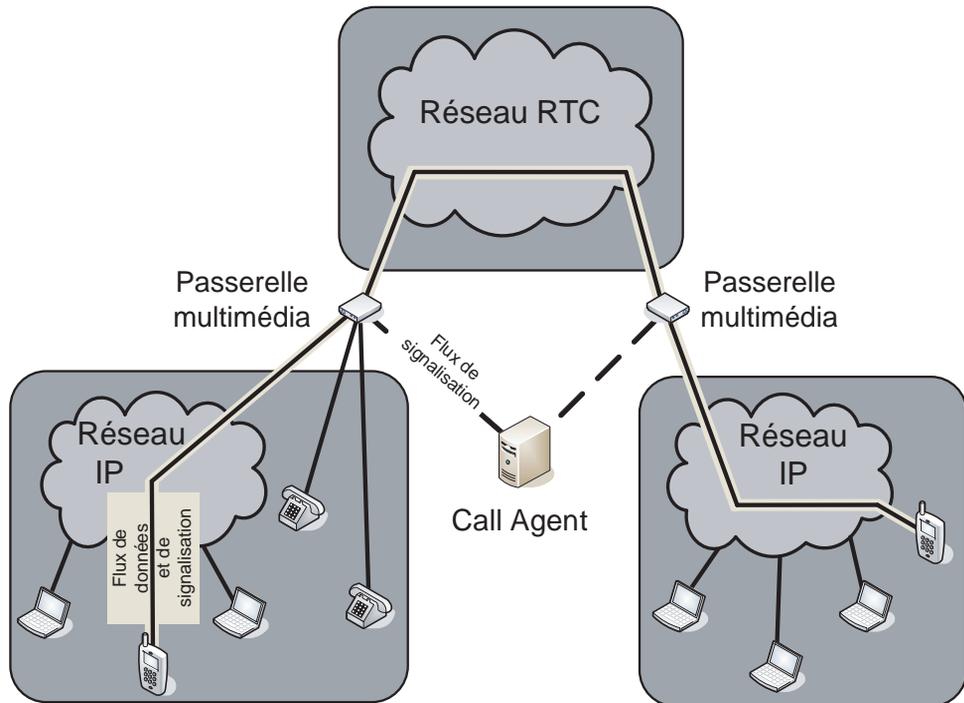
Pour l'opérateur, deux contraintes fondamentales doivent être gérées par la mise à disposition de ces boîtiers : ces boîtiers doivent être peu coûteux, puisqu'ils sont diffusés à très grande échelle, et ils doivent être configurables à distance. Par exemple, pour corriger une défaillance protocolaire du boîtier ou ajouter des fonctionnalités complémentaires, les boîtiers doivent être à portée de contrôle du FAI.

C'est exactement le rôle du protocole MGCP : les passerelles sont des composants simples, pilotés par une entité centrale, le Call Agent. C'est la raison pour laquelle ce protocole est couramment utilisé par les FAI français.

Pour la fourniture du service de téléphonie, les FAI doivent tenir compte des combinés téléphoniques traditionnels de leurs abonnés, qui ne sont généralement compatibles qu'avec le réseau RTC. Cela implique une convergence des réseaux RTC et IP au niveau local à l'abonné.

La figure 5.4 illustre ce fonctionnement dans une architecture considérablement simplifiée par rapport à la réalité. Le Call Agent contrôle toutes les passerelles. À ce titre, il est distinct de tous les réseaux. Dans la pratique, il est situé chez l'opérateur. Le réseau RTC n'est que la porte d'entrée vers le réseau Internet. Le FAI doit mettre en place de nouvelles passerelles depuis ce réseau RTC d'où proviennent les flux des terminaux des abonnés vers le réseau Internet.

**Figure 5.4**  
*MGCP chez les FAI*



Un boîtier d'opérateur n'est autorisé à émettre ou recevoir un appel qu'après en avoir fait la demande auprès du Call Agent. Celui-ci peut contrôler les appels et généralement mettre en place parallèlement un système de facturation, ainsi que des mécanismes de sécurité associés à l'appel.

### **Avantages et inconvénients de MGCP**

Comme indiqué précédemment, les avantages du protocole MGCP sont doubles : le réseau peut être configuré de manière centralisée, et les passerelles multimédias sont des éléments simples.

Le Call Agent a un rôle central de gestion des passerelles multimédias. Il offre ainsi le moyen de mettre à jour des fonctionnalités sur les passerelles, sans avoir besoin d'intervenir sur chacune d'elles. Le réseau devient ainsi facilement administrable à distance.

La création de nouveaux services est simplifiée, puisque leur implémentation et leur gestion sont automatiquement propagées. À l'inverse, le Call Agent peut imposer un traitement personnalisé selon des configurations paramétrables qui supportent le nomadisme des utilisateurs. Par exemple, un utilisateur peut avoir la possibilité de personnaliser sa sonnerie d'appel dans une base de données cliente, à laquelle accède le Call Agent. Lorsque l'utilisateur se déplace et se connecte sur un autre réseau, le Call Agent peut détecter les préférences de l'utilisateur dans la base de données, de façon à lui conserver la sonnerie qu'il avait configurée.

Dans la mesure où elles n'assument pas la logique de contrôle des appels, mais seulement les traitements de bas niveau, les passerelles sont des entités relativement simples et peu coûteuses. Si l'une d'elles tombe en panne, il est facile de la remplacer car il n'est pas nécessaire de reprogrammer toutes les configurations de la passerelle décrivant son état avant la panne : c'est le Call Agent qui se charge de configurer dynamiquement la passerelle.

### Inconvénients

MGCP propose une architecture centralisée chargée du contrôle dans le réseau. Par conséquent, le réseau est dépendant de cette entité centrale, qui constitue un point de vulnérabilité.

En cas de dysfonctionnement de ce serveur, le réseau tout entier devient défaillant puisque aucun contrôle ne peut plus y être effectué. Néanmoins, le protocole MGCP a prévu des procédures pour que, en cas de panne, une passerelle puisse basculer d'un contrôleur vers un autre.

Un autre inconvénient de MGCP est qu'il impose globalement une plus grande quantité de messages de signalisation. Les terminaux qui l'implémentent ne se connectent jamais directement entre eux, mais doivent impérativement au préalable en demander l'autorisation au centre de contrôle.

Ce mode de fonctionnement se distingue nettement de ceux des protocoles H.323 et SIP, qui permettent aux terminaux de communiquer entre eux sans faire intervenir d'entité tierce. Globalement, les protocoles H.323 et SIP assurent le contrôle des appels eux-mêmes, tandis que le protocole MGCP assure le contrôle des entités du réseau.

## Principes d'établissement d'une communication

On appelle *endpoint* un équipement dit de terminaison, qui représente soit la source soit la destination d'un message multimédia.

Un routeur réseau n'est pas un endpoint puisqu'il se contente d'acheminer des données, sans être à l'origine de l'envoi. Le Call Agent n'est pas non plus un endpoint, puisqu'il ne traite pas des messages multimédias.

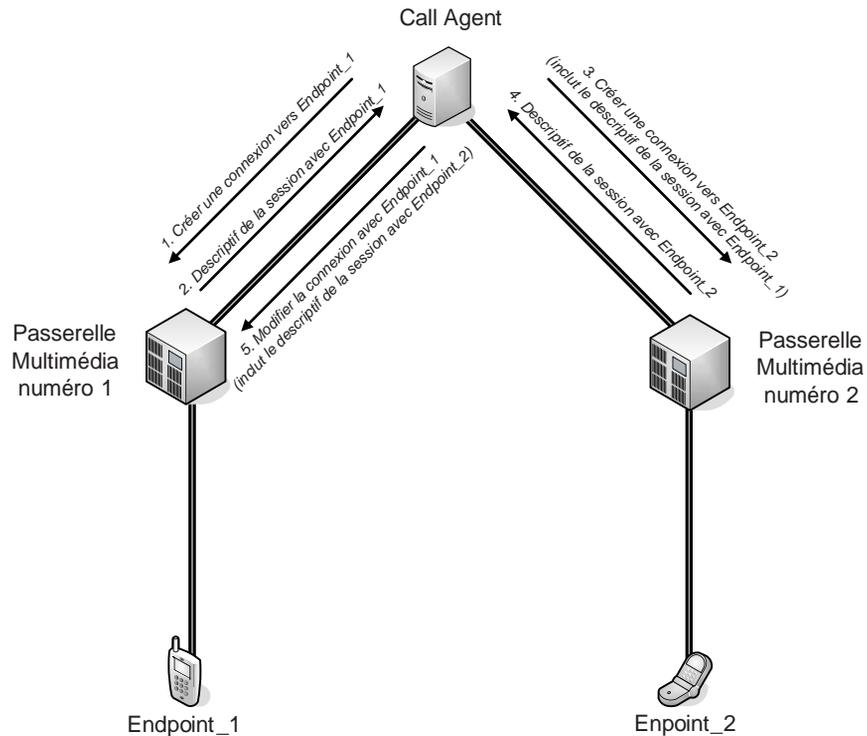
Dès lors qu'une entité participe aux échanges de médias et se place comme source ou destinataire de ces échanges, même si elle n'est pas la source initiale ou le destinataire

final et qu'elle ne joue qu'un rôle d'intermédiaire dans ces échanges, elle est considérée comme un endpoint. Les terminaux des utilisateurs sont des endpoints de référence.

Supposons que nous souhaitons connecter deux terminaux, appelés des endpoints. Chacun d'eux se trouve localisé derrière une passerelle multimédia. Ces deux passerelles sont elles-mêmes contrôlées par un Call Agent, comme l'illustre la figure 5.5.

**Figure 5.5**

*Mise en relation de deux endpoints*



Pour mettre en relation les deux endpoints, les cinq étapes suivantes sont nécessaires :

1. Requête de création de connexion vers la première passerelle. Le Call Agent sollicite la création d'une connexion avec un endpoint auprès de la passerelle concernée.
2. Réponse de la première passerelle. Celle-ci se charge de joindre le endpoint et lui attribue les ressources nécessaires à la communication : une session est créée entre la passerelle et le endpoint. En retour, la passerelle envoie au Call Agent un descriptif de la session créée. Ce descriptif contient l'ensemble des paramètres permettant de joindre le endpoint, incluant l'adresse IP de ce dernier, le port UDP sur lequel la communication est en attente et les codecs supportés.
3. Requête de création de connexion vers la seconde passerelle. Le Call Agent procède de la même façon pour le second endpoint et sa passerelle : il sollicite cette dernière en lui envoyant un message pour la création d'une connexion avec le second endpoint.

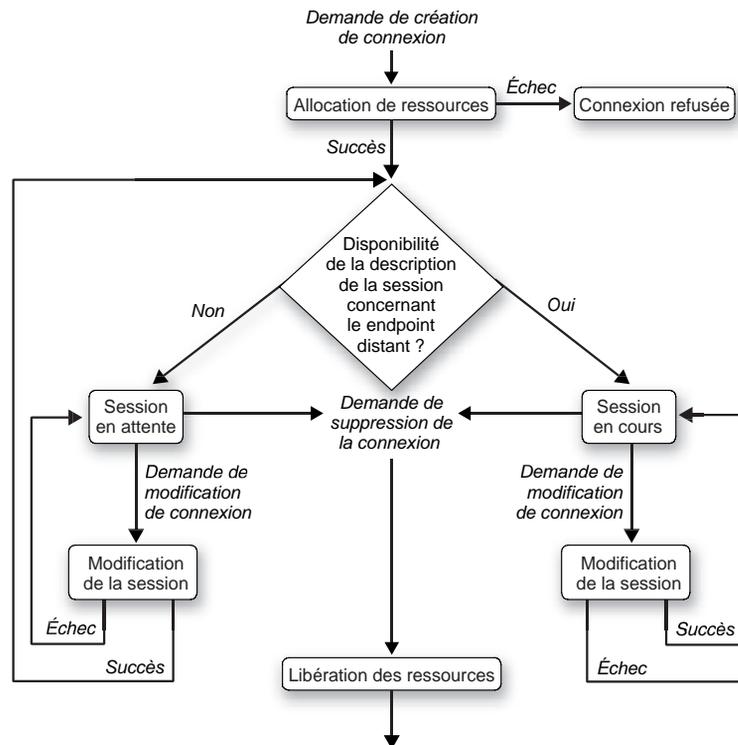
En plus, et dans le même message, le Call Agent lui fait parvenir le descriptif de session que lui a retourné la première passerelle.

4. Réponse de la seconde passerelle. La seconde passerelle joint le endpoint concerné et alloue les ressources nécessaires à cette communication. En retour, elle transmet au Call Agent un descriptif de session contenant les paramètres permettant de joindre le second endpoint.
5. Mise en relation des deux endpoints. Le Call Agent contacte la première passerelle et lui transmet le descriptif de la session retournée par la seconde passerelle. Comme une connexion existe déjà avec le endpoint, il n'est pas nécessaire de créer une nouvelle connexion. Il suffit de modifier celle qui existe et de la compléter. C'est donc une commande de modification qui est effectuée par le Call Agent.

Une fois ces étapes achevées, la communication débute dans les deux sens. Elle peut être modifiée à tout moment par le Call Agent, qui peut imposer, par exemple, un changement de codec, d'adresse IP ou de port. De même, le Call Agent peut mettre fin à la communication à tout moment en envoyant un message aux passerelles, qui doivent alors rompre les connexions.

On peut résumer tous les états possibles d'une passerelle multimédia comme illustré à la figure 5.6.

**Figure 5.6**  
Diagramme d'états  
d'une passerelle



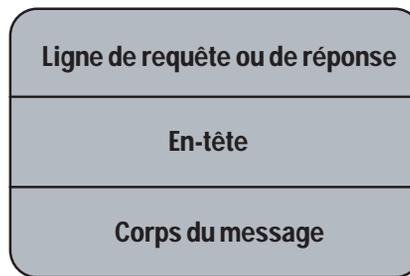
## Les messages MGCP

La communication avec MGCP obéit à un modèle de type client-serveur. Un message MGCP est soit une requête, soit une réponse à une requête. Il est constitué sous forme textuelle, ce qui simplifie son usage (traitement sans compilateur, donc plus rapide, et débogage immédiat), et présente plusieurs analogies avec le protocole SIP. Ainsi, une transaction MGCP est-elle constituée d'une requête et de la réponse à cette requête, éventuellement précédée de réponses temporaires.

Le format d'un message MGCP est illustré à la figure 5.7.

Figure 5.7

Format d'un message MGCP



Dans ce message, on distingue trois parties :

- Ligne de requête ou de réponse : notifie la commande à exécuter (s'il s'agit d'une requête) ou le résultat de la commande (s'il s'agit d'une réponse). C'est une partie indispensable.
- En-tête : spécifie la liste des paramètres du message. C'est une partie facultative.
- Corps du message : décrit les paramètres de la session à établir. C'est une partie facultative.

Plusieurs lignes peuvent constituer chacune des parties. On sépare chaque ligne par des retours chariot, ou CR (Carriage Return), et des sauts de ligne, ou LF (Line Feed), ou par des retours chariot seulement.

Notons que, dans la RFC 3435, la partie spécifiant la ligne de requête ou de réponse et celle spécifiant l'en-tête sont regroupées.

### Adressage des endpoints

L'adressage d'un endpoint est représenté dans un format semblable à l'e-mail, dans le respect de la RFC 821.

Sa syntaxe est la suivante :

*endpoint@domaine[:port]*

La partie *domaine* spécifie le nom de domaine absolu, conformément à la RFC 1034 (DNS), incluant le nom de la passerelle permettant d'accéder au domaine. Par exemple, un nom de domaine peut être :

*ma\_passerelle.mon\_domaine.fr*

Le nom de domaine peut aussi être spécifié par une adresse MAC ou une adresse IP, au format IPv4 ou IPv6, à condition de l'indiquer entre crochets.

La partie *endpoint* spécifie le nom de l'entité considérée. Elle est définie selon trois niveaux hiérarchiques séparés par le symbole /, de la façon suivante :

*niveau\_hierarchique\_1/niveau\_hierarchique\_2/niveau\_hierarchique\_3*

La hiérarchisation est de plus en plus spécifique au fur et mesure qu'on se déplace vers la droite du nom du endpoint. Ainsi, le *niveau\_hierarchique\_3* est plus précis que le *niveau\_hierarchique\_2*, lui-même plus précis que le *niveau\_hierarchique\_1*.

Les parties *endpoint* et *domaine* peuvent être formées de n'importe quel caractère en dehors des symboles espace, arobase et slash, qui font déjà office de séparateurs.

Les symboles de multiplication (\*) et dollar (\$) sont également interdits dans les parties *endpoint* et *domaine*, car ils possèdent une signification particulière. Le symbole de multiplication peut remplacer n'importe quelle autre chaîne constituée d'un ou de plusieurs caractères, tandis que le caractère dollar peut remplacer une et une seule lettre. Ce sont des caractères couramment utilisés avec cette signification dans les langages de programmation informatique. De cette manière, il est possible d'adresser plusieurs composants différents en même temps. Les parties *endpoint* et *domaine* peuvent avoir plus de 255 caractères. La spécification du port est facultative.

L'adressage d'un Call Agent est comparable à celui des endpoints. Il respecte la syntaxe suivante :

*callagent@domaine[:port]*

Les restrictions de nommage des parties *callagent* et *domaine* sont semblables à celles concernant les endpoints.

### Exemples de noms d'endpoint et de Call Agent

D'une manière générale, les noms des endpoints et des Call Agents peuvent être librement choisis en respectant les contraintes précédemment mentionnées.

L'appendice E de la RFC 3435 spécifie des conventions de nommage, dont nous donnons ci-dessous quelques exemples.

Pour adresser la ligne téléphonique analogique d'un terminal (par exemple ligne numéro 1), on utilise typiquement le nom suivant :

*aaln/1@ma\_passerelle.mon\_domaine*

où *aaln* désigne la ligne analogique d'un endpoint (*analog access line endpoint*).

Si l'on considère une seconde ligne téléphonique de terminal, on utilise le nom suivant :

*aaln/2@ma\_passerelle.mon\_domaine*

Si l'on souhaite envoyer un message s'affichant sur l'écran de la première ligne téléphonique considérée, le message est adressé de la façon suivante :

*disp/aaln/1@ma\_passerelle.mon\_domaine*

où *disp* désigne l'affichage (*display*).

Pour un Call Agent, on utilise :

*Mon\_Call\_Agent@Call\_Agent.mon\_domaine*

Le fait que ce nom soit générique est très pratique pour localiser le Call Agent de façon indépendante. Par exemple, si le Call Agent est déplacé dans un autre réseau, les passerelles n'ont pas besoin d'être mises à jour.

### **Identifiant de transaction**

Pour corréler une requête avec sa ou ses réponses, le protocole MGCP utilise un code appelé identifiant de transaction. De cette manière, une entité dispose de la possibilité d'émettre plusieurs requêtes successivement, sans en avoir reçu les réponses. L'entité peut déterminer à quelle requête fait référence une réponse en analysant la valeur de l'identifiant de transaction.

L'identifiant de transaction permet en outre à une entité (une passerelle ou le Call Agent) de repérer des éventuelles duplications de message.

Il existe deux possibilités qu'un message soit dupliqué :

- Pour une passerelle, il y a duplication entre deux messages reçus si les deux messages comportent le même identifiant de transaction.
- Pour un Call Agent, il y a duplication entre deux messages reçus si les deux messages comportent à la fois le même identifiant de transaction et le même nom de passerelle à l'origine du message.

On retiendra que les passerelles ne se concertant aucunement avant d'envoyer un message au Call Agent, elles peuvent utiliser un même identifiant de connexion entre elles. Ce n'est donc pas un critère discriminant au niveau du Call Agent. Par contre, pour chacune de ces requêtes émises, le Call Agent fait varier ses identifiants de transaction, indépendamment de la passerelle qui réceptionne le message.

L'identifiant de transaction correspond à un nombre strictement compris entre 0 et un million (ces deux valeurs n'étant pas incluses). Comme ces valeurs sont limitées, les identifiants peuvent être réutilisés, mais au minimum trois minutes après l'utilisation de ce code.

## Paramètres généraux pour les requêtes et les réponses

Les en-têtes et corps d'un message sont communs à tous les messages MGCP.

### En-tête d'un message

Cette partie est, selon les messages, obligatoire, optionnelle ou interdite. Elle mentionne des attributs caractérisant le message.

Le format général d'un paramètre d'en-tête respecte le modèle suivant :

*nom\_paramètre:valeur\_paramètre*

Le nom du paramètre est formé d'un ou de deux caractères (lettre ou chiffre). Le tableau 5.1 fournit la liste de tous les paramètres possibles.

**Tableau 5.1 – Paramètres d'en-tête d'un message MGCP**

Paramètre	Code	Description
BEARERINFORMATION	<i>B</i>	Définit des informations sur le codage des données envoyées ou reçues. Le seul attribut défini pour ce paramètre est l'encodage, représenté par le caractère <i>e</i> et dont la valeur peut être soit <i>A</i> , soit <i>mu</i> . On trouve donc couramment la valeur <i>B:e:mu</i> . D'autres attributs peuvent étendre ce paramètre.
CALL-ID	<i>C</i>	Identifiant d'appel. C'est une chaîne de caractères hexadécimaux, comportant au plus 32 caractères. Par exemple <i>C:1234567abc</i> .
CAPABILITIES	<i>A</i>	Liste les capacités du endpoint (incluant les paquetages, les modes de connexion et éventuellement les extensions supportés). Cela inclut notamment les éléments suivants : <ul style="list-style-type: none"> <li>– liste des codecs supportés ;</li> <li>– liste des réseaux supportés ;</li> <li>– durée de mise en paquet ;</li> <li>– bande passante nécessaire ;</li> <li>– suppression d'écho (supportée ou non) ;</li> <li>– suppression des silences (supportée ou non) ;</li> <li>– réservation de ressources (supportée ou non) ;</li> <li>– sécurité (cryptage du média ou non) ;</li> <li>– liste des paquetages supportés ;</li> <li>– liste des modes de connexion supportés.</li> </ul>
CONNECTIONID	<i>I</i>	Liste les identifiants de connexion pour toutes les connexions existantes sur le endpoint.
CONNECTIONMODE	<i>M</i>	Liste les modes de connexion supportés par le endpoint. On distingue les modes suivants : <ul style="list-style-type: none"> <li>– <i>sendonly</i> : émission de paquet seulement ;</li> <li>– <i>recvonly</i> : réception de paquet seulement ;</li> <li>– <i>sendrecv</i> : émission et réception de paquet ;</li> <li>– <i>confrence</i> : conférence avec plusieurs intervenants ;</li> <li>– <i>inactive</i> : communication inactive ;</li> <li>– <i>loopback</i> : bouclage ;</li> <li>– <i>conttest</i> : test de continuité ;</li> <li>– <i>netwloop</i> : bouclage du réseau ;</li> <li>– <i>netwtest</i> : test de continuité du réseau.</li> </ul>

Tableau 5.1 – Paramètres d'en-tête d'un message MGCP (*suite*)

Paramètre	Code	Description
CONNECTIONPARAMETERS	<i>P</i>	Liste les paramètres de connexion supportés par le endpoint.
DETECTEVENTS	<i>T</i>	Liste l'ensemble des événements qui doivent être détectés.
DIGITMAP	<i>D</i>	Mentionne le plan de numérotation utilisé par le endpoint. Ce paramètre est vide si le endpoint n'a pas de plan de numérotation.
EVENTSTATES	<i>ES</i>	Indique les états pour lesquels un événement est contrôlable.
LOCALCONNECTIONOPTIONS	<i>L</i>	Liste les options utilisées par le endpoint local, incluant le type de codec utilisé, le type de réseau ( <i>LOCAL</i> pour un réseau local, <i>IM</i> pour Internet, etc.), le débit, la qualité de service, le type de cryptage des flux, etc.
MAXMGCPDATAGRAM	<i>MD</i>	Indique la taille maximale d'un datagramme MGCP qui peut être supporté par le endpoint (excluant les couches inférieures à MGCP). Le support de ce paramètre est optionnel. L'unité est l'octet.
NOTIFIEDENTITY	<i>N</i>	Indique l'entité notifiée sur le endpoint.
OBSERVEDEVENTS	<i>O</i>	Liste les événements observés par le endpoint.
PACKAGELIST	<i>PL</i>	Liste les paquetages supportés (avec le numéro de version du paquetage) par le endpoint. Le support de ce paramètre est optionnel.
QUARANTINEHANDLING	<i>Q</i>	Liste les événements qui doivent être ignorés temporairement.
REASONCODE	<i>E</i>	Indique la valeur du dernier message de code de retour expliquant le traitement de la requête retourné par le Call Agent vers une passerelle avec les commandes <i>RESTARTINPROGRESS</i> ou <i>DELETECONNECTION</i> . Retourne la valeur <i>000</i> si l'état du endpoint est normal.
REQUESTEDEVENTS	<i>R</i>	Liste un ensemble d'événements surveillés, avec l'action (ou les actions) à entreprendre lorsque l'événement survient (si aucune action n'est mentionnée, l'action par défaut est activée).
REQUESTEDINFO	<i>F</i>	Liste toutes les données qui sont sollicitées.
REQUESTIDENTIFIER	<i>X</i>	Retourne l'identifiant de requête de la dernière commande <i>NOTIFICATIONREQUEST</i> reçue. Si aucune requête de ce type n'a été reçue, la valeur <i>0</i> est retournée.
RESPONSEACK	<i>K</i>	Liste les identifiants de toutes les transactions qui ont été validées. Une plage d'identifiants peut être mentionnée avec le symbole de tiret, par exemple <i>K:12315-12350, 12355, 12399</i> . Dans ce cas, les transactions d'identifiants <i>12315</i> à <i>12350</i> , ainsi que <i>12355</i> et <i>12399</i> sont acquittées.
RESTARTDELAY	<i>RD</i>	Indique le délai (en seconde) avant redémarrage.
RESTARTMETHOD	<i>RM</i>	Indique le redémarrage du endpoint (après le délai spécifié dans le paramètre <i>RESTARTDELAY</i> ).
SECONDCONNECTIONID	<i>I2</i>	Identifiant de la connexion
SECONDENDPOINTID	<i>Z2</i>	Identifiant du endpoint
SIGNALREQUESTS	<i>S</i>	Liste tous les signaux actifs.
SPECIFICENDPOINTID	<i>Z</i>	Identifiant (au format respectant la RFC 821) composé d'une chaîne de caractères arbitraire, suivi, après une arobase, du nom de domaine de la passerelle à laquelle le endpoint est associé.
REMOTECONNECTION- DESCRIPTOR	<i>RC</i>	Description de la session distante
LOCALCONNECTION- DESCRIPTOR	<i>LC</i>	Description de la session locale

## Corps d'un message

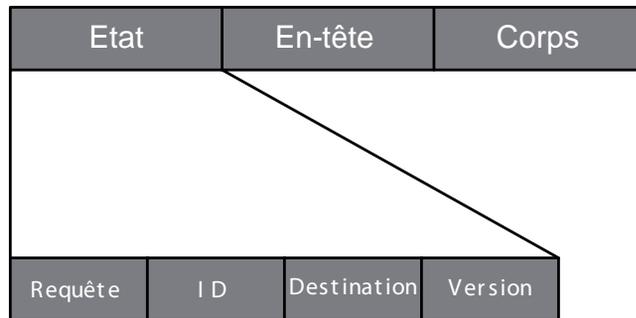
Cette partie est, selon les messages, obligatoire, optionnelle ou interdite. Elle mentionne des attributs relatifs à la communication sollicitée. Elle est implémentée en utilisant le protocole SDP, introduit au chapitre précédent dédié à SIP.

## La ligne d'état MGCP

La ligne d'état est constituée des quatre éléments suivants, illustrés à la figure 5.8 :

- Requête : indique l'action qui va être entreprise par ce message.
- Identifiant : tel qu'il a été présenté précédemment.
- Destination : spécifie l'adresse de la ou des destinations concernées par le message.
- Version : indique la version du protocole MGCP utilisée.

**Figure 5.8**  
*Détail de la ligne  
d'état MGCP*



## Les requêtes

Le protocole MGCP définit neuf requêtes permettant de spécifier l'action à entreprendre.

Les commandes sont lancées entre le Call Agent et les passerelles (Media Gateway). Comme MGCP est un protocole de type maître-esclave, toutes les entités n'ont pas des possibilités comparables, et ces commandes ne peuvent être lancées qu'à l'initiative de l'une de ces entités, soit le Call Agent, soit la Media Gateway.

On distingue donc deux catégories de commandes : celles qui sont lancées par le Call Agent vers une ou plusieurs passerelles et celles qui vont dans l'autre sens, de la passerelle vers le Call Agent.

À chaque requête correspond un code en quatre lettres de caractères ASCII, qui permet de condenser la taille de la requête. Les neuf requêtes et leur code respectif sont récapitulés au tableau 5.2.

Le protocole MGCP étant extensible, d'autres requêtes pourront venir l'enrichir dans les prochaines versions.

Nous décrivons brièvement ci-après ces neuf requêtes fondamentales.

Tableau 5.2 – Format des neuf requêtes MGCP

Format complet	Format abrégé
AUDITCONNECTION	AUCX
AUDITENDPOINT	AUEP
CREATECONNECTION	CRCX
DELETECONNECTION	DLCX
ENDPOINTCONFIGURATION	EPCF
MODIFYCONNECTION	MDCX
NOTIFICATIONREQUEST	RQNT
NOTIFY	NTFY
RESTARTINPROGRESS	RSIP

### Du Call Agent vers les passerelles

Le Call Agent peut agir sur les passerelles dont il a le contrôle par l'intermédiaire de sept commandes.

#### CreateConnection

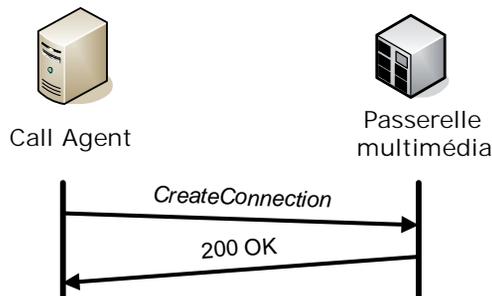
Cette commande permet de créer une connexion sur un endpoint à travers une passerelle. L'option LOCALCONNECTIONOPTIONS permet de définir une QoS.

Pour l'établissement et la libération des connexions, MGCP se sert de signaux et d'événements.

La figure 5.9 présente un scénario de création d'une connexion MGCP entre deux entités.

Figure 5.9

Création d'une connexion



#### ModifyConnection

Cette commande permet de modifier les paramètres associés à une connexion déjà établie.

Par exemple, il est possible de modifier le codec utilisé ou le temps de paquets.

### DeleteConnection

Cette commande demande la terminaison d'une connexion établie.

### NotificationRequest

Cette commande demande à une passerelle de surveiller des événements particuliers concernant un terminal. Par exemple, le Call Agent peut demander d'être prévenu lorsqu'un terminal répond à un appel ou lorsque sa ligne est libre ou qu'elle devient occupée ou encore lorsque la tonalité d'un fax retentit sur un poste.

De cette manière, le Call Agent donne des directives de contrôle sur lesquelles il est susceptible d'intervenir. Par défaut, la passerelle n'a pas à alerter le Call Agent de tous les événements qu'elle détecte. Elle se contente de remonter les événements demandés.

Une requête de notification d'événements se présente généralement sous la forme suivante :

*Si événement(s)*

*Alors action(s)*

Cela revient à agir suivant les actions spécifiées lorsque les événements indiqués surviennent. Une action peut être réduite à la seule indication de l'événement, mais peut aussi imposer que cet événement soit ignoré ou qu'il provoque une modification de codec, par exemple.

### AuditEndpoint

Cette commande demande la détection d'informations concernant un terminal.

Par exemple, un Call Agent peut solliciter une passerelle en charge d'un terminal pour savoir si le terminal est présent dans le réseau ou non, s'il a une communication en cours ou s'il est disponible, ou pour connaître les capacités du terminal (quels codecs et débit il peut supporter notamment) et la sonnerie qu'il utilise.

Toutes ces informations sont déterminées par la passerelle, généralement en utilisant le protocole SNMP (Simple Network Management Protocol).

### AuditConnection

Cette commande demande la détection de paramètres concernant une connexion.

Par exemple, un Call Agent peut solliciter une passerelle en charge d'un terminal pour savoir si une connexion existe ou non et connaître les types de flux (voix, vidéos, données), codecs et protocoles utilisés dans la communication.

### EndpointConfiguration

Cette commande est utilisée pour la configuration du type de codage des flux qui sont reçus par un terminal téléphonique sur le lien téléphonique traditionnel (c'est-à-dire le lien circuit, et non IP).

## D'une passerelle vers le Call Agent

Une passerelle dispose de trois commandes pour communiquer avec le Call Agent, dont l'une est similaire à celle qu'utilise le Call Agent.

### DeleteConnection

En principe, c'est le Call Agent qui doit indiquer la terminaison d'un appel et non l'inverse. Nous avons présenté cette commande précédemment dans ce cadre.

Dans certains cas, la passerelle peut rencontrer des problèmes techniques qui l'empêchent de maintenir la communication et la contraignent à achever la connexion. Cette commande permet d'indiquer au Call Agent qu'un événement non prévu a interrompu la connexion.

### Notify

Cette requête fait suite à une requête RQNT envoyée par le Call Agent. Elle indique que l'événement pour lequel le Call Agent avait sollicité une alerte est survenu.

### RestartInProgress

La passerelle peut avertir le Call Agent de l'indisponibilité d'un ou de plusieurs terminaux d'extrémité au moyen de cette commande.

Le Call Agent peut décider de tester les terminaux et de les mettre hors service.

## Destination

La destination est spécifiée selon le format d'adressage que nous détaillons plus loin dans ce chapitre.

## Version

L'indication d'une version permet de s'assurer de la compatibilité entre les entités communicantes.

Le récepteur peut interpréter le message s'il utilise la même version du protocole MGCP. Pour spécifier la version du protocole MGCP dans un message de requête, le mot-clé MGCP doit précéder le numéro de version avec une espace comme séparateur. Par exemple, pour la version 1.0 actuellement utilisée, on indique MGCP 1.0.

Optionnellement, il est possible d'ajouter à la suite une nouvelle espace suivie d'un message textuel représentant un profil. Le profil est utile afin de distinguer différentes catégories d'utilisateurs et de leur accorder des droits et des restrictions particulières.

En recevant ce message, le récepteur doit adapter son comportement selon le profil renseigné. Notamment, on peut imaginer que l'appel soit interdit sur certains profils ou nécessite une authentification particulière.

Un message de requête complet serait de la forme suivante :

```
CRCX      1204      aa1n/1@ma_passerelle.mon_domaine.fr      MGCP 1.0

C: A3C47F21456789F0
L: p:10, a:PCMU
M: recvnly
```

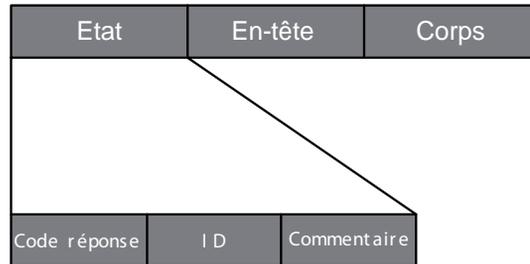
## Les réponses MGCP

Toutes les requêtes MGCP sont acquittées par un message de réponse.

Le format de ces messages de réponse est illustré à la figure 5.10.

**Figure 5.10**

*Format des réponses*



Comme pour SIP, les messages de réponse à une requête sont envoyés par un code de retour à trois chiffres. Là aussi, on distingue plusieurs catégories de codes de retour, assez comparables à ceux de SIP.

Le premier chiffre d'un code de retour désigne la catégorie de code de retour à laquelle le code appartient. Le tableau 5.3 indique quelques codes d'état qui ont été définis et les catégories auxquelles ils appartiennent.

**Tableau 5.3 – Principaux codes d'état des réponses MGCP**

Code d'état	Commentaire
<i>0xx – Messages d'acquiescement</i>	<i>La requête a bien été reçue.</i>
000	Réponse d'acquiescement (indique seulement la réception de la requête).
<i>1xx – Message d'information</i>	<i>C'est une réponse temporaire, qui informe l'émetteur. Une réponse définitive sera émise plus tard.</i>
100	La requête est en cours de traitement.
101	La requête est en attente. Elle sera traitée dès que les requêtes qui la précèdent auront été exécutées.

Tableau 5.3 – Principaux codes d'état des réponses MGCP (*suite*)

Code d'état	Commentaire
<i>2xx – Message de succès</i>	
La requête a été reçue, comprise et acceptée par le serveur.	
200	Requête exécutée avec succès. N'importe quelle requête peut être validée par ce code de retour.
250	La requête DELETECONNECTION a été exécutée avec succès (autrement dit, la connexion a bien été supprimée).
<i>4xx – Message signalant une erreur temporaire</i>	
La même requête pourra éventuellement être envoyée plus tard.	
400	Erreur temporaire qui n'est pas précisée.
401	Le téléphone est décroché.
402	Le téléphone est raccroché.
403	Les ressources sont insuffisantes pour traiter la requête.
404	La bande passante est insuffisante pour satisfaire la requête.
405	L'équipement est en train de redémarrer.
406	Dépassement de délai : la requête n'a pu être exécutée dans des délais raisonnables de traitement.
407	La requête a été annulée par un événement externe (par exemple, une requête DELETECONNECTION peut interrompre une requête MODIFYCONNECTION).
409	Le endpoint est surchargé pour le moment.
410	Aucune entité n'est disponible pour prendre en charge la requête.
<i>5xx – Message signalant une erreur permanente</i>	
Cette requête ne sera jamais prise en charge.	
500	Le endpoint n'est pas reconnu.
501	Le endpoint n'est pas prêt (éventuellement il ne fonctionne pas).
502	Les ressources du endpoint ne lui permettent pas de prendre en charge la requête.
503	L'astérisque (utilisé pour l'adressage des endpoints) est trop compliqué.
504	La commande n'est pas reconnue ou n'est pas supportée.
505	Le paramètre REMOTECONNECTIONDESCRIPTOR n'est pas supporté. Cette réponse devrait être déclenchée lorsqu'un champ du descripteur REMOTECONNECTIONDESCRIPTOR n'est pas supporté.
506	Impossible de satisfaire les paramètres LOCALCONNECTIONOPTIONS et REMOTECONNECTIONDESCRIPTOR en même temps. En principe, cette réponse est déclenchée lorsque des champs de ces paramètres présentent un conflit.
507	Une fonctionnalité non spécifique n'est pas supportée. Cette réponse n'est pas recommandée, car trop générale.
508	La liste des événements que la requête demande d'ignorer ( <i>quarantine handling</i> ) est inconnue ou n'est pas supportée.

Tableau 5.3 – Principaux codes d'état des réponses MGCP (suite)

Code d'état	Commentaire
509	Le paramètre REMOTECONNECTIONDESCRIPTOR présente une erreur. Cette réponse devrait être déclenchée lorsqu'un champ du descripteur REMOTECONNECTIONDESCRIPTOR présente une erreur syntaxique ou sémantique.
510	Une erreur protocolaire non spécifique a été détectée. Cette erreur ne doit être utilisée qu'en dernier recours, car elle est trop générale.
511	La commande comporte une extension qui n'a pu être reconnue.
512	La passerelle n'est pas capable de détecter l'un des événements que la requête sollicite.
513	La passerelle n'est pas capable de générer l'un des signaux que la requête sollicite.
514	La passerelle n'est pas capable d'envoyer l'annonce que la requête sollicite.
515	Le Connection-Id est incorrect (ne fait pas référence à une valeur référencée)
516	Le Call-Id est incorrect (c'est-à-dire que le Connection-Id n'est pas associé à ce Call-Id) ou bien il est inconnu.
517	Le mode utilisé est incorrect ou non supporté.
518	Le paquetage n'est pas supporté ou est inconnu. La liste des paquetages disponibles est généralement spécifiée dans le message.
519	Le endpoint n'a pas de plan de numérotation.
520	Le endpoint est en train de redémarrer. L'erreur 405 est préférable si le redémarrage n'est pas persistant, ce qui est le plus souvent le cas. Ce code est surtout mentionné pour assurer la compatibilité entre les versions.
521	Le endpoint est redirigé vers un autre Call Agent.
522	L'événement ou le signal mentionné dans la requête n'existe pas.
523	L'action demandée est inconnue ou la combinaison d'actions n'est pas permise.
524	Le paramètre LOCALCONNECTIONOPTIONS comporte des champs contradictoires.
525	Le paramètre LOCALCONNECTIONOPTIONS comporte une extension inconnue.
526	La bande passante n'est pas suffisante. Indique en principe, un manque de bande passante temporaire. Si la bande passante demandée est trop importante à obtenir sur la ligne considérée, une erreur 404 est préférable.
527	Le paramètre REMOTECONNECTIONDESCRIPTOR n'a pas été spécifié.
528	La version du protocole MGCP utilisée dans la requête est incompatible.
529	Une erreur matérielle interne a été détectée.
530	Erreur avec un protocole de signalisation CAS.
531	Erreur sur un ensemble de faisceaux.
532	Le paramètre LOCALCONNECTIONOPTIONS contient des valeurs qui ne sont pas supportées.
533	La réponse est trop longue.
534	Échec lors de la négociation de codec.

Tableau 5.3 – Principaux codes d'état des réponses MGCP (suite)

Code d'état	Commentaire
535	La période de paquetsisation est incorrecte.
536	La méthode RESTARTMETHOD n'est pas supportée ou est inconnue.
537	L'extension du plan de numérotation est inconnue ou n'est pas supportée.
538	Un paramètre de signal ou d'événement est incorrect (inconnu, non supporté, erroné ou manquant dans la requête).
539	Un paramètre de commande est invalide ou n'est pas supporté.
540	La limite du nombre de connexion par endpoint a été dépassée.
541	Le paramètre LOCALCONNECTIONOPTIONS est invalide ou n'est pas supporté.

Les codes de retour numérotés de 800 à 899 et 903 à 905 inclus sont réservés pour les paquetages.

Les codes de messages non définis sont interprétés selon la correspondance établie au tableau 5.4.

Tableau 5.4 – Interprétation des codes d'erreur inconnus

Code d'erreur inconnu commençant par le chiffre	Interprété comme s'il s'agissait du code
0	000
1	100
2	200
3	521
4	400
5, 6, 7, 8 ou 9	510

Un message de réponse complet serait de la forme suivante :

200	1204	OK
<pre> I: FDE234C8  v=0 o=- 25678 753849 IN IP4 128.96.41.1 s=- c=IN IP4 128.96.41.1 t=0 0 m=audio 3456 RTP/AVP 96 a=rtpmap:96 G726-32/8000 </pre>		

## Conclusion

Comme il se place en parallèle des protocoles de signalisation intérieurs des réseaux, MGCP ne souffre pas vraiment de concurrence. Il est du reste le protocole de référence pour les fournisseurs d'accès à Internet, qui l'utilisent afin de contrôler les équipements qu'ils mettent à disposition des utilisateurs.

Tandis que l'avenir des protocoles H.323 et SIP semble se profiler avec la spécification d'un protocole de nouvelle génération, H.325, conçu par l'UIT pour simplifier notamment la gestion des équipements de contrôle, de la qualité de service et du passage par les pare-feu d'entreprise, l'avenir de MGCP semble quant à lui serein.

Son successeur annoncé MeGaCo existe depuis l'année 2000, sans véritablement susciter d'intérêt chez les équipementiers ni manifester de véritables innovations qui justifieraient un changement de protocole.



# 6

## La qualité de service

---

Comme expliqué en début d'ouvrage, pour transporter de la parole téléphonique, il faut que le temps de transport de bout en bout soit limité puisque nous avons affaire à une application avec interaction humaine. Cette limitation est d'une centaine de millisecondes pour obtenir une très bonne qualité et jusqu'à 300 ms pour une conversation passant par un satellite géostationnaire.

Pour obtenir ces temps de réponse, il faut que le réseau offre une qualité de service. Plusieurs solutions peuvent être mises en œuvre pour cela selon deux grandes directions : un contrôle effectué au niveau applicatif et un contrôle effectué au niveau réseau.

Dans le premier cas, l'application s'adapte au réseau. Si le réseau est chargé, l'adaptation s'effectue en diminuant le débit du flux. Dans le second cas, c'est le réseau qui s'adapte à l'application. Il faut en ce cas demander au réseau un SLA (Service Level Agreement), qui, compte tenu du débit et des contraintes temporelles, assure la qualité de service demandée.

Nous examinons dans ce chapitre plusieurs propositions appartenant à l'une ou l'autre de ces solutions. Dans la première, nous nous pencherons sur RTP/RTCP et dans la seconde sur IntServ, DiffServ et MPLS/GMPLS. Avant cela, nous expliquerons pourquoi les protocoles de transport TCP et UDP sont peu qualifiés pour assurer ces fonctions.

### Le contrôle et les protocoles de transport

Avec quel protocole transporter les flux de ToIP et plus généralement tous les flux multimédias, soumis à des contraintes temporelles particulièrement fortes dans les réseaux IP ?

Le protocole TCP exige de nombreuses procédures de contrôle, qui le rendent peu adapté au transport des données multimédias avec de fortes contraintes de délai.

À l'inverse, le protocole UDP ne propose aucun mécanisme de contrôle, ce qui ne le qualifie guère pour le traitement des données multimédias.

### ***TCP et le transport de données multimédias temps réel***

Le protocole TCP implémente plusieurs mécanismes de contrôle :

- Contrôle de séquence. Chaque trame est numérotée au niveau de l'émetteur. Cela permet de reconstituer l'ordre des trames au niveau du récepteur, grâce à l'estampille séquentielle.
- Contrôle de flux. Un mécanisme de fenêtrage limite le nombre de paquets qu'il est possible d'émettre.
- Contrôle d'erreur. Le récepteur envoie un message d'acquiescement pour toutes les trames reçues. Il peut exister des acquiescements cumulatifs, permettant d'acquiescer plusieurs trames en même temps. Dans le cas où le récepteur reçoit des trames qui ne sont pas intègres, par exemple si une erreur est détectée lors du contrôle de redondance, il ne les acquiesce pas. Si certaines trames de l'émetteur ne reçoivent pas d'acquiescement passé un certain délai, appelé temporisateur de réémission, l'émetteur considère que ces trames sont perdues dans le réseau (un routeur saturé détruit les paquets qui arrivent en surplus) ou que le récepteur ne les a pas reçues de façon correcte. Il entreprend alors de retransmettre toutes les trames qui n'ont pas été acquiescées. De cette façon, l'intégralité des trames est nécessairement reçue, et la fiabilité des échanges comme l'intégrité des données sont garanties.
- Contrôle de congestion. Des mécanismes (appelés Slow Start et Congestion Avoidance) sont utilisés pour augmenter progressivement le débit d'envoi des données au niveau de l'émetteur. Le débit progresse par paliers successifs. Dès qu'un palier est atteint, le suivant est accessible, et ainsi de suite jusqu'à atteindre la limite fixée. Dans le cas où un palier n'est pas correctement validé parce que les acquiescements des trames envoyées ne parviennent plus jusqu'à l'émetteur, le débit est automatiquement abaissé jusqu'à son palier le plus bas. En effet, si un seuil pose problème, il ne sert à rien d'aller au suivant, car, avec un débit plus important, les erreurs risquent de se multiplier, aggravant l'état de saturation du réseau. En repartant d'un débit très faible, l'émetteur allège la charge du réseau susceptible de le stabiliser et de réguler son état avant que l'émetteur le sollicite. À nouveau, à partir du seuil le plus bas, la procédure d'augmentation de débit progressive est enclenchée.

Toutes ces fonctionnalités assurent un service de transport fiable, mais posent globalement deux problèmes. D'une part, elles engendrent un surplus de données important. D'autre part, ce n'est pas la fiabilité qui est l'élément le plus important dans les communications temps réel, mais le temps.

Du fait des contrôles, chaque trame est enrichie d'un en-tête TCP. Dans les transmissions temps réel, les trames sont de petite taille, car le temps de mise en paquets par l'émetteur retarde la diffusion chez le récepteur. De plus, comme la diffusion doit se faire très progressivement, les paquets sont découpés par petits morceaux. Toutes les trames ont donc un surplus conséquent de données, appelé overhead, constitué par l'en-tête TCP, ce

qui est problématique pour la charge globale du réseau, dont les capacités sont limitées. Un interlocuteur, au lieu d'utiliser toute sa bande passante pour y coder la voix avec une bonne qualité, doit se contenter d'une qualité réduite afin de permettre l'ajout des données TCP.

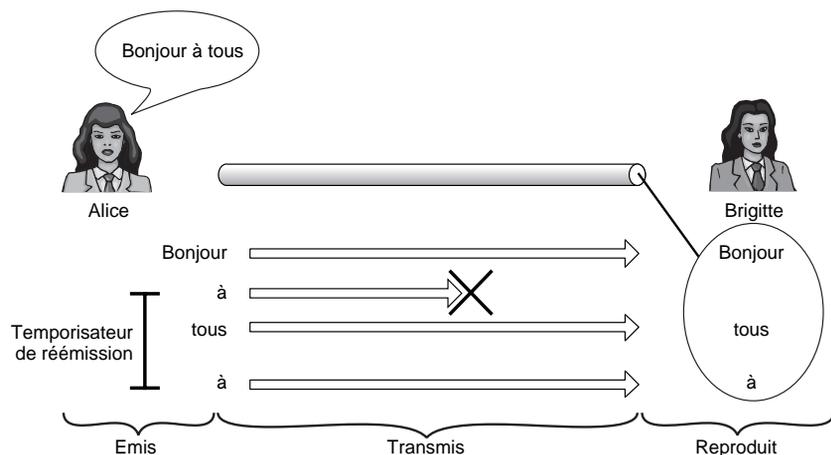
Concernant le temps, si une trame n'est pas correctement reçue par un terminal, il n'est pas forcément utile de solliciter une réémission. Le temps d'être reçue à nouveau, la trame peut devenir obsolète et sans intérêt pour le récepteur.

La figure 6.1 illustre ce qui ne devrait pas se produire. Alice envoie trois messages à Brigitte, contenant respectivement les mots « bonjour », « à », « tous » (il s'agit bien sûr d'un cas d'école, le découpage réel des trames se faisant à la durée et non au mot). Imaginons que seuls les mots « bonjour » et « tous » soient reçus par Brigitte. Avec le protocole TCP, une réémission du mot « à » va être effectuée par le terminal d'Alice, une fois le temporisateur de réémission écoulé. Comme il s'agit d'une conversation téléphonique, le terminal de Brigitte doit diffuser les messages reçus immédiatement (en fait un système de cache permet de réduire plus ou moins cet effet, mais sans pallier le problème pour autant puisqu'il n'offre qu'un délai supplémentaire limité).

Si l'application a délivré les deux mots reçus, il ne sert à rien de retransmettre le mot manquant par la suite, car celui-ci sera décorrélé de la conversation. Sa diffusion auprès du récepteur produira une perturbation plutôt qu'une amélioration. En outre, si un message est perdu, il est probable que la cause en soit la forte charge du réseau. En sollicitant une réémission de la trame, on accentue la charge et l'engorgement du réseau.

Il est donc préférable de perdre définitivement le paquet plutôt que de le réémettre.

**Figure 6.1**  
*Réémission avec TCP*



Le protocole TCP est donc bien inadapté au temps réel, puisque tous les contrôles qu'il met en place pour le transport des paquets pénalisent ses délais de transmission.

La contrainte de fiabilité n'étant pas compatible avec la contrainte de délai imposée par la ToIP, TCP n'est donc pas un bon candidat pour les transferts de type temps réel.

## UDP et le transport de données multimédias temps réel

Le protocole UDP ne comporte que des fonctionnalités de transport pur, sans aucun mécanisme de contrôle. L'adressage des données avec les ports de communication utilisés est sa seule fonction fondamentale. C'est un atout par rapport aux éléments contraignants mentionnés pour le protocole TCP. UDP est ainsi notablement plus rapide que ne l'est TCP.

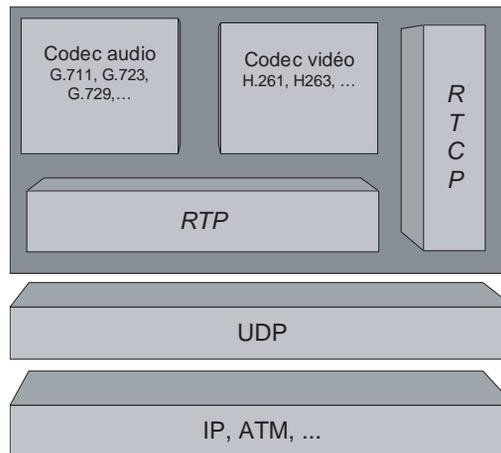
Mais la simplicité de ce modèle devient rapidement limitative. En particulier, UDP ne dispose d'aucun mécanisme lui permettant de reconstituer l'ordre des flux auprès du récepteur. Les datagrammes UDP sont totalement épurés, et aucune estampille d'horodatage, ni de numérotation n'y est insérée. Or, dans un réseau IP, les paquets peuvent emprunter des chemins différents. Avec le seul protocole UDP, la séquence temporelle originale ne peut être reconstituée au récepteur.

### En résumé

Des deux protocoles candidats au transport des données multimédias, l'un est « trop complet » et l'autre trop limité. Il est cependant possible de partir du protocole UDP et de lui ajouter des fonctionnalités d'ordonnement. Le protocole RTP a été proposé à cette seule fin de reconstitution de l'ordre du flux d'origine. Pour sa part, RTCP a été conçu pour offrir une vision de l'état du réseau et permettre à une application d'adapter les flux en conséquence.

Figure 6.2

Pile protocolaire RTP/RTCP



La figure 6.2 illustre la pile protocolaire dans laquelle s'insèrent généralement RTP et RTCP. Ces derniers étant indépendants de la couche réseau, ils peuvent fonctionner avec le protocole IP versions 4 ou 6 ou même avec le protocole ATM.

Notons que RTP et RTCP sont indépendants des couches basses et que leur utilisation conjointe avec UDP n'est qu'une application possible, même si elle demeure la plus

courante. N'importe quel autre protocole de transport, IPX par exemple, pourrait parfaitement convenir. De même, le choix du protocole de niveau réseau est laissé libre.

## Les protocoles RTP et RTCP

Comme indiqué précédemment, le couple de protocoles RTP/RTCP a été conçu dans le but d'enrichir les fonctions d'UDP et de fournir à ce dernier ce dont il a besoin pour gérer efficacement les données multimédias temps réel.

Aujourd'hui, ce couple s'utilise systématiquement dans les applications multimédias interactives, à la fois pour la téléphonie, la vidéo, les jeux vidéo et la réalité virtuelle.

### *RTP (Real-time Transport Protocol)*

RTP a été standardisé par le groupe de travail AVT-WG (Audio Video Transport-Work Group) de l'IETF.

Décrit en janvier 1996 dans la RFC 1889, rendue obsolète par la RFC 3550 en juillet 2003, il a été fortement soutenu par de nombreux constructeurs et éditeurs de logiciels, parmi lesquels Intel et Microsoft.

#### Fonctionnalités

RTP est utilisé pour le transport de bout en bout de flux ayant des contraintes temporelles fortes, typiquement pour les flux multimédias avec interactivité, tel le service de téléphonie sur IP.

Initialement, RTP était conçu pour un environnement multicast, dans lequel un émetteur diffuse son contenu vers plusieurs récepteurs en parallèle. Le cas d'un flux unicast, dans lequel un émetteur n'émet que pour un unique récepteur, n'est qu'un cas particulier et plus simple d'application multicast.

RTP assure un contrôle spécifique des données temps réel. Il permet de reconstituer les propriétés temps réel des flux médias en opérant sur deux niveaux, la synchronisation des flux d'un côté et la reconstitution de l'ordre des paquets émis et la détection des pertes de paquets de l'autre :

- Synchronisation des flux. Si l'audio et la vidéo sont transmis séparément, le destinataire doit jouer la séquence audio de façon que cette dernière coïncide avec la séquence vidéo. Pour cela, RTP ajoute aux paquets émis une estampille de date, appelée horodatage, ou timestamp. Cette estampille indique le moment où le paquet a été émis, ce qui permet de reproduire les mêmes délais interpaquet et de jouer les paquets audio et vidéo de manière synchronisée.
- Reconstitution de l'ordre des paquets émis et détection des pertes de paquets. Les paquets IP sont transmis indépendamment les uns des autres. En conséquence, leur ordre d'arrivée chez le destinataire n'est pas forcément conforme à leur ordre d'émission.

Or cet ordre est indispensable pour reconstituer le message initial et le rendre intelligible à un auditeur. En recevant plusieurs paquets, le destinataire doit savoir lequel jouer avant les autres. Pour cela, un numéro de séquence qui s'incrémente progressivement est affecté à chaque paquet. Ce numéro permet de déterminer un ordre de préséance des paquets. Par effet de bord, il permet de déterminer quels sont les paquets qui ont été perdus. Si les paquets numérotés  $i$  et  $i + 2$  sont reçus, passé un délai d'attente maximal, le terminal récepteur en déduit que le paquet numéroté  $i + 1$  est manquant.

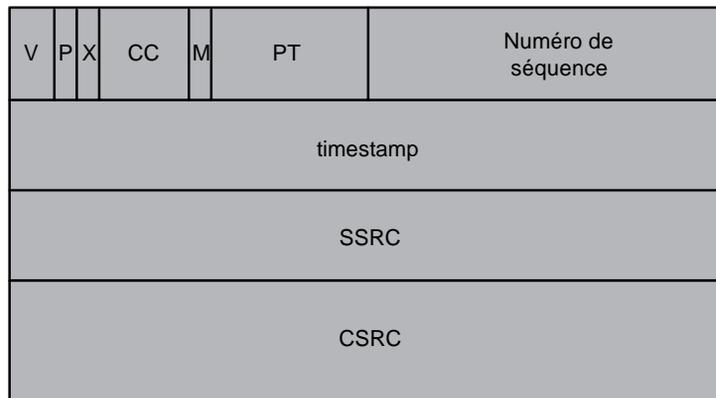
Un mécanisme de compensation de la perte de paquets est généralement mis en place au niveau applicatif. Ce mécanisme n'est pas spécifié par le protocole RTP, mais son usage est rendu possible par la détection des pertes. Par exemple, un mécanisme classique de compensation consiste à prolonger la durée d'écoute des paquets précédents et suivants (les paquets  $i$  et  $i + 2$  dans notre exemple), de façon à combler les pertes et réduire la perception humaine, tout en respectant la synchronisation des données.

### Format des paquets RTP

Le format de l'en-tête d'un paquet RTP est illustré à la figure 6.3.

**Figure 6.3**

*Format de l'en-tête RTP*



Les différents champs de l'en-tête RTP sont les suivants :

- V pour version (sur 2 bits) : indique la version du protocole RTP utilisée. Actuellement, c'est la 2 qui est exploitée.
- P pour padding (sur 1 bit) : bit indiquant si un bourrage est effectué dans les champs de données du flux multimédia.
- X pour extension (sur 1 bit) : indique si l'en-tête possède une extension d'en-tête à sa suite.
- CC pour CSRC Count (sur 4 bits) : nombre de sources ayant contribué à la génération du paquet.

- M pour marker (sur 1 bit) : indique si des descriptifs sont associés.
- PT (sur 7 bits) : décrit le format de données.
- Numéro de séquence (sur 16 bits) : compteur incrémenté d'une unité entre chaque paquet.
- Timestamp (sur 32 bits) : estampille temporelle permettant la synchronisation des flux.
- SSRC pour synchronization source (sur 32 bits) : identifie la source de la synchronisation.
- CSRC pour *contributing source* (optionnel, sur  $n$  fois 32 bits) : identifie les contributeurs à la génération du paquet.

Facultatif, le champ CSRC est utilisé lorsque plusieurs sources ont contribué à la conception d'un message. Le cas classique correspond à une conférence au cours de laquelle plusieurs personnes conversent simultanément. Si une entité se charge de rassembler les flux avant de les relayer à chaque source (comme le fait la MCU), alors cette entité est source initiale du message, tandis que les contributeurs sont toutes les personnes qui ont émis leur flux vers elle.

L'horloge utilisée pour le champ timestamp doit être partagée par la source comme par l'émetteur. Il faut donc que l'un et l'autre soient synchronisés par une référence commune, et ce dès le premier paquet échangé. Pour cela, le protocole NTP (Network Time Protocol) est généralement utilisé. Il retourne l'heure courante, sous différents formats, aux différents intervenants.

L'estampille temporelle et la numérotation des paquets comportent toutes deux des fonctionnalités d'ordonnancement, mais il n'y a pas de redondance entre ces deux paramètres. L'estampille temporelle sert à synchroniser les flux, c'est-à-dire à préciser le moment où le flux doit être joué. De la même façon, si les flux vidéo et audio sont envoyés séparément, ce qui peut se révéler pratique si tous les intervenants d'une conférence ne peuvent ou ne veulent supporter les flux vidéo mais se contentent des flux audio, l'estampille temporelle assure la concordance de la voix avec la vidéo.

En revanche, l'estampille temporelle ne permet pas de détecter les pertes. À l'inverse, la numérotation des paquets n'assure pas la synchronisation des flux mais permet de détecter les pertes. Les paquets perdus ne sont pas réémis, puisque les contraintes de temps ne le permettent généralement pas. Cependant, il est important de les détecter afin de permettre une synthèse des paquets précédents et suivants et ainsi de rendre la perte de paquets moins sensible aux interlocuteurs.

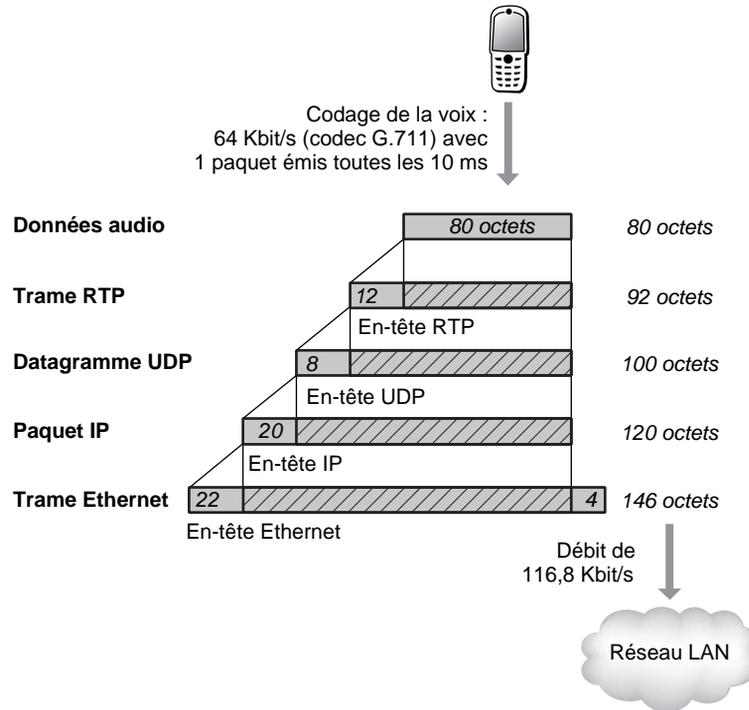
### Exemple de mise en paquet et overhead

Considérons une application audio qui utilise le codec G.711, c'est-à-dire qui transmet des données à un débit de 64 Kbit/s, et émet un paquet toutes les 10 ms.

La figure 6.4 illustre les encapsulations successives qui vont être appliquées aux données générées par le codec.

Figure 6.4

Overhead d'un paquet



En 10 ms, pour respecter le débit de 64 Kbit/s, un paquet contiendra 80 octets de données utiles. Plusieurs en-têtes successifs vont être appliqués pour encapsuler ces données :

- En-tête RTP, de 12 octets au minimum (répartis comme indiqué précédemment).
- En-tête UDP de 8 octets (2 octets pour le port source, 2 octets pour le port destination, 2 octets pour la longueur totale du datagramme et 2 octets pour le contrôle d'intégrité du datagramme).
- En-tête IP de 20 octets : 1 octet pour la version du protocole IP utilisée, 1 octet pour la longueur de l'en-tête IP, 2 octets pour le type de service permettant la gestion de la qualité de service, 2 octets pour la longueur totale du paquet, incluant l'en-tête IP, 16 bits pour l'identification d'un paquet IP, qui sont utilisés, lorsque le paquet IP est fragmenté, pour reconstituer les parties, 3 bits de drapeau, 13 bits indiquant la position d'un fragment pour recomposer les parties dans l'ordre, 1 octet pour le champ TTL, 1 octet pour coder le protocole encapsulé, 2 octets pour contrôler l'intégrité du paquet, 4 octets pour l'adresse IP source, 4 octets pour l'adresse IP destination. On ne compte pas les options de l'en-tête IP, qui ajouteraient 4 octets supplémentaires.
- En-tête Ethernet, de 26 octets : 8 octets de préambule pour la synchronisation des trames, 6 octets pour l'adresse MAC du terminal de destination, 6 octets pour l'adresse MAC du terminal source, 2 octets pour le type de protocole et 4 octets pour le CRC (contrôle de redondance cyclique), concaténés à la suite du paquet IP original.

On ne compte pas les éventuels octets de bourrage, et on utilise une version courante du protocole Ethernet, bien qu'il en existe d'autres.

Au final, on arrive à 146 octets transmis dans un réseau LAN de type Ethernet toutes les 10 ms. Cela équivaut à un débit effectif réel de 116,8 Kbit/s. Si l'on compare ce résultat avec les données réellement utiles transmises à un débit de 64 Kbit/s, on constate que le débit a presque été doublé (il est 1,8 fois plus important).

Dans les applications de téléphonie, les paquets sont très petits et sont donc très fortement surchargés par les en-têtes, pouvant être 10 fois plus importantes avec certains codecs. Cette surcharge est appelée overhead. Elle comprend les données qui ne sont pas utiles, c'est-à-dire les données de transport non multimédias. Avec la vidéo, les paquets sont plus importants, si bien que la surcharge devient globalement négligeable.

### Extensions et limitations

Deux entités, le multiplexeur et le convertisseur, sont couramment utilisées avec RTP pour faciliter le transport des flux multimédias et les adapter :

- Le multiplexeur permet d'adapter les flux aux utilisateurs. Le multiplexeur (en anglais *mixer*) est un équipement chargé de multiplexer, resynchroniser et gérer les débits entre différents utilisateurs. Typiquement, lors d'une conférence, si un intervenant dispose d'une forte bande passante tandis qu'un autre ne dispose que d'une faible bande passante, le multiplexeur reçoit les flux et fournit au premier un flux de haute qualité et au second un flux de moins bonne qualité. Le multiplexeur n'est pas un équipement transparent. Il modifie les en-têtes RTP, en particulier le champ SSRC indiquant l'adresse de l'équipement source ayant produit le flux.
- Le convertisseur permet de modifier les codages et de traverser les pare-feu. Le convertisseur (en anglais *translator*) adapte la syntaxe des flux entre les liens. Par exemple, si, entre deux réseaux, les débits sont différents, le convertisseur modifie le codec à la volée. Contrairement au multiplexeur, le convertisseur est un équipement transparent pour les utilisateurs, qui ne modifie pas les champs RTP.

Une application classique du convertisseur concerne la sécurité. Considérons une conférence audio faisant intervenir plusieurs correspondants en multicast. Au sein d'une entreprise, il est classique que les flux multicast soient prohibés en ce qu'ils sont potentiellement nuisibles aux réseaux et peuvent provoquer des attaques. Le convertisseur permet de contourner ce problème. Pour cela, on met en place un traducteur devant et derrière le pare-feu, c'est-à-dire en entrée et en sortie du réseau.

Le convertisseur situé devant le pare-feu réalise l'opération suivante : l'adresse multicast avec laquelle l'émetteur envoie un paquet va être transformée en l'adresse du convertisseur situé derrière le pare-feu, et l'adresse multicast originale va être sauvegardée et masquée à l'intérieur du paquet. Ce flux deviendra ainsi licite puisqu'il ne sera plus multicast, mais seulement à destination du convertisseur situé derrière le pare-feu. Il pourra donc traverser le pare-feu.

Parallèlement, le convertisseur situé derrière le pare-feu réalise l'opération inverse, en recevant les paquets. L'adresse unicast à laquelle le paquet est reçu est remplacée par l'adresse multicast originale, qui se trouve à l'intérieur du paquet. Le paquet est donc envoyé vers ses destinataires véritables. Le convertisseur permet de la sorte de contourner le pare-feu de l'entreprise en remplaçant les flux unicast en flux multicast et réciproquement.

## **RTCP (Real-time Transport Control Protocol)**

Décrit dans la RFC 3550, RTCP est un protocole de contrôle et de supervision du réseau. Il opère comme une sonde qui rend compte aux émetteurs des performances dont la communication en cours bénéficie. Son objectif est d'offrir aux participants d'une session une vision sur l'état du réseau et de s'y adapter de façon dynamique. Il fournit pour cela un rapport sur la qualité de distribution, incluant le délai de bout en bout, la gigue et le taux de pertes. Ce rapport est envoyé de façon périodique de façon que les intervenants disposent d'une mise à jour fréquente de l'état du réseau.

Par exemple, si un utilisateur fait de la téléphonie et que, par le biais du protocole RTCP, son correspondant lui envoie des rapports signalant un fort taux de perte de paquets, il peut considérer que le codec qu'il utilise est trop gourmand pour être supporté dans les conditions actuelles du réseau. Il peut dès lors sélectionner un codec de qualité un peu moins bonne mais nécessitant moins de ressources.

Dans sa spécification, RTCP n'est aucunement indispensable pour le fonctionnement de RTP. La réciproque est vraie également. Néanmoins, leur association apporte une cohérence globale dans le traitement des communications multimédias. Tous deux doivent être pensés et intégrés au niveau applicatif. Les rapports fournis par RTCP peuvent optimiser la qualité de la transmission.

### **Les catégories de paquets RTCP**

Les paquets RTCP sont classés en cinq catégories :

- SR (Sender Report). Ce type de paquet véhicule un rapport de l'émetteur, sous forme d'un ensemble de statistiques relatives à la qualité de service concernant l'émetteur. On trouve parmi ces informations le nombre de paquets perdus et la gigue mesurée par l'émetteur. La gigue est importante, car elle permet d'apprécier la régularité de l'arrivée des paquets transportant de la parole. On repère ces paquets SR par la valeur du champ PT (Payload Type), qui est mis à la valeur 201.
- RR (Receiver Report). Ce type de paquet véhicule un rapport de récepteur, semblable aux paquets SR mais concernant le récepteur. La valeur du champ PT est 202.
- SDES (Source Description). Ce type de paquet décrit une source, avec un ensemble de paramètres spécifiques parmi lesquels le nom permanent de la source, ou CNAME (Canonical Name), le nom du participant, NAME, son adresse e-mail, EMAIL, son numéro de téléphone (PHONE), sa localisation (LOC), le nom de l'application qu'il utilise, avec si possible sa version (TOOL), et d'autres paramètres spéciaux (PRIV et

NOTE pour ajouter des informations complémentaires). Ce type de paquet porte la valeur 203 dans le champ PT (Payload Type).

- BYE. Ce type de paquet est envoyé pour indiquer que l'émetteur quitte une session multimédia. Le champ PT (Payload Type) prend la valeur 204.
- APP (Application). Ce type de paquet est réservé pour transporter des paramètres spécifiques d'une application. Ce type de paquet est indiqué par la valeur 205 du champ PT (Payload Type).

La fréquence des échanges de transmission des rapports doit être dosée en fonction du type de média transporté et du nombre de participants. Plus le nombre d'utilisateurs, et donc de rapports échangés, est important, plus les rapports risquent de perturber les communications en réclamant une quantité de bande passante importante. En règle générale, on considère qu'un overhead de 5 % par rapport aux données réelles est une borne maximale.

### **RTP/RTCP et la qualité de service**

Les protocoles RTP et RTCP ne sont pas proposés pour garantir une qualité de service. Ils ne supportent pas notamment les services suivants :

- Réserve de ressources dans le réseau. Si, à un instant donné, le débit permet de faire de la vidéo, le protocole RTP ne garantit en rien qu'à l'instant suivant ce sera toujours possible.
- Fiabilisation des échanges. Une perte de paquets est équivalente au niveau du récepteur à un silence. Le terminal du récepteur peut combler ces silences en allongeant la durée des messages vocaux précédents et suivants de façon à masquer les manques et les rendre moins perceptibles à l'oreille.
- Garantie des délais de transit dans le réseau. Les paquets peuvent être retardés dans leur cheminement de bout en bout, et RTP ne garantit pas qu'ils seront reçus au bon moment. La variation du délai de transit de bout en bout des paquets est perçue au niveau du récepteur par une voix saccadée. Pour réduire cet effet, l'utilisation de tampons (*buffers*) est nécessaire afin de stabiliser la variation dans l'arrivée des paquets.

### **Les contrôles au niveau réseau**

L'IETF propose deux grandes catégories de services pour les contrôles réseau. Ils se déclinent en sous-services dotés de différentes qualités de service : les services intégrés IntServ (Integrated Services) et les services différenciés DiffServ (Differentiated Services).

Les services intégrés sont gérés indépendamment les uns des autres, tandis que les services différenciés rassemblent plusieurs applications simultanément. La première solution

est souvent choisie pour le réseau d'accès et la seconde pour l'intérieur du réseau, lorsqu'il y a beaucoup de flots à gérer.

Les services IntServ disposent des trois classes suivantes :

- service garanti (Guaranteed Service) ;
- service contrôlé (Controlled Load) ;
- best-effort.

Les services DiffServ disposent des trois classes suivantes :

- service garanti (Expedited Forwarding), ou service Premium ;
- service contrôlé (Assured Forwarding), ou service Olympic ;
- best-effort.

### ***IntServ (Integrated Services)***

Le service IntServ intègre deux niveaux de service avec garantie de performance. C'est un service orienté flot, dans lequel chaque flot peut faire sa demande spécifique de qualité de service.

Pour obtenir une garantie précise, le groupe de travail IntServ a considéré que seule une réservation de ressources était capable de fournir à coup sûr les moyens de garantir la demande.

Comme expliqué précédemment, trois sous-types de services sont définis dans IntServ : un service avec garantie totale, un service avec garantie partielle et le service best-effort. Le premier correspond aux services rigides, avec de fortes contraintes à respecter. Les deuxième et troisième sont des services dits élastiques, qui n'ont pas de contraintes fortes.

Lorsqu'ils reçoivent une demande *via* le protocole RSVP, les routeurs peuvent l'accepter ou la refuser. Cette demande s'effectue du récepteur vers l'émetteur après une phase aller. Une fois la demande acceptée, les routeurs placent les paquets correspondants dans une file d'attente de la classe de service demandée.

Le service IntServ doit posséder les mécanismes suivants :

- Procédure de signalisation pour avertir les nœuds traversés. Le protocole RSVP est supposé remplir cette tâche.
- Méthode permettant d'indiquer la demande de qualité de service de l'utilisateur dans le paquet IP afin que les nœuds en tiennent compte.
- Contrôle de trafic pour maintenir la qualité de service.
- Mécanisme permettant de faire passer le niveau de qualité au réseau sous-jacent, s'il existe.

Le service garanti GS (Guaranteed Service) affecte une borne supérieure au délai d'acheminement. Pour cela, un protocole de réservation tel que RSVP est nécessaire. La demande de réservation comporte deux parties : une spécification de la qualité de service déterminée par un FlowSpec et une spécification des paquets qui doivent être pris en compte par un filtre, le FilterSpec. En d'autres termes, certains paquets du flot peuvent avoir une qualité de service mais pas forcément les autres. Chaque flot possède sa qualité de service et son filtre, qui peut être fixe (*fixed filter*), partagé avec d'autres sources (*shared-explicit*) ou encore spécifique (*wildcard filter*).

Le service partiellement garanti CL (Controlled Load) doit garantir une qualité de service à peu près égale à celle offerte par un réseau peu chargé. Cette classe est essentiellement utilisée pour le transport des services élastiques. Les temps de transit dans le réseau des flots CL sont similaires à ceux de clients d'une classe best-effort dans un réseau très peu chargé. Pour arriver à cette fluidité du réseau, il faut intégrer une technique de contrôle.

Les deux services doivent pouvoir être réclamés par l'application *via* une interface située entre l'application et le protocole de mise en place du service IntServ. Deux possibilités sont proposées dans IntServ : l'utilisation de la spécification GQoS Winsock2, qui permet le transport d'applications point-à-point et multipoint, et RAPI (RSVP API), qui est une interface applicative sous UNIX.

L'ordonnancement des paquets dans les routeurs est un deuxième mécanisme nécessaire. Un de ceux les plus classiquement proposés est le WFQ (Weighted Fair Queuing). Cet algorithme placé dans chaque routeur demande une mise en file d'attente des paquets suivant leur priorité. Les files d'attente sont servies dans un ordre déterminé dépendant de l'opérateur. Généralement, le nombre de paquets servis à chaque passage dans le serveur dépend du paramètre de poids de la file d'attente.

Il existe de nombreuses solutions pour gérer la façon dont le service est affecté aux files d'attente, généralement fondées sur des niveaux de priorité. Citons notamment l'algorithme Virtual Clock, qui utilise une horloge virtuelle pour déterminer les temps d'émission, et SCFQ (Self-Clocked Fair Queuing), qui travaille sur des intervalles de temps minimaux entre deux émissions de paquets d'une même classe, intervalle dépendant de la priorité.

Le service IntServ pose le problème du passage à l'échelle, ou scalabilité, qui désigne la possibilité de continuer à bien se comporter lorsque le nombre de flots à gérer devient très grand, comme c'est le cas sur Internet. Le contrôle IntServ s'effectuant sur la base de flots individuels, les routeurs du réseau IntServ doivent garder en mémoire les caractéristiques de chaque flot. Une autre difficulté concerne le traitement des différents flots dans les nœuds IntServ : quel flot traiter avant tel autre lorsque des milliers de flots arrivent simultanément avec des classes et des paramètres associés distincts ?

En l'absence de solution reconnue à tous ces problèmes, la seconde grande technique de contrôle, DiffServ, essaie de trier les flots dans un petit nombre bien défini de classes, en multiplexant les flots de même nature dans des flots plus importants, mais toujours en

nombre limité. IntServ peut cependant s'appliquer à de petits réseaux comme les réseaux d'accès.

D'autres recherches vers des processeurs de gestion spécialisés dans la qualité de service ont débouché récemment sur des équipements capables de traiter plusieurs dizaines, voire centaines de milliers de flots.

Le groupe de travail ISSLL (Integrated Services Over Specific Link Layers) de l'IETF cherche à définir un modèle IntServ agissant sur un niveau trame de type ATM, Ethernet, relais de trames, PPP, etc. L'objectif est de proposer des mécanismes permettant de faire passer le niveau de priorité de la classe vers des classes parfois non équivalentes et de choisir dans le réseau sous-jacent des algorithmes susceptibles de donner un résultat équivalent à celui qui serait obtenu dans le monde IP.

### ***DiffServ (Differentiated Services)***

Le principal objectif de DiffServ est de proposer un schéma général permettant de déployer de la qualité de service dans un grand réseau IP et de réaliser ce déploiement assez rapidement.

DiffServ sépare l'architecture en deux composantes majeures : la technique de transfert et la configuration des paramètres utilisés lors du transfert. Cela concerne aussi bien le traitement reçu par les paquets lors de leur transfert dans un nœud que la gestion des files d'attente et la discipline de service, c'est-à-dire la façon dont les clients sont servis ; les disciplines de service les plus connues étant FIFO (First In First Out), LCFS (Last Come, First Served) et PS (Processor Sharing). La configuration de tous les nœuds du chemin s'effectue selon un critère appelé PHB (Per-Hop Behavior). Ces PHB déterminent les différents traitements correspondant aux flots qui ont été différenciés dans le réseau.

DiffServ définit la sémantique générale des PHB, et non les mécanismes spécifiques permettant de les implémenter. Les PHB sont définis une fois pour toutes, tandis que les mécanismes peuvent être modifiés et améliorés, voire être différents suivant le type de réseau sous-jacent.

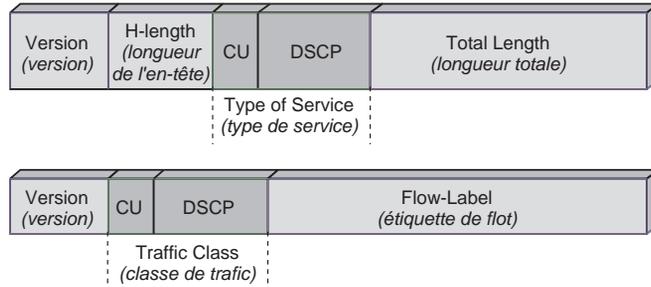
Les PHB et les mécanismes associés doivent pouvoir être facilement déployés dans les réseaux IP, ce qui demande que chaque nœud puisse gérer les flots grâce à des mécanismes tels que l'ordonnancement, la mise en forme ou la perte des paquets traversant un nœud.

DiffServ agrège les flots en classes, appelées agrégats, qui offrent des qualités de service spécifiques. La qualité de service est assurée par des traitements effectués dans les routeurs spécifiés par un indicateur situé dans le paquet IP. Les points d'agrégation des trafics entrants sont généralement placés à l'entrée du réseau. Les routeurs sont configurés grâce au champ DSCP (Differentiated Service Control Point) du paquet IP, qui forme la première partie d'un champ plus général appelé DS (Differentiated Service) et contenant aussi un champ CU (Currently Unused). Dans IPv4, ce champ DS est pris sur la zone ToS (Type of Service), qui est de ce fait redéfinie par rapport à sa première utilisation. Dans IPv6, ce champ se situe dans la zone TC (Traffic Class) de la classe de service.

La figure 6.5 illustre le champ DS des paquets IPv4 et IPv6. Le champ DSCP prend place dans le champ TOS (Type of Service) d'IPv4 et dans le champ Traffic Class d'IPv6. Le champ DSCP tient sur 6 des 8 bits et est complété par deux bits CU. Le DSCP détermine la classe de service PHB (Per-Hop Behavior).

**Figure 6.5**

*Le champ DS des paquets IPv4 et IPv6*



Le champ de 6 bits du DSCP est interprété par le nœud afin d'attribuer au paquet le traitement correspondant à la classe PHB indiquée. Les deux bits CU sont ignorés lors du traitement dans un nœud DiffServ normalisé. Par l'intermédiaire d'une table, les valeurs du DSCP déterminent les PHB acceptables par le nœud. Une valeur par défaut doit toujours être indiquée lorsque le champ DSCP ne correspond à aucun PHB.

Les opérateurs de télécommunications peuvent définir leurs propres valeurs de DSCP pour un PHB donné, à la place de la valeur recommandée par la standardisation de l'IETF. Ces opérateurs doivent toutefois fournir dans les passerelles de sortie la valeur standard du DSCP de façon que ce champ soit interprété convenablement par l'opérateur suivant. En particulier, un DSCP non reconnu doit toujours être interprété par une valeur par défaut.

La définition de la structure du champ DS est incompatible avec celle du champ ToS de la RFC 791 définissant IPv4. Ce champ TOS avait été conçu pour indiquer les critères à privilégier dans le routage. Parmi les critères prévus se trouvent le délai, la fiabilité, le coût et la sécurité.

Outre le service BE (best-effort), deux PHB assez semblables à ceux d'IntServ sont définis dans DiffServ :

- EF (Expedited Forwarding), ou service garanti, que l'on appelle aussi service Premium.
- AF (Assured Forwarding), ou service assuré, que l'on appelle aussi Assured Service ou Olympic.

Il existe quatre sous-classes de services en AF déterminant des taux de perte acceptables pour les flots de paquets considérés. Elles sont classées en Platinum (platine), Gold (or), Silver (argent) et Bronze (bronze). Comme cette terminologie n'est pas normalisée, il est possible d'en rencontrer d'autres.

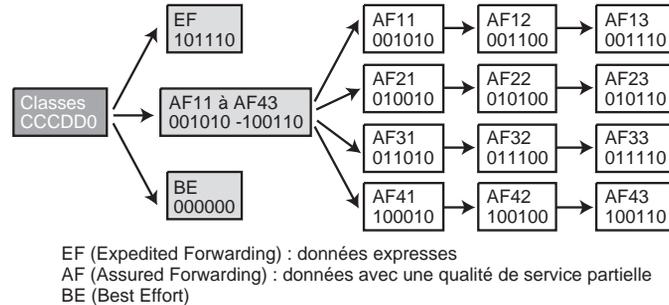
À l'intérieur de chacune de ces classes, trois sous-classes triées selon leur degré de précedence, c'est-à-dire leur niveau de priorité les uns par rapport aux autres, sont définies. La classe AF1x est la plus prioritaire, puis vient la classe AF2x, etc. Il existe donc au total

douze classes standardisées, mais peu d'opérateurs les mettent en œuvre. En règle générale, ces derniers se satisfont des trois classes de base du service AF, et donc de cinq classes au total en ajoutant les services EF et BE.

Les valeurs portées par le champ DSCP associé à ces différentes classes sont illustrées à la figure 6.6. Par exemple, la valeur 101110 du champ DSCP indique que le paquet est de type EF (Expedited Forwarding). La classe best-effort porte la valeur 000000.

**Figure 6.6**

*Classes de service DiffServ et valeurs des champs DSCP associés*



Le DSCP 11x000 est réservé à des classes de clients encore plus prioritaires que EF. Il peut, par exemple, être utilisé pour des paquets de signalisation.

### EF (Expedited Forwarding)

Le PHB EF (Expedited Forwarding) est défini comme un transfert de paquets pour une agrégation de flots provenant de nœuds DiffServ dans laquelle le taux de service des paquets de l'agrégat est supérieur à un taux déterminé par l'opérateur. Le trafic EF doit pouvoir recevoir un taux de service indépendamment des autres trafics circulant dans le réseau. En terme encore plus précis, le taux du trafic EF doit être supérieur ou égal au taux déterminé par l'opérateur mesuré sur n'importe quel intervalle de temps au moins égal à la taille d'une MTU (Mean Transmission Unit). Si le PHB EF est implémenté grâce à un mécanisme de priorité sur les autres trafics, il faut que le taux de trafic de l'agrégat EF ne dépasse pas une certaine limite, qui deviendrait inacceptable pour les PHB des autres classes de trafic.

Plusieurs types de mécanismes d'ordonnancement peuvent être utilisés pour répondre à ces contraintes. Une file prioritaire est le mécanisme le plus simple pour réaliser le service (le PHB) EF tant qu'il n'y a pas d'autres files d'attente plus prioritaires. Il est possible d'utiliser une file d'attente normale dans un groupe de files d'attente gérées par un mécanisme de tour de rôle avec poids (Weighted Round Robin) ou d'utiliser un partage de la bande passante de la file de sortie du nœud, permettant à la file EF d'atteindre le taux de service garanti par l'opérateur. Un autre mécanisme potentiel, appelé partage CBQ (Class-based Queueing), donne à la file EF une priorité suffisante pour obtenir au moins le taux de service garanti par l'opérateur.

Le trafic Expedited Forwarding correspond au trafic sensible au délai et à la gigue. Il est doté d'une priorité forte dans les nœuds mais doit être contrôlé afin que la somme des trafics provenant des différentes sources et passant sur une même liaison ne dépasse pas la capacité nominale déterminée par l'opérateur.

Plusieurs solutions permettent de réserver la bande passante proposée aux flots de paquets EF. Un protocole de type RSVP, par exemple, peut effectuer les réservations de bande passante nécessaires. Une autre solution consiste à utiliser un serveur spécialisé dans la distribution de la bande passante, ou Bandwidth Broker. Ce serveur de bande passante réalise le contrôle d'admission en proposant une réservation centralisée.

### AF (Assured Forwarding)

Les PHB AF (Assured Forwarding) assurent le transfert de paquets IP pour lesquels une certaine qualité de service peut être garantie. Les trafics AF sont subdivisés en  $n$  classes AF distinctes. Dans chaque classe un paquet IP se voit assigner un taux de perte maximal et une priorité à la perte, correspondant à des classes de précedence. Un paquet IP qui appartient à la classe AF $i$  et qui possède un taux de perte correspondant à la précedence  $j$  est marqué par un DSCP AF $ij$ .

Comme expliqué précédemment, douze classes sont définies pour DiffServ, correspondant à quatre classes AF avec des garanties sur la perte de paquets. Ces quatre classes correspondant aux taux de perte garantie sont appelées Platinum, Or, Argent et Bronze, chaque classe ayant trois niveaux de précedence différents.

Les paquets d'une classe AF sont transférés indépendamment de ceux des autres classes AF. En d'autres termes, un nœud ne peut pas agréger de flots ayant des DSCP différents dans une classe commune.

Un nœud DiffServ doit allouer un ensemble de ressources minimales à chaque PHB AF afin que ceux-ci puissent remplir le service pour lequel ils ont été mis en place. Une classe AF doit posséder des ressources minimales en mémoire et en bande passante pour qu'un taux de service minimal, déterminé par l'opérateur, puisse être réalisé sur une échelle de temps potentiellement assez longue. En d'autres termes, sur un intervalle de temps relativement long, pouvant se compter en seconde, une garantie de débit doit être procurée aux services AF.

Un nœud AF doit pouvoir être configuré pour permettre à une classe AF de recevoir plus de ressources de transfert que le minimum quand des ressources supplémentaires sont disponibles dans le réseau. Cette allocation supplémentaire n'est pas forcément proportionnelle au niveau de la classe, mais l'opérateur doit être capable de réallouer les ressources libérées par la classe EF sur les PHB AF. Les précedences doivent toutefois être respectées, une classe de meilleure précedence ne devant pas perdre plus de paquets qu'une classe avec une précedence inférieure, même si la perte reste en dessous du niveau admissible.

Un domaine implémentant des services AF doit, par l'intermédiaire des routeurs frontière, être capable de contrôler les entrées de trafics AF pour que les qualités de service

déterminées pour chaque classe AF soient satisfaites. Les routeurs frontière doivent pour cela mettre en place des mécanismes de mise en forme du trafic (*shaper*), de destruction de paquets (*dropper*), d'augmentation ou de diminution des pertes de paquets par classe AF et de réassignation de trafics AF dans d'autres classes AF. Les actions d'ordonnancement ne doivent pas causer de remise en ordre des paquets d'un même microflot, un microflot étant un flot particulier à l'intérieur d'un agrégat d'une classe de PHB.

L'implémentation d'une stratégie AF doit minimiser le taux de congestion à l'intérieur de chaque classe, même si des congestions de courte durée sont admissibles suite à des superpositions de flots continus de paquets (*bursts*). Cela demande un algorithme de gestion dynamique dans chaque nœud AF. Un exemple d'un tel algorithme est RED (Random Early Drop). La congestion à long terme doit aussi être évitée grâce à des pertes de paquets correspondant aux niveaux de précedence, et celle à court terme grâce à des files d'attente permettant de mettre en attente certains paquets. Les algorithmes de mise en forme du trafic (*shaper*) doivent par ailleurs être capables de détecter les paquets susceptibles d'engendrer des congestions à long terme.

L'algorithme de base permettant d'effectuer le contrôle des trafics AF est WRED, ou Weighted RED. Il consiste à essayer de maintenir le réseau dans un état fluide. Les pertes de paquets doivent être proportionnelles à la longueur des files d'attente. En d'autres termes, les paquets en surplus puis les paquets normaux sont éliminés dès que le trafic n'est plus fluide. Le temps écoulé depuis la dernière perte sur un même agrégat est pris en compte dans l'algorithme. La procédure essaie de distribuer le contrôle à l'ensemble des nœuds et non plus au seul nœud congestionné. Les algorithmes de destruction de paquets doivent être indépendants du court terme et des microflots, ainsi que des microflots à l'intérieur des agrégats.

L'interconnexion de services AF peut être assez difficile à réaliser du fait de la relative imprécision des niveaux de service des différents opérateurs s'interconnectant.

Une solution pour permettre la traversée d'un agrégat dans un réseau IP non conforme à DiffServ consiste à réaliser un tunnel avec une qualité de service supérieure à celle du PHB. Lorsqu'un agrégat de paquets AF utilise le tunnel, la qualité de service assurée par ce dernier doit permettre au PHB de base d'être respecté à la sortie du tunnel.

Un client qui demande un trafic Assured Forwarding doit négocier un agrément de service, appelé SLA (Service Level Agreement), correspondant à un profil déterminé par un ensemble de paramètres de qualité de service, nommé SLS (Service Level Specification). Le SLS indique un taux de perte et, pour les services EF, un temps de réponse moyen et une gigue du temps de réponse. Le trafic n'entrant pas dans le profil est détruit en priorité si un risque de congestion existe qui ne permettrait pas au trafic conforme d'atteindre sa qualité de service.

### Architecture d'un nœud DiffServ

L'architecture d'un nœud DiffServ est illustrée à la figure 6.7. Elle comprend une entrée contenant un classificateur (Classifier), dont le rôle est de déterminer le bon chemin à l'intérieur du nœud. L'embranchement choisi dépend de la classe détectée par le classificateur.

Viennent ensuite des organes appelés Meter, ou métreurs. Un métreur détermine si le paquet a les performances requises par sa classe et décide de la suite du traitement. Le métreur connaît l'ensemble des files d'attente du nœud ainsi que les paramètres de qualité de service demandés par l'agrégat auquel appartient le paquet. Il peut décider de la destruction éventuelle d'un paquet, si sa classe le permet, ou de son envoi vers une file d'attente de sortie. Le nœud DiffServ peut aussi décider de changer ce paquet de classe ou de le multiplexer avec d'autres flots, comme nous le verrons. L'organe Dropper, ou supprimeur, peut décider de perdre ou non, c'est-à-dire de détruire ou non le paquet, tandis que le supprimeur absolu (Absolute Dropper) élimine automatiquement le paquet.

En d'autres termes, le métreur (Meter) peut prendre une décision de destruction et envoyer le paquet dans un supprimeur absolu (Absolute Dropper), alors que le métreur (Meter) ne fait que déterminer les paramètres de performance et laisse au supprimeur (Dropper) le soin de détruire ou non le paquet suivant d'autres critères que la mesure brute de la performance.

Pour certains paquets, comme les paquets BE (best-effort), il n'est pas nécessaire de se poser la question de la performance puisqu'il n'y a aucune garantie sur l'agrégat. Il suffit de savoir si le paquet doit être perdu ou non. Cela correspond à la branche X sur la figure 6.7 Sur cette même figure, la première branche (A) correspond aux clients EF ou Premium, les deux suivantes (B et C) à des clients AF, avec des clients Platinum ou Gold dans le chemin haut et Silver ou Bronze dans l'autre chemin.

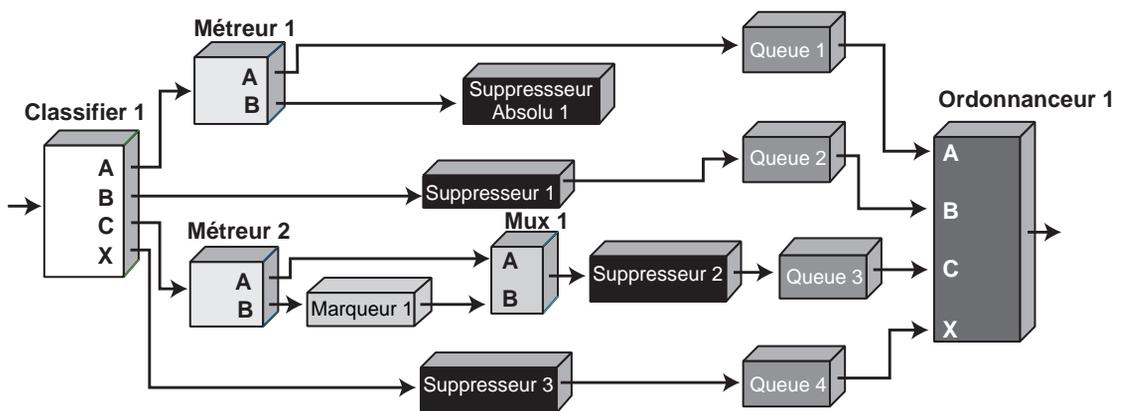


Figure 6.7

Architecture d'un nœud DiffServ

L'architecture d'un nœud DiffServ se termine par des files d'attente destinées à mettre en attente les paquets avant leur émission sur la ligne de sortie déterminée par le routage. Un algorithme de précedence est utilisé pour traiter l'ordre d'émission des paquets. L'ordonnanceur (Scheduler) s'occupe de cette fonction. L'algorithme le plus simple revient à traiter les files suivant leur ordre de priorité et à ne pas laisser passer les clients d'une autre file tant qu'il y a encore des clients dans une file prioritaire.

De nombreux autres algorithmes permettent de donner un poids spécifique aux files d'attente, de telle façon qu'un client non prioritaire puisse être servi avant un client prioritaire. Parmi ces algorithmes, nous avons déjà évoqué WFQ (Weighted Fair Queueing), dans lequel chaque file d'attente comporte un poids, par exemple 70 pour la file EF, 20 pour la file AF Gold et 10 pour l'autre file AF. L'ordonnanceur laisse passer pendant 70 % du temps les clients EF. Si ces clients dépassent l'utilisation de 70 %, l'ordonnanceur accepte de laisser passer des clients AF Gold pendant les 20 % du temps restant et pendant 10 % des clients AF Silver ou Bronze.

L'ensemble des actions subies par un paquet dans un nœud DiffServ est réalisé par un organe général appelé conditionneur (Conditioner). Un conditionneur de trafic peut contenir les éléments suivants : mètreur (Meter), marqueur (Marker), metteur en forme (Shaper) et supprimeur de paquets (Dropper). Un flot est sélectionné par le classificateur. Un mètreur est utilisé pour mesurer le trafic en comparaison du profil. La mesure effectuée par le mètreur pour un paquet peut être utilisée pour déterminer s'il faut envoyer le paquet vers un marqueur ou un supprimeur de paquet.

Lorsque le paquet sort du conditionneur, il doit posséder la valeur appropriée du DSCP. Le mètreur obtient les propriétés temporelles du flot de paquets sélectionnés par le classificateur en fonction d'un profil déterminé par un TCA (Traffic Conditioning Agreement). Le mètreur envoie ces informations aux autres organes du conditionneur, lesquels mettent en œuvre des fonctions spécifiques adaptées aux paquets afin que ceux-ci reçoivent les traitements appropriés, qu'ils se trouvent dans le profil ou hors profil.

Les marqueurs de paquets positionnent le champ DSCP à une valeur particulière et ajoutent le paquet au flux agrégé correspondant. Le marqueur peut être configuré pour marquer tous les paquets à la bonne valeur du DSCP ou pour choisir un DSCP particulier pour un ensemble de PHB prédéterminé.

Les metteurs en forme (Shaper) ont pour objectif de retarder des paquets d'un même flot pour les mettre en conformité avec un profil déterminé. Un metteur en forme possède généralement une mémoire de taille finie permettant de retarder les paquets en les mettant en attente. Ceux-ci peuvent être détruits s'il n'y a pas de place disponible en mémoire pour les mettre en conformité.

Les supprimeurs détruisent les paquets d'un même flot qui ne sont pas conformes au profil de trafic. Ce processus est parfois appelé « policing » de trafic. Un supprimeur est parfois implémenté dans le metteur en forme lorsqu'un paquet doit être rejeté, s'il est impossible de le remettre dans le profil.

Les conditionneurs de trafic sont le plus souvent placés dans les nœuds d'entrée et de sortie des domaines DS. Puisque le marquage des paquets est effectué par les nœuds d'entrée du domaine, un agrégat provenant d'un autre opérateur est supposé être conforme au TCA approprié.

## L'ingénierie de trafic

Les méthodes de contrôle de la qualité de service que nous avons vues sont essentiellement statistiques. On joue soit sur le débit applicatif, en espérant que l'ensemble des débits pourra être supporté par le réseau, soit sur des classes de service, afin de limiter le débit de la classe la plus haute, en espérant là encore que le débit total des flux les plus prioritaires restera suffisamment faible pour que le réseau soit fluide.

Ces solutions sont surtout envisageables dans des environnements que le gestionnaire du réseau peut maîtriser parce qu'il en connaît les clients. Pour les réseaux d'opérateurs, ces solutions ne sont guère acceptables étant donné la méconnaissance des flux qui peuvent potentiellement transiter à tout instant. C'est la raison pour laquelle les opérateurs et les très grandes entreprises se dirigent vers une autre solution, appelée ingénierie de trafic, pour assurer le contrôle de la qualité de service.

Pour effectuer de l'ingénierie de trafic, il faut savoir par où passent les flux et connaître leurs caractéristiques. La connaissance des flux de téléphonie est particulièrement simple, puisqu'elle dépend principalement des codecs et de leurs paramètres. Si l'on sait par où passent les flux et si l'on connaît leurs caractéristiques, l'ingénierie de trafic est une bonne solution. Le principal réseau qui effectue ce type de contrôle est MPLS et son extension GMPLS.

Les caractéristiques les plus importantes de la norme MPLS (MultiProtocol Label-Switching) sont les suivantes :

- Spécification des mécanismes pour transporter des flots de paquets IP avec diverses granularités des flots entre un routeur, appelé LSR (Label Switched Router), d'entrée et un LSR de sortie. La granularité désigne la grosseur du flot, lequel peut intégrer plus ou moins de flots utilisateur.
- Indépendance du niveau trame et du niveau paquet. En fait, seul le transport de paquets IP est pris en compte.
- Mise en relation de l'adresse IP du destinataire avec une référence d'entrée dans le réseau.
- Reconnaissance par les routeurs de bord des protocoles de routage, de type OSPF (Open Shortest Path First), et de signalisation, tels LDP (Label Distribution Protocol) ou RSVP.
- Utilisation de différents types de trames.

Quelques propriétés supplémentaires méritent d'être soulignées :

- L'ouverture du circuit virtuel est fondée sur la topologie, bien que d'autres possibilités soient également définies dans la norme.

- L'assignation des références est effectuée par l'aval, c'est-à-dire à la demande d'un nœud émettant un message en direction de l'émetteur.
- La granularité des références est variable.
- Le stock des références est géré selon la méthode « dernier arrivé premier servi ».

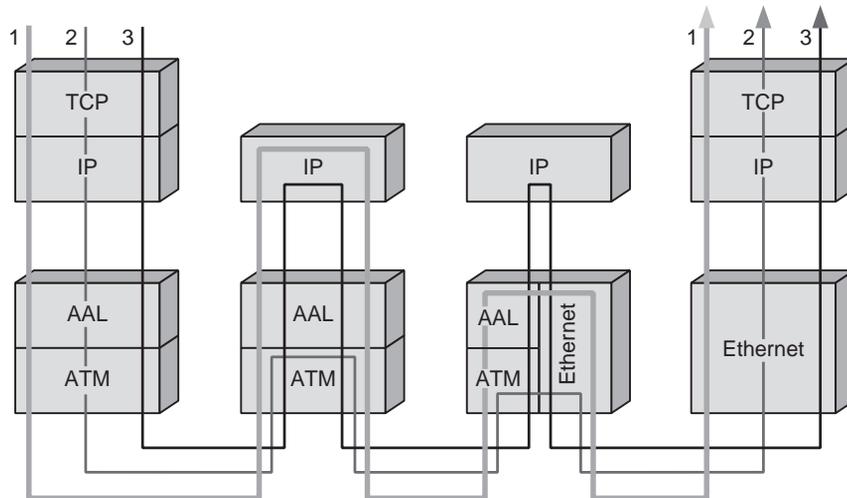
Il est possible de hiérarchiser les demandes.

Un temporisateur TTL (Time to Live) est utilisé. Une référence est encapsulée dans la trame, incluant un TTL et une qualité de service.

Le point fort du protocole MPLS réside dans la possibilité, illustrée à la figure 6.8, de transporter les paquets IP sur plusieurs types de réseaux commutés. Il est ainsi acceptable de passer d'un réseau ATM à un réseau Ethernet ou à un réseau relais de trames. En d'autres termes, il peut s'agir de n'importe quel type de trame, à partir du moment où une référence peut y être incluse.

**Figure 6.8**

*Un réseau MPLS  
avec des sous-  
réseaux distincts*



Les transmissions de données s'effectuent sur des chemins nommés LSP (Label Switched Path). Un LSP est une suite de références partant de la source et allant jusqu'à la destination. Les LSP sont établis avant la transmission des données (*control-driven*) ou à la détection d'un flot qui souhaite traverser le réseau (*data-driven*).

Les références qui sont incluses dans les trames sont distribuées en utilisant un protocole de signalisation, dont les plus importants sont LDP (Label Distribution Protocol) et RSVP (Resource reSerVation Protocol), éventuellement associés à un protocole de routage, comme BGP (Border Gateway Protocol) ou OSPF (Open Shortest Path First). Les trames acheminant les paquets IP transportent les références de nœud en nœud.

Les nœuds qui participent à MPLS sont appelés LER (Label Edge Router) et LSR (Label Switched Router). Un LSR est un routeur situé dans le cœur du réseau, qui participe à la mise en place du circuit virtuel par lequel les trames sont acheminées. Un LER est un nœud d'accès au réseau MPLS. Un LER peut avoir des ports multiples permettant d'accéder à plusieurs réseaux distincts, chacun pouvant avoir sa propre technique de commutation. Les LER jouent un rôle important dans la mise en place des références.

Un équipement qui effectue à la fois une commutation sur une référence et un routage s'appelle un LSR. Les tables de commutation LSFT (Label Switching Forwarding Table) consistent en un ensemble de références d'entrée auxquelles correspondent des ports de sortie. À une référence d'entrée peuvent correspondre plusieurs files de sortie pour tenir compte des adresses multipoint.

La table de commutation peut être plus complexe. À une référence d'entrée peut correspondre le port de sortie du nœud dans une première sous-entrée mais aussi, dans une deuxième sous-entrée, un deuxième port de sortie correspondant à la file de sortie du prochain nœud qui sera traversé, et ainsi de suite. De la sorte, à une référence peuvent correspondre tous les ports de sortie qui seront empruntés lors de l'acheminement du paquet.

Les tables de commutation peuvent être spécifiques à chaque port d'entrée d'un LSR et regrouper des informations supplémentaires, comme une qualité de service ou une demande de ressources particulière.

Une FEC (Forward Equivalence Class) est une classe représentant un ensemble de flots qui partagent les mêmes propriétés. Toutes les trames d'une FEC bénéficient du même traitement dans les nœuds du réseau MPLS. Les trames sont introduites dans une FEC au nœud d'entrée et ne peuvent plus être distinguées des autres flots à l'intérieur de la classe.

Une FEC peut être bâtie de différentes façons et avoir une adresse de destination bien déterminée, un même préfixe d'adresse, une même classe de service, etc. Chaque LSR possède une table de commutation qui indique les références associées aux FEC. Toutes les trames d'une même FEC sont transmises sur la même interface de sortie. Cette table de commutation est appelée LIB (Label Information Base).

Les références utilisées par les FEC peuvent être regroupées de deux façons :

- Par plate-forme : les valeurs des références sont uniques pour l'ensemble des LSR d'un domaine, et les références sont distribuées sur un ensemble commun géré par un nœud particulier.
- Par interface : les références sont gérées par interface, et une même valeur de référence peut se retrouver sur deux interfaces distinctes.

Une référence en entrée permet de déterminer la FEC par laquelle transite le flot. Cette solution ressemble beaucoup à la notion de conduit virtuel dans le monde ATM, dans lequel les circuits virtuels sont multiplexés. Ici, nous avons la même chose, avec un multiplexage de tous les circuits virtuels à l'intérieur d'une FEC, de telle sorte que, dans

ce conduit, l'on ne puisse plus distinguer les circuits virtuels. Le LSR examine la référence et envoie la trame dans la direction indiquée. On voit le rôle capital joué par les LER, qui assignent aux flots de paquets des références qui permettent de commuter les trames sur le bon LSP. La référence n'a de signification que localement, puisqu'il y a modification de sa valeur sur la liaison suivante.

Une fois le paquet classifié dans une FEC, une référence est assignée à la trame qui va le transporter. Cette référence détermine le point de sortie par le chaînage des références. Dans le cas des trames classiques, comme LAP-F du relais de trames ou d'ATM, la référence est positionnée dans le DLCI (Data Link Connexion Identifier) ou dans le VPI/VCI. Dans les autres cas, il faut ajouter un champ, le « shim label ».

Il est difficile de réaliser une ingénierie de trafic dans Internet. L'une des raisons à cela est que le protocole BGP n'utilise que des informations de topologie du réseau. Pour réaliser une ingénierie de trafic efficace, l'IETF a introduit dans l'architecture MPLS un routage à base de contrainte, appelé CR-LDP (Constraint-based Routing-LDP), et un protocole de routage interne à état des liens étendu, appelé RSVP-TE (RSVP-Traffic Engineering).

Comme nous l'avons vu, chaque trame encapsulant un paquet IP qui entre dans le réseau MPLS se voit ajouter par le LSR d'entrée, ou Ingress LSR, une référence au niveau de l'en-tête. Cette référence permet d'acheminer la trame dans le réseau, les chemins étant préalablement ouverts par un protocole de signalisation, RSVP ou LDP. À la sortie du réseau, le label ajouté à l'en-tête de la trame est supprimé par le LSR de sortie, ou Egress LSR. Au LSP (Label Switched Path), qui est le chemin construit entre le LSR d'entrée et le LSR de sortie, sont associés des attributs permettant de contrôler les ressources attribuées à ces chemins.

Ces attributs sont détaillés au tableau 6-1. Ils concernent essentiellement la bande passante nécessaire au chemin, son niveau de priorité, son aspect dynamique par l'intermédiaire du protocole utilisé pour son ouverture et sa flexibilité en cas de panne.

**Tableau 6.1 – Attributs des chemins LSP dans un réseau MPLS**

Attribut	Description
Bande passante	Besoins minimaux de bande passante à réserver sur le chemin du LSP
Attribut de chemin	Indique si le chemin du LSP doit être spécifié manuellement ou dynamiquement par l'algorithme CR-LDP.
Priorité de démarrage	Le LSP le plus prioritaire se voit allouer une ressource demandée par plusieurs LSP.
Priorité de préemption	Indique si une ressource d'un LSP peut lui être retirée pour être attribuée à un autre LSP plus prioritaire.
Affinité ou couleur	Exprime des spécifications administratives.
Adaptabilité	Indique si le chemin d'un LSP doit être modifié pour devenir optimal.
Flexibilité	Indique si le LSP doit être rerouté en cas de panne sur le chemin du LSP.

## Conclusion

Dans ce chapitre, nous avons introduit les différentes solutions permettant d'apporter de la qualité de service à un réseau IP.

Les trois premières solutions concernent les réseaux IP utilisant le routage. En tout premier, le protocole RTP/RTCP qui essaie d'adapter le flux applicatif aux possibilités du réseau : si le réseau devient chargé alors l'application doit ralentir en compressant de façon plus forte les données. Si le réseau devient peu chargé, l'application peut améliorer sa qualité en transportant plus d'information.

Cette solution qui est encore celle utilisée sur le réseau Internet est remplacée petit à petit par une autre méthode : il faut que le réseau s'adapte à l'application et non le contraire. Dans ce cas, IntServ et DiffServ sont les deux propositions de l'IETF. La première ne passant pas l'échelle, c'est DiffServ qui s'est imposé. Les entreprises qui sont passées à la téléphonie sur IP utilisent toutes un réseau DiffServ. Les opérateurs préfèrent la solution des réseaux commutés dans lesquels un chemin est tracé. Dans ce cas, une ingénierie de trafic permet de bien maîtriser les flots et de parvenir assez simplement à de la qualité de service.

Les technologies futures, comme la fibre optique, en cours d'installation par les opérateurs partout en France, nous laissent entrevoir des possibilités presque infinies en matière de bande passante, qui promettent une meilleure stabilité aux applications téléphoniques.

De même, le protocole IPv6 permettra d'étendre le champ d'adressage à 128 bits et contiendra dans son en-tête le champ de priorité DiffServ pour assurer les traitements privilégiés aux données temps réel.

Des protocoles tels que RTP et RTCP devraient néanmoins conserver leur rôle respectif, car ils n'assurent pas directement des fonctions de qualité de service, se contentant de contrôler les applications multimédias. Les plus grands industriels ont approuvé et confirmé l'intégration des protocoles RTP et RTCP au sein de leurs produits. Par ailleurs, ils ont également introduit ce protocole RTP/RTCP dans la conception d'autres protocoles. Ainsi le protocole H.323 intègre-t-il RTP. De même, le protocole SIP a délégué à RTP la tâche de transporter les informations possédant des propriétés temps réel.

Cependant, la solution qui s'impose aujourd'hui dans les entreprises est DiffServ. En effet, peu de complexité est ajoutée aux routeurs : ils doivent juste être capables de lire le champ de priorité et de traiter les paquets en fonction de ce paramètre. Toutes les entreprises qui sont passées à la téléphonie sur IP utilisent DiffServ.

En revanche, la solution DiffServ est moins utilisée par les opérateurs de télécommunications et les FAI. En effet, la norme DiffServ ne fournit pas de garantie forte sur la qualité de service. Cette norme joue statistiquement sur le fait qu'il doit y avoir peu de clients prioritaires et donc que ces clients traversent le réseau comme s'il était vide. Pour les réseaux d'entreprise, cette hypothèse est raisonnable, car on peut facilement surdimensionner les composants du réseau pour que les flux de ces clients prioritaires n'aient

aucun problème. Dans un réseau d'opérateur, beaucoup plus complexe, il n'est pas possible de jouer sur un effet statistique. De ce fait, il est préférable de miser sur des techniques de commutation de type MPLS, dans lesquelles il est facile de réaliser de l'ingénierie de trafic puisque tous les paquets d'un même client passent par le même chemin à l'intérieur du réseau.

# Architectures et sécurité

---

Dans ce chapitre, nous abordons les architectures de la ToIP dans différents types de réseaux. Nous commençons par examiner la téléphonie sur Ethernet, puis sur ATM et enfin sur le relais de trames. Nous nous penchons ensuite sur la téléphonie sur Wi-Fi, qui prend son essor grâce aux Internet Box proposées par les opérateurs. Nous introduisons en fin de chapitre la ToIP sur les nouvelles technologies WiMax pour réseaux métropolitains.

## La téléphonie sur Ethernet

La téléphonie sur Ethernet concerne les entreprises qui souhaitent se doter d'un environnement de ToIP. Selon la taille de l'entreprise, cela peut concerner quelques postes de travail connectés à un même réseau local, jusqu'à des infrastructures internationales permettant à des milliers de postes de travail de communiquer.

Après avoir examiné les problématiques générales d'intégration des services de ToIP et de données informatiques dans les réseaux d'entreprise, nous détaillerons les cas des réseaux d'entreprise à un seul site puis ceux des entreprises multisites.

### *L'intégration voix-données*

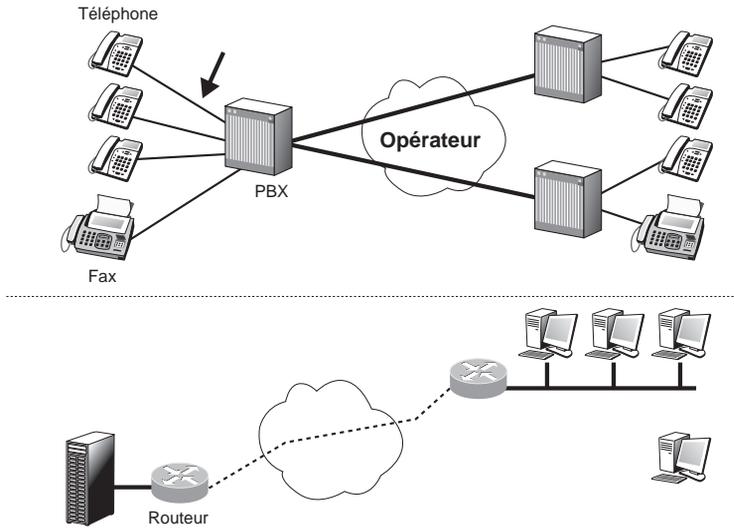
Comme l'illustre la figure 7.1, une forte majorité d'entreprises utilisent deux réseaux différents pour transporter la parole téléphonique et les données jusqu'au poste terminal de l'utilisateur.

Le réseau téléphonique est raccordé à un PABX (Private Automatic Branch eXchange), ou autocommutateur privé. Le réseau de données prend en charge les terminaux informatiques.

Parfois, un troisième réseau s'occupe du transport d'images animées, comme un réseau de télésurveillance ou un réseau de distribution de télévision.

**Figure 7.1**

*Coexistence de deux réseaux d'entreprise pour la voix et les données*

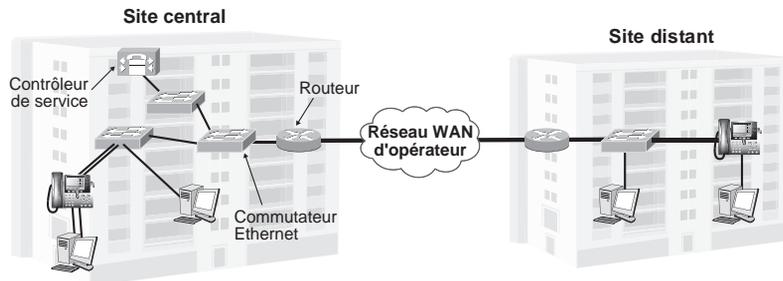


Dans la nouvelle génération de réseaux d'entreprise, ces deux réseaux sont intégrés dans un même réseau de type IP. Ce réseau unique est construit autour d'un réseau d'entreprise Ethernet.

Cette génération de réseaux d'entreprise est illustrée à la figure 7.2. Le réseau d'entreprise comprend plusieurs sites, qui sont reliés entre eux par un réseau d'opérateur, qui peut être de type VPN (Virtual Private Network). Dans chaque site, des téléphones IP sont connectés à des commutateurs Ethernet. Les trois sites sont reliés par un réseau IP d'opérateur raccordé aux sites par le biais de routeurs.

**Figure 7.2**

*Un seul réseau d'entreprise voix-données*



La partie téléphonique de ce réseau est composée de terminaux téléphoniques IP du type illustré à la figure 7.3. Ce téléphone est en fait un routeur intégrant un codec qui paquetsise les octets téléphoniques dans un paquet IP puis encapsule ce paquet dans une trame Ethernet.

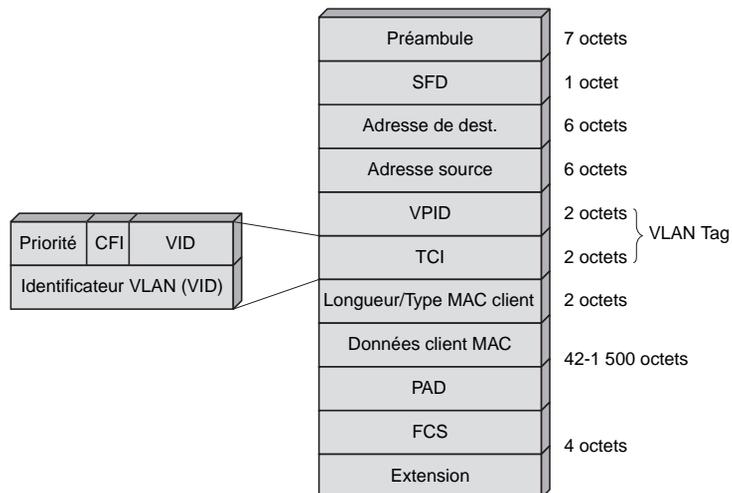
Le téléphone IP est capable de placer le bon niveau de priorité dans le paquet IP et de le traduire dans la trame Ethernet. Le téléphone IP dispose généralement d'une ou plusieurs sorties Ethernet pour connecter le téléphone à un commutateur Ethernet, à un ordinateur personnel et à une autre machine IP.

**Figure 7.3**  
*Téléphone IP*



Dans ce nouveau réseau intégré, il faut que la parole transite dans des temps acceptables pour que l'application téléphonique puisse être reconstituée en sortie. Pour cela, les téléphones IP génèrent les trames Ethernet contenant les paquets IP qui transportent les octets de parole.

**Figure 7.4**  
*Format de la trame Ethernet VLAN*



Suivant le processus DiffServ, les paquets IP se voient attribuer la classe EF (Expedited Forwarding). Comme le réseau d'entreprise est composé essentiellement de commutateurs Ethernet, il faut que cette priorité des paquets IP de parole téléphonique puisse être prolongée dans les trames Ethernet. C'est le niveau de priorité situé dans la zone VLAN (Virtual LAN) qui s'en charge.

Le format de la trame Ethernet VLAN, décrite dans les normes IEEE 802.3ac et IEEE 802.1q, est illustré à la figure 7.4.

L'identificateur VLAN (VLAN Tag) de 4 octets contient un champ VPID (VLAN Protocol Identifier) et un champ TCI (Tag Control Information). Le champ VPID contient en général la valeur constante 0x8100 qui indique une trame de type 802.1q. Le VLAN Tag est inséré entre l'adresse source et le champ Longueur/type du client MAC.

Le champ TCI contient lui-même les trois champs suivants :

- Champ de priorité de 3 bits permettant jusqu'à huit niveaux de priorité.
- Champ d'un bit, le bit CFI (Canonical Format Indicator), qui n'est pas utilisé dans les réseaux IEEE 802.3 et doit être mis à 0 dans ce cas.
- Champ VID (VLAN Identifier) de 12 bits, qui indique l'adresse du VLAN.

Le rôle du champ de priorité de 3 bits est primordial, car c'est lui qui permet d'affecter des priorités aux différentes applications. Cette fonctionnalité est décrite dans la norme IEEE 802.1p. Huit niveaux de priorités permettent d'autoriser des services temps réel comme la parole.

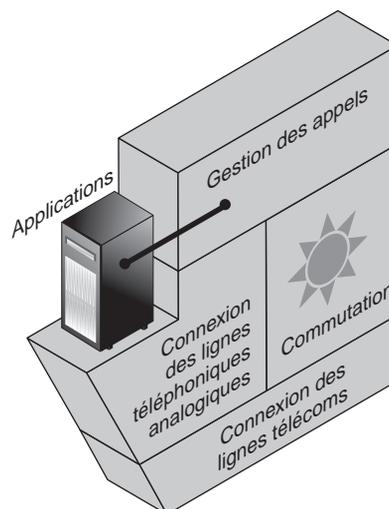
La valeur 7 est réservée à la signalisation. La valeur 6 est celle que l'on utilise pour la téléphonie (classe EF de DiffServ). Les classes 5 à 1 correspondent aux classes AF (Assured Forwarding) et BE (best-effort).

Il est possible que les trames Ethernet soient décapsulées pour passer dans un routeur. Dans ce cas, la zone DSCP du paquet IP est utilisée pour router selon la bonne priorité.

Lorsqu'il existe deux réseaux distincts pour la voix et les données, l'environnement de téléphonie est géré par un PABX tel que celui illustré à la figure 7.5. En cas de réseau unique, ce PABX centralisé est remplacé par un PABX distribué adapté au monde IP, appelé un PBX-IP.

**Figure 7.5**

*Fonctions d'un PABX*



Dans la nouvelle génération de PBX-IP, la gestion des appels est réalisée par un micro-ordinateur qui peut se trouver n'importe où dans l'entreprise, sur n'importe lequel de ses sites. Il peut également être situé chez l'opérateur (solution Centrex). La fonction de commutation est réalisée par l'ensemble des commutateurs de l'entreprise, de même que les connexions des lignes téléphoniques et des lignes télécoms. Seule la signalisation passe par une machine centralisée.

## La téléphonie sur ATM

La technique de transfert ATM a été conçue pour transporter de la parole téléphonique de type G.711 à 64 Kbit/s. C'est la raison de la petite taille de la cellule ATM.

Les 48 octets de données de la trame sont remplis en 48 fois 125  $\mu$ s, c'est-à-dire 6 ms, ce qui reste acceptable, même lorsqu'il y a des échos et que le temps de transit doit rester inférieur à 28 ms. Si la parole téléphonique est compressée par un codeur G.729 à 8 Kbit/s, il faut un temps de 48 ms de remplissage des 48 octets de données puisque le signal donne naissance à un octet à chaque milliseconde.

L'émulation de circuit CES (Circuit Emulation Service) a été la première solution adoptée pour transporter de la téléphonie en paquet. Cette émulation de circuit utilise la couche AAL1 (ATM Adaptation Layer de type 1) de l'environnement ATM, et plus précisément le service CBR (Constant Bit Rate). Les PABX interconnectés par cette solution utilisent des interfaces E1 normalisées (G.703 et G.704). Le service ATM est de type circuit virtuel permanent. La signalisation sur l'interface est portée dans l'IT16 de l'interface E1.

Une autre solution, appelée VTOA (Voice and Telephony Over ATM), ne privilégie pas de protocole AAL spécifique mais demande le support du service VBR-rt (Variable Bit Rate-real time). Le PABX est relié au nœud d'accès du réseau de l'opérateur par un canal de type E1 structuré. La signalisation utilise toujours l'IT16 de l'interface ou un circuit virtuel permanent dédié. Des normes classiques, comme la recommandation CCITT n° 7 ou QSIG (Q-Interface Signaling Protocol), une signalisation développée par l'UIT-T, sont également utilisées. La parole elle-même est transportée par des liaisons permanentes ou commutées.

### AAL2

La couche AAL2, la troisième du modèle ATM, est celle qui s'occupe de la fragmentation et du réassemblage des messages pour obtenir des blocs à la dimension des cellules ATM. Comme nous allons le voir, l'AAL2 détermine des fragments qui peuvent être tout petits de façon à ne pas perdre de temps à attendre des octets téléphoniques et à envoyer les fragments aussi vite que possible.

La parole étant aujourd'hui pratiquement toujours compressée, lorsque la compression est forte, comme lors de l'utilisation du codeur G.723.1, qui diminue le débit du flot à 6,3 Kbit/s, le temps de remplissage d'un paquet ATM devient très long, même pour une petite cellule.

Un calcul simple montre que, pour remplir une cellule de 48 octets à la vitesse de 6,3 Kbit/s, il faut plus de 60 ms. Ce temps est inacceptable si la communication génère des échos ou si d'autres temps d'attente incompressibles s'ajoutent. C'est notamment le cas de la parole numérique dans les réseaux de mobiles, dans lesquels, au temps de remplissage de la cellule, s'ajoute un temps d'accès important sur l'interface air. Une solution possible, mais guère enthousiasmante, à ce problème est de ne remplir que partiellement les cellules. En supposant, par exemple, une compression de 50 %, amenant le débit à 32 Kbit/s, si l'on veut garder les mêmes contraintes que pour des flux à 64 Kbit/s, il ne faut remplir les cellules qu'à moitié. Cette solution induit un flux à 64 Kbit/s de cellules à moitié remplies.

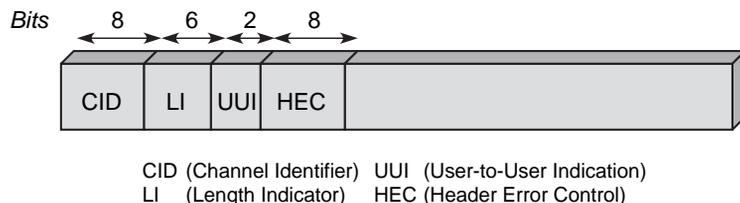
Le rôle de l'AAL2 est de remplir une cellule d'octets provenant de plusieurs connexions de parole, mais avec des débits variables pour les différentes voies basse vitesse. La solution du multiplexage des voies de parole à débit constant est simple, puisqu'il suffit de connaître le numéro de l'octet pour récupérer le numéro de la connexion. Lorsque les flux sont variables, il faut ajouter une information pour savoir à quelle voie de parole appartient le segment.

## Les microtrames AAL2

Dans l'AAL2, le multiplexage des voies de parole est effectué par des microtrames, appelées paquets CPS (Common Part Sublayer). La microtrame AAL2 est illustrée à la figure 7.6.

Figure 7.6

La microtrame de l'AAL2

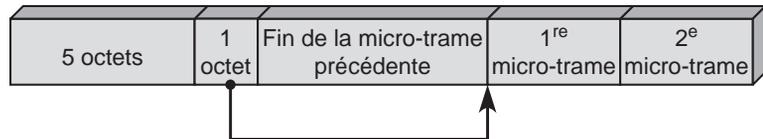


L'en-tête de la microtrame tient sur 3 octets. La zone CID (Channel Identifier) est un identificateur de la voie de parole. Sa longueur de 1 octet permet de multiplexer jusqu'à 248 voies de parole (les valeurs 0 à 7 sont réservées). Le champ LI (Length Indicator) indique la longueur de la microtrame. Le champ UUI (User-to-User Indication) permet de transmettre de l'information d'une extrémité à l'autre de la connexion.

Le champ HEC (Header Error Control) permet la détection et la correction des erreurs sur les deux octets précédents de l'en-tête. La longueur maximale d'une microtrame est de 64 octets, si bien que le transport d'une microtrame requiert parfois plus d'une seule cellule.

Les microtrames étant encapsulées dans les cellules ATM, des bits de bourrage complètent la cellule pour arriver à une longueur de 47 octets, un octet étant réservé, comme dans l'AAL1, pour transmettre des informations de contrôle. La cellule AAL2 est illustrée à la figure 7.7.

**Figure 7.7**  
*La cellule AAL2*



L'octet de contrôle permet de pointer sur le début de la première microtrame encapsulée. En effet, il se peut que le début d'une microtrame ait été transporté dans la cellule précédente. Pour trouver cette valeur, il faut connaître la longueur de la dernière microtrame et compter les octets déjà envoyés dans la fin de la cellule précédente. Le pointeur est réellement utile lorsqu'une cellule est perdue et qu'il faut retrouver le début d'une microtrame. Le pointeur requérant 6 bits, il reste 2 bits, qui permettent d'effectuer une numérotation modulo 2 et une vérification de parité.

Malgré la surcharge engendrée par l'en-tête des micro-trames, l'AAL2 est beaucoup plus efficace que l'utilisation d'une connexion unique pour une voie de parole et, d'une façon générale, que la téléphonie sur IP. C'est la raison pour laquelle les technologies de troisième génération, comme l'UMTS ou HSDPA (High Speed Downlink Packet Access), ont adopté cette solution.

## La téléphonie sur le relais de trames

Le transport de la parole dans le relais de trames (Frame Relay) pose des problèmes similaires à celui des réseaux ATM. Sur une liaison virtuelle, où les paquets peuvent atteindre plus de 4 000 octets, il est indispensable de multiplexer sur une même liaison plusieurs voies de parole. La proposition FRF.11 du Frame Relay Forum décrit une solution de mini-trame semblable à celle de l'AAL2 pour transporter les voies de parole.

La possibilité d'avoir un commutateur occupé par la transmission d'une longue trame LAP-F crée toutefois une difficulté supplémentaire. Il faut donc un mécanisme de priorité pour laisser passer les petits paquets portant de la parole téléphonique.

## Intégration de la téléphonie dans le relais de trame

L'intégration de la téléphonie dans un réseau relais de trames est fortement utilisée depuis une dizaine d'années. Elle reste une solution simple et peu onéreuse, même si de nouvelles solutions plus puissantes sont apparues depuis le début des années 2000.

La figure 7.8 illustre l'intégration de la voix et des données par l'intermédiaire d'un VFRAD (Voice Frame Relay Access Device), un équipement capable de remplir des trames LAP-F dans un réseau en relais de trames.

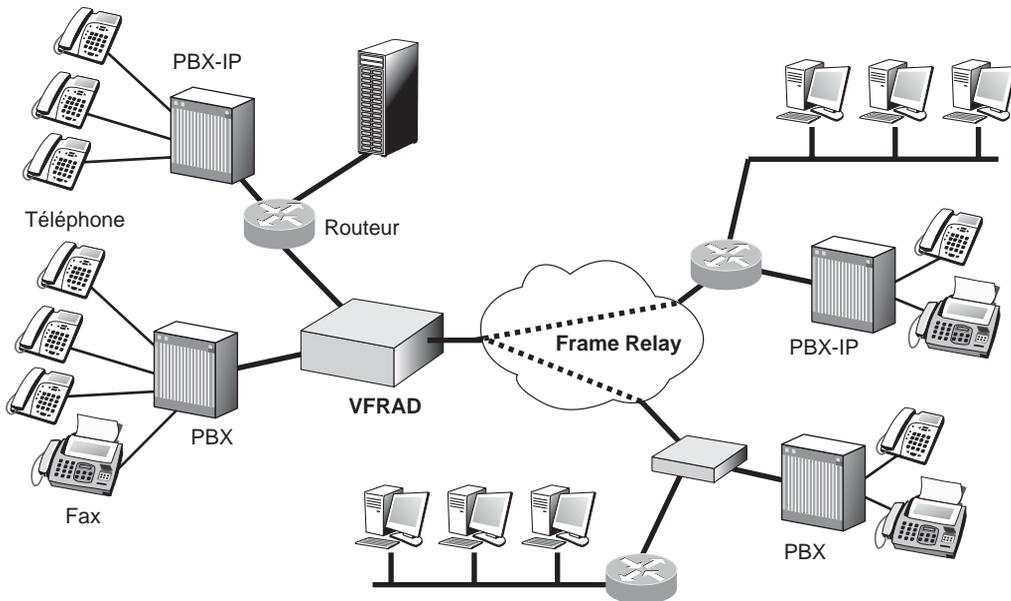


Figure 7.8

*Intégration voix-données par un VFRAD*

Un VFRAD est un équipement qui permet d'encapsuler dans les trames LAP-F du relais de trames des octets de parole téléphonique et de les multiplexer sur une même liaison virtuelle, qui n'est autre qu'un circuit virtuel de niveau 2 tel que défini dans le relais de trames.

Dans cette architecture, il faut autant de VFRAD que d'accès au réseau pour relier les différents sites entre eux, et les réseaux de chaque site ne sont pas intégrés. L'intégration n'a lieu que sur le réseau relais de trames, où une même liaison virtuelle permet de transporter les trames de parole et de données. La parole téléphonique peut être restituée à l'arrivée, car le relais de trames comporte une option qui garantit le délai de transit dans le réseau.

Avec cette option, le transport de voix utilise une compression de la parole téléphonique à un taux moyen de 8 Kbit/s.

La figure 7.9 illustre l'intégration des différentes applications dans cette solution.

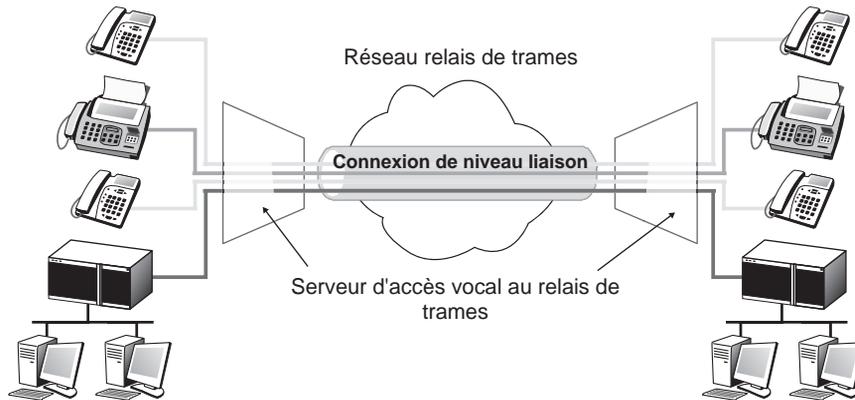


Figure 7.9

*Intégration voix-données dans le relais de trames*

Cette solution, normalisée en décembre 1998 par le Frame Relay Forum, met en œuvre un mécanisme simple de point-à-point utilisant un circuit virtuel. La figure 7.10 illustre l'encapsulation de plusieurs voix de parole dans une trame suivie d'une trame de données.

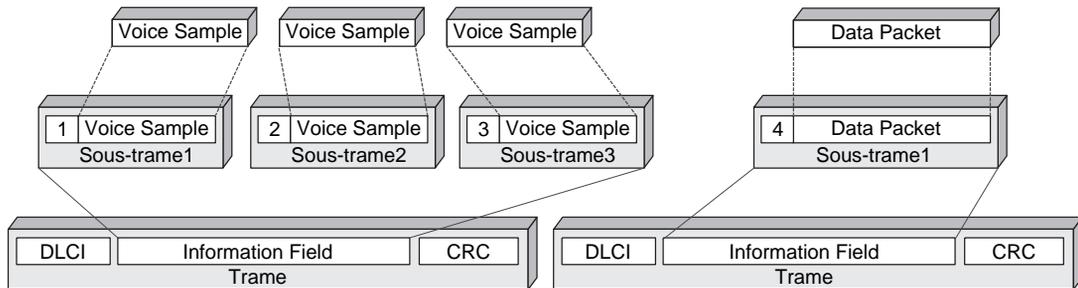


Figure 7.10

*Intégration voix-données au niveau trame*

La deuxième solution que nous allons étudier est celle de l'intégration dans un même environnement IP de la parole téléphonique provenant de combinés analogiques et des données émises par les terminaux informatiques. Cette solution est décrite à la figure 7.11.

Les combinés téléphoniques analogiques sont reliés à un PBX-IP, c'est-à-dire un auto-commutateur privé capable de générer en sortie des paquets IP transportant les octets de parole téléphonique numérisée. Le PBX-IP gère également la supervision en transformant la signalisation téléphonique classique CCITT n° 7 en une signalisation capable de traverser le monde IP, comme H.323 ou SIP.

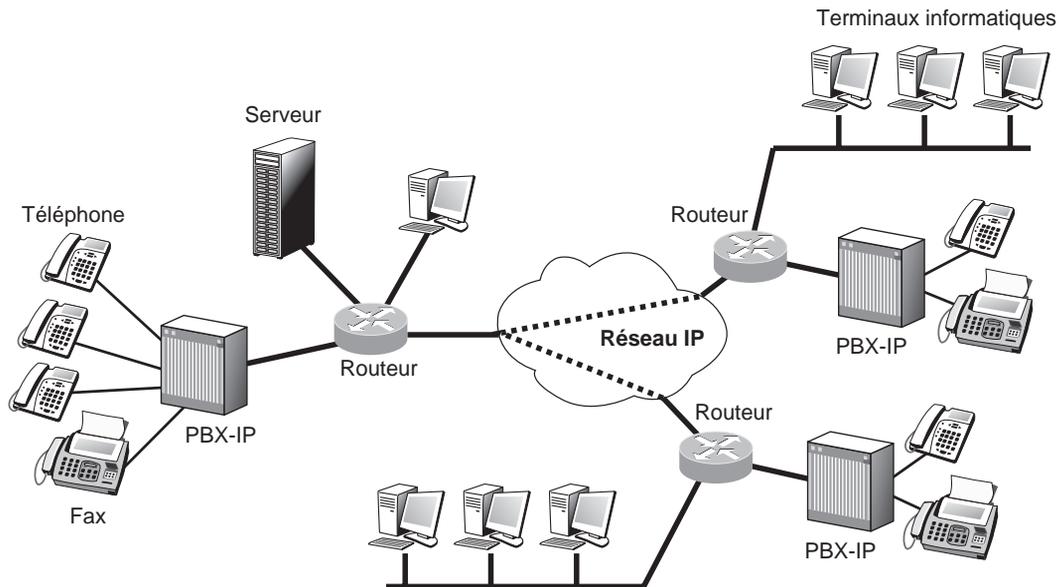


Figure 7.11

*Intégration voix-données dans une architecture IP*

Dans cette solution, les paquets IP sont encapsulés, au niveau des serveurs et des stations de travail IP, dans des trames Ethernet, lesquelles sont transmises sur un réseau Ethernet jusqu'au routeur de sortie de l'entreprise. Après décapsulation de la trame Ethernet, les paquets IP sont intégrés dans une trame LAP-F du relais de trames pour être acheminés vers le routeur situé de l'autre côté du réseau. Dans ce routeur, la trame LAP-F est de nouveau décapsulée afin de retrouver le paquet IP et d'insérer ce dernier dans une trame Ethernet. Cette trame est acheminée vers le terminal informatique ou vers le PBX-IP.

Le réseau intersite peut être un relais de trames, un réseau IP d'opérateur ou un réseau IP privé d'entreprise. Le réseau de l'opérateur peut s'appuyer sur des techniques diverses, comme ATM ou MPLS. Dans tous les cas, pour garantir la qualité de service nécessaire à la parole téléphonique, il faut que le réseau intersite soit capable de garantir un temps de transit au travers du réseau. Cette garantie est négociée avec l'opérateur du réseau, opérateur de télécommunications ou gestionnaire du réseau d'entreprise, de façon que les performances soient compatibles avec la qualité attendue par l'utilisateur. Pour cela, un SLA (Service Level Agreement) est négocié avec l'opérateur du réseau, afin de préciser le temps maximal de transfert à l'intérieur du réseau.

Dans cette solution, le réseau d'entreprise global doit posséder un réseau sur chaque site de l'entreprise et un réseau intersite, tous individuellement capables d'assurer une qualité de service, et donc des temps de traversée bornés. On peut en déduire que le temps de transit d'un équipement terminal à un autre est borné. Ce temps doit rester inférieur à 150 voire 300 ms pour une application de téléphonie. Pour cela, le réseau doit faire appel à des techniques de priorité permettant aux paquets prioritaires de voir le système comme étant quasiment vide. Il faut donc un surdimensionnement du réseau de site par rapport à la somme des débits des différentes voix téléphoniques. Cela n'est généralement pas compliqué à obtenir avec des commutateurs Ethernet qui gèrent les trois bits de priorité IEEE 802.3q et un réseau d'un débit égal à 100 Mbit/s.

Il faut en outre modifier les PBX en PBX-IP de façon que les octets de parole puissent être encapsulés dans des paquets IP. Une autre solution, allant jusqu'au bout de l'intégration dans IP, consiste à disposer de téléphones IP capables de générer directement des paquets IP, de telle sorte qu'il n'y ait plus besoin de PBX-IP. Dans ce cas, un serveur de supervision prend en charge la signalisation afin de permettre de faire sonner le téléphone distant.

La figure 7.12 illustre cette solution d'intégration du réseau d'entreprise dans un environnement complètement IP.

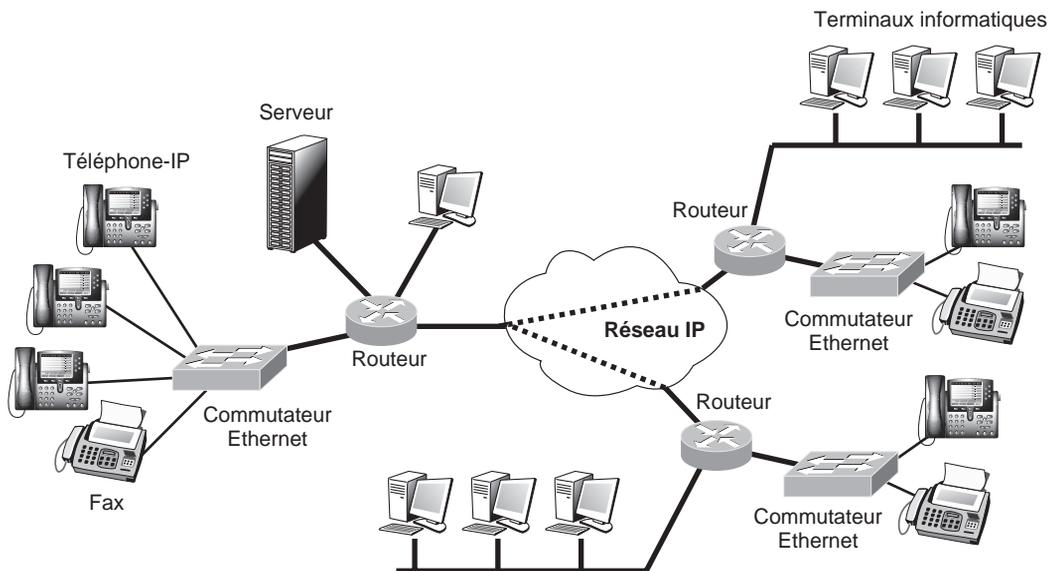


Figure 7.12

*Intégration voix-données dans un réseau d'entreprise tout-IP*

Les avantages de cette solution proviennent du vrai multiplexage IP et de l'intégration de bout en bout des applications voix et données. Ce très haut degré d'intégration rendra à moyen terme cette solution peu onéreuse. À long terme, les coûts devraient encore baisser du fait de l'arrivée en grand nombre d'opérateurs IP proposant des SLA.

Parmi les inconvénients de cette technique, citons notamment les suivants :

- mutation des équipements à mettre en œuvre à relativement long terme ;
- coût encore élevé des téléphones IP ;
- relative incompatibilité entre équipementiers des algorithmes de gestion de la qualité de service.

De nombreuses solutions intermédiaires, comme celle illustrée à la figure 7.13, permettent à l'entreprise de passer doucement d'un environnement non intégré à un environnement intégré. Dans cette figure, le PBX est découpé en plusieurs parties, des PBX-IP et des PBX non-IP. Les PBX-IP produisant les paquets IP sont raccordés aux routeurs, et les octets téléphoniques transitent par la partie IP du VFRAD. Les PBX numériques classiques intègrent pour leur part leurs octets téléphoniques directement dans une trame LAP-F du relais de trames.

Les inconvénients de cette transition sont les suivant :

- nécessité d'avoir des équipements voix sur IP dans l'ensemble des sites pour pouvoir récupérer les communications IP et les envoyer vers des PBX classiques ;
- investissement dans des FRAD intégrant la voix sur IP, c'est-à-dire des routeurs spécifiques relativement coûteux par rapport aux routeurs tout-IP.

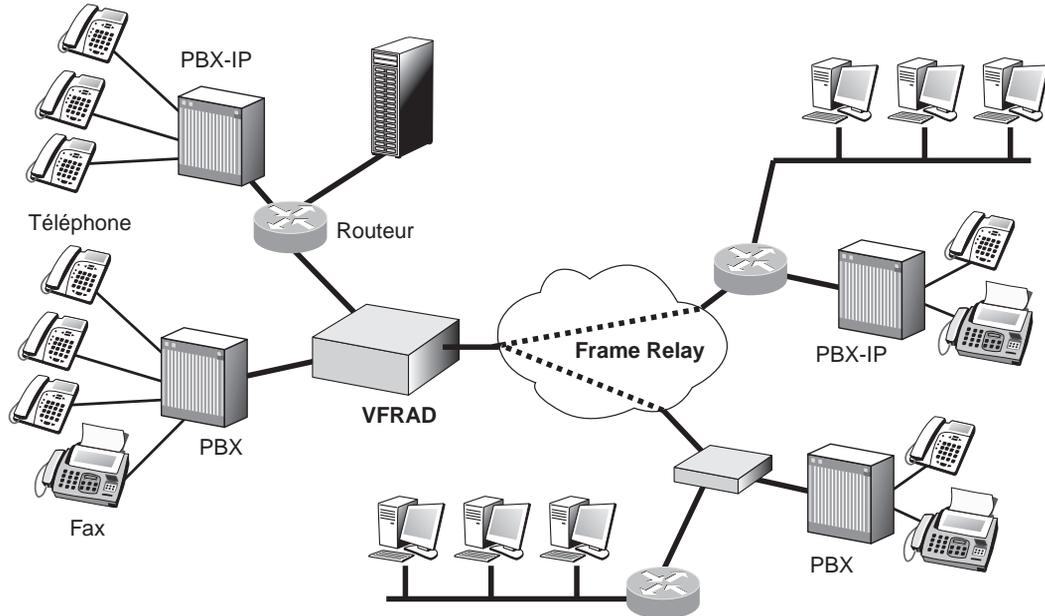


Figure 7.13

*Passage d'un environnement non intégré à un environnement intégré*

La tendance à l'intégration est inéluctable. Vers les années 2010, toutes les entreprises auront intégré la téléphonie et les données sur un même réseau, ainsi que la vidéo. La vidéo génère en effet un flot de paquets synchrones, mais avec des contraintes temporelles moins fortes. Elle s'intégrera au réseau de l'entreprise dès que les interfaces d'accès et les réseaux internes seront suffisamment puissants.

## La téléphonie sur réseaux sans fil

Plus qu'une évolution technologique, la ToIP est aujourd'hui une application réseau incontournable, dont l'atout majeur est incontestablement le prix.

Si les communications téléphoniques sont généralement facturées à la durée et à la distance, les communications sur Internet sont le plus souvent forfaitaires, sans limitation de distance ni de temps. Toutes les applications dont les flux transitent sur le réseau Internet sont incluses dans le prix. Encore faut-il trouver les applications susceptibles de séduire les utilisateurs. La ToIP est précisément l'une des possibilités.

Si la ToIP a pour principale vocation d'offrir une solution de rechange à moindre coût à la téléphonie sur circuit classique, le sans-fil offre bien d'autres avantages. Les réseaux sans fil permettent aux utilisateurs de s'abstraire des contraintes de câblage sur les derniers mètres de raccordement jusqu'à leur poste. La zone de couverture devient étendue, ce qui procure une souplesse d'utilisation accrue. Le gain se traduit également par une facilité de déploiement et d'extensibilité du réseau, ce dernier devenant ambiant. C'est donc presque naturellement que la ToIP a donné naissance à la téléphonie sur réseaux sans fil, ou ToWLAN (Telephony over Wireless LAN).

Cette nouvelle application ne va toutefois pas sans contraintes. Il s'agit principalement d'assurer la portabilité en sans-fil des éléments essentiels au déploiement de la voix.

### Contraintes de la ToIP sans fil

La technologie de ToIP sans fil a pour objectif de permettre aux utilisateurs de téléphoner directement en IP, donc à un coût extrêmement bas et depuis de nombreux points sans avoir besoin d'une connexion physique. Elle représente un pas supplémentaire vers la convergence des flux audio, vidéo et données sur un médium unique.

La problématique de la ToIP sur WLAN est évidemment semblable à celle de la téléphonie sur réseau terrestre, si ce n'est qu'il s'y ajoute la difficulté de réaliser la traversée de l'interface air dans un laps de temps suffisamment court pour satisfaire à la contrainte temps réel. Il s'agit d'une vraie difficulté dans la mesure où les équipements raccordés au point d'accès Wi-Fi sont *a priori* indépendants les uns des autres. De plus, les environnements Wi-Fi offrent un débit particulièrement fluctuant, et il est quasiment impossible de connaître le débit disponible à l'instant suivant.

Les terminaux de ToIP sur Wi-Fi assurent les fonctions de codec, de paquetsation et d'encapsulation dans la trame Ethernet pour transiter sur le réseau Wi-Fi.

### Contraintes de la voix

Dans la mesure où la téléphonie est une application d'utilisation courante, les usagers ont de fortes exigences à l'égard du service. En ToIP, il est impératif de respecter ces exigences afin de rendre un service comparable à celui proposé par la téléphonie RTC ou PSTN (Public Switched Telephone Network).

Dans une communication interactive, le temps constitue le plus fondamental des facteurs. Les deux paramètres temporels nécessaires pour assurer un confort d'utilisation raisonnable sont le délai de transit et la gigue.

Le délai de transit sur le réseau (*end-to-end delay*) désigne le laps de temps entre le moment où l'émetteur prend la parole et celui où le destinataire écoute le message. Il inclut en fait la somme des temps de conception du paquet au niveau de l'émetteur (numérisation, codage, compression, paquetsisation), de traitement du paquet au niveau du récepteur (décodage, décompression, réassemblage, conversion du signal numérique en un signal analogique) et d'acheminement du paquet de bout en bout dans le réseau (délai de propagation, de transmission et de commutation dans les nœuds, incluant le délai de séjour dans les files d'attente des commutateurs).

Pour ne pas perturber le rythme d'une conversation et assurer une interactivité entre les correspondants, ce délai doit être aussi court que possible, comme nous l'avons vu au chapitre 2. Une tolérance de plusieurs centaines de millisecondes, voire de plusieurs secondes est admise pour le temps d'établissement de la communication. Ce délai peut bien sûr être mis à profit pour configurer le réseau. Toutefois, le pourcentage de succès d'un appel initié doit être supérieur à 95 %, et la conversation ne doit jamais être interrompue. La disponibilité du système constitue un élément clé dans la fiabilité d'une application de voix sur IP.

La gigue est un paramètre également très important de la parole téléphonique. Prenons un exemple. Si les délais de transit de tous les paquets transportant une conversation téléphonique sont égaux à 100 ms, la moyenne est également de 100 ms. Si maintenant le délai de transit d'un paquet sur deux est de 120 ms et d'un paquet sur deux de 80 ms, la moyenne est toujours de 100 ms mais la gigue a augmenté, elle est passée de 0 à 20 ms. Tant que la gigue reste acceptable, il est possible de resynchroniser les paquets. Mais lorsque la gigue augmente trop et dépasse le temps maximal de latence, l'application commence à se dégrader. Par exemple, si le délai maximal acceptable dans une application de téléphonie est de 150 ms et si la moyenne est de 100 ms mais la gigue de nombreux paquets est supérieure à 50 ms, cela veut dire que de nombreux paquets ne pourront pas être remis correctement puisqu'ils arriveront trop tard au récepteur.

La téléphonie sur IP peut offrir un ensemble de fonctionnalités supplémentaires visant à simplifier son usage et à améliorer la convivialité. C'est dans ce domaine que les services peuvent innover à moindre frais. Le nomadisme (possibilité de se connecter avec des informations de profils identiques dans des localisations distinctes) est ainsi rendu possible par l'attribution à un utilisateur d'un même identifiant, quelle que soit sa localisation. Il devient même envisageable d'offrir la mobilité (possibilité de ne pas rompre une communication durant un déplacement).

## Contraintes des transmissions sans fil

Le câblage des terminaux est une opération souvent fastidieuse, surtout dans de nouvelles installations. Avec le sans-fil, il n'est plus besoin de câbler les derniers mètres qui relient l'utilisateur au réseau. Mais, dans ce cas, la ressource rare reste la bande passante. Les débits offerts sont limités et rarement garantis, ce qui constitue une contrainte majeure pour offrir la qualité de service indispensable à la voix.

En outre, la transmission sans fil repose sur la qualité du lien radio. Or celle-ci dépend de la qualité de l'interface radio, laquelle peut se dégrader sous l'influence de plusieurs facteurs, tels que des interférences avec d'autres équipements utilisant la même bande de fréquences, des obstacles entre la source et la destination ou de l'éloignement important de l'émetteur et du récepteur.

Un problème à ajouter est l'indépendance des terminaux entre eux et donc la quasi-impossibilité *a priori* d'introduire des priorités qui permettraient de favoriser la voix sur IP. Comme nous le verrons dans cette section, il est cependant possible d'introduire une certaine priorité, comme le propose la norme IEEE 802.11e.

Déjà importantes dans les réseaux IP, les pertes de paquets sont particulièrement courantes en sans-fil. La perte d'un paquet entraîne la perte d'une bribe de parole. Pour une bonne qualité d'écoute, le taux de perte de paquets doit être inférieur à 20 %. Pour faire baisser le taux d'erreur sur l'interface radio, la technologie Wi-Fi réémet automatiquement les trames, augmentant d'autant le temps de traversée du réseau.

Si le taux d'erreur dépasse une certaine limite, la transmission d'un élément binaire est effectuée sur deux signaux d'horloge à la place d'un seul, ce qui diminue le débit d'un facteur deux. Dans le cas d'un réseau IEEE 802.11b, le débit de base passe alors de 11 Mbit/s à 5,5 puis à 2 puis à 1 Mbit/s. Cette baisse du débit pose un énorme problème de qualité de service, puisque la bande passante n'est plus du tout celle attendue. De plus, le débit réel correspond approximativement à la moitié du débit brut.

Une autre contrainte vient du fait que la confidentialité des communications entre usagers doit être préservée des écoutes clandestines. Cette précaution est d'autant plus indispensable lorsqu'il s'agit de transmissions sans fil, puisque l'interface air est par nature ouverte et accessible à toute personne située dans la portée des ondes hertziennes. Bien qu'indispensable, la gestion de la sécurité engendre un débit supplémentaire, qui peut nuire aux données temps réel, ce qui nécessite de trouver un compromis entre sécurité et rapidité des flux.

## La qualité de service

Au fur et à mesure que les normes des réseaux sans fil évoluent, les débits augmentent, de 11 Mbit/s pour la version IEEE 802.11b à 54 Mbit/s pour 802.11a et 802.11g et 190 Mbit/s pour la nouvelle version 802.11n. Parallèlement, les applications les plus courantes deviennent de plus en plus gourmandes en bande passante, si bien que les besoins croissent en même temps que les débits. Il devient dès lors nécessaire d'adapter une politique de gestion appropriée aux flux qui transitent dans le réseau.

Dans l'Internet IPv4 que l'on utilise aujourd'hui, la qualité de service est de type best-effort, c'est-à-dire « au mieux », manière détournée de dire que la qualité de service n'est pas gérée, puisque tous les flux sont traités de la même façon. Pour le traitement de la voix, c'est une technique insuffisante. Il faut donc différencier les flux selon leur importance.

En fait, il ne suffit pas de mettre en œuvre un mécanisme sur une seule couche. Il faut optimiser chacun des traitements pour les adapter aux spécificités d'une application synchrone temps réel. On distingue deux manières de gérer les flux : soit au niveau du routage de bout en bout, en réservant de la bande passante pour garantir un débit ou en différenciant les flux, soit au niveau MAC, en adaptant l'accès au support en fonction de la priorité des flux à transmettre.

Nous avons vu au chapitre 6 différentes techniques de gestion de la qualité de service. Elles sont également applicables à la téléphonie sur Wi-Fi, sauf pour l'interface radio, sur laquelle la gestion des priorités entre équipements distribués est complexe.

Nous allons examiner en détail la technologie 802.11e, qui a été mise au point pour introduire une forme de priorité en raccourcissant ou rallongeant les temporisateurs des équipements plus ou moins prioritaires.

### La norme IEEE 802.11

Le rôle de la couche MAC (Medium Access Control) est de permettre à plusieurs stations d'accéder à un support partagé. Dans le cas des réseaux sans fil, c'est l'interface air qui constitue le support des ondes hertziennes et permet de diffuser les données. Bien qu'elle soit spécifique au réseau sans fil, la couche MAC définie par la norme 802.11 est très proche de celle des réseaux Ethernet (802.3).

802.11 définit différents temporisateurs intertrame, ou IFS (Inter-Frame Space), qui permettent d'instaurer des mécanismes de priorités différenciées selon le type de trames émises. Plus le temporisateur est court, moins la station doit attendre avant d'émettre et donc plus sa trame est prioritaire.

On distingue les quatre temporisateurs IFS suivants :

- SIFS (Short IFS), qui accorde une priorité haute, essentiellement aux messages d'acquiescement.
- PIFS (PCF IFS), qui correspond à une priorité moyenne, permettant au point d'accès d'avoir un temps d'accès prioritaire par rapport aux stations, et est adapté aux services temps réel.
- DIFS (DFC ou Distributed IFS), qui correspond à une priorité faible, de type best-effort et est généralement réservé à la transmission de données sans contrainte de temps.
- EIFS (Extended IFS), le plus long des IFS, qui est utilisé lorsqu'une erreur est détectée.

La norme 802.11 propose deux mécanismes élémentaires de gestion de l'accès au médium (la couche MAC) : la méthode DCF (Distributed Coordination Function), avec contention, et la méthode optionnelle PCF (Point Coordination Function), sans contention (pas de collision possible).

DCF est la méthode d'accès par défaut dans 802.11. Elle a été conçue pour prendre en charge des données asynchrones (sans priorité). Son fonctionnement repose sur le protocole CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), ou accès multiple à détection de porteuse et évitement de collision. Ce dernier fonctionne selon trois procédures : l'écoute de la porteuse, l'algorithme de back-off et l'utilisation d'acquievements positifs.

L'écoute du support précède toute transmission. Elle s'effectue sur deux couches distinctes du modèle OSI. Au niveau physique, c'est le mécanisme PCS (Physical Carrier Sense) qui est utilisé. Il permet de détecter l'activité des autres stations en analysant les flux diffusés et transitant sur le support hertzien. Au niveau liaison, sur la sous-couche MAC, c'est le mécanisme VCS (Virtual Carrier Sense) qui détermine l'activité du support.

Pour cela, chaque station gère un temporisateur NAV (Network Allocation Vector) indiquant la durée d'occupation du support, donc le temps que les stations doivent attendre pour tenter d'émettre à leur tour. La durée de ce temporisateur est fixée en exploitant le contenu de trames spéciales, appelées RTS (Ready To Send) et CTS (Clear To Send). Ces trames servent à effectuer une réservation explicite du support.

Le message RTS est diffusé à l'initiative de la station qui souhaite émettre. Il comporte l'indication de la station source, de la station destinataire et de la durée de la transmission envisagée (incluant la durée des messages d'acquievement du récepteur). Si le destinataire accepte la communication, celui-ci retourne un message CTS en réponse à la requête RTS. Comme toutes les stations qui veulent émettre sont à l'écoute de l'activité du support, elles reçoivent également le RTS et/ou le CTS. Le champ de durée de ces trames leur permet de mettre à jour leur temporisateur NAV.

Le mécanisme RTS/CTS permet de pallier le problème de la station cachée. Ce problème évoque le cas où deux stations A et B sont trop lointaines l'une de l'autre pour se détecter mutuellement et tentent de communiquer avec une même station intermédiaire I. Dans ce cas, comme la station B n'entend pas lorsque la station A émet une trame à la station I, elle risque d'émettre elle-même vers la station I une trame qui entrera en collision avec la trame de la station A. Avec le mécanisme RTS/CTS, la station B n'entend pas non plus la demande RTS faite par A, mais elle entend la réponse CTS diffusée par I. Ainsi, la demande préalable et surtout la réponse du destinataire permettent d'assurer qu'une seule station est en train d'émettre.

Le mécanisme VCS est optionnel et peut être appliqué systématiquement, à la demande ou jamais. En règle générale, son utilisation est préconisée lorsque les trames sont de grande taille. En cas de collision d'une trame, non seulement le support est parasité pendant toute la durée de la transmission, ce qui constitue un gaspillage de la bande passante, mais sa retransmission devient coûteuse. Inversement, l'ajout de trames RTS/CTS peut

s'avérer moins rentable que la perte de la trame à transmettre. Il s'agit donc de trouver un compromis. La perte (par collision, interférence, etc.) d'une trame longue étant coûteuse en bande passante, cette trame est fragmentée en unités appelées MSDU (MAC Service Data Unit) afin de restreindre les retransmissions éventuelles. Tous les fragments sont émis séquentiellement (sans nouvelle demande du support) et acquittés par le destinataire, lequel se charge de réassembler les fragments. Le support n'est libéré qu'après la transmission de l'intégralité de la trame.

Le processus CSMA/CA débute par l'écoute de la porteuse. Une station souhaitant émettre une trame vérifie la disponibilité du support en l'écoutant pendant un délai de valeur DIFS. Si cette condition est vérifiée, la station peut émettre. Dans le cas contraire, elle poursuit l'écoute jusqu'à ce que le support devienne libre. Afin de réduire la probabilité d'émission simultanée par plusieurs stations, elle enclenche ensuite une procédure fondée sur l'algorithme de back-off. Dans cet algorithme, la station retarde sa transmission pendant une durée, appelée back-off time, choisie aléatoirement et bornée entre les valeurs 0 et CW. CW est la fenêtre de contention (Contention Windows), comprise entre deux seuils prédéfinis  $CW_{\min}$  et  $CW_{\max}$ .

Si la disponibilité du support se confirme tout au long de ce délai, la station est autorisée à émettre. À l'inverse, si une autre station vient à émettre durant ce délai, le temporisateur est suspendu tant que le canal reste occupé, et une nouvelle valeur de back-off time est choisie. Pour restreindre davantage les risques de collision, la valeur de la variable CW est incrémentée (jusqu'à la valeur seuil maximale  $CW_{\max}$ ) à chaque tentative, selon la formule suivante :

$$CW_{\text{nouveau}} = (CW_{\text{ancien}} + 1) \times FP - 1$$

où FP est un facteur de persistance fixé à la valeur 2 dans l'algorithme de back-off et qui détermine l'accroissement relatif entre deux retransmissions.

L'acquiescement positif permet de valider la réception des trames. Dans les systèmes hertziens, la transmission de données couvre la capacité de réception d'un terminal, ce qui entraîne qu'une station ne peut émettre et recevoir en même temps. Par conséquent, les collisions ne peuvent être détectées lors de l'envoi, mais seulement après la diffusion complète de la trame.

Pour réduire le nombre de collisions et accroître les performances du réseau, le mécanisme de prévention de collisions est utilisé. Il consiste en un acquiescement explicite par le récepteur, à défaut duquel l'émetteur est contraint d'émettre à nouveau sa trame, passé un délai d'attente déterminé. Ainsi, à la réception d'une trame, la station destination vérifie son intégrité, par le biais du champ CRC (Cyclic Redundancy Check), puis, si la trame est correcte, émet un acquiescement (après un délai SIFS). Si la station émettrice ne reçoit pas l'acquiescement de sa trame, elle considère qu'une collision ou perte s'est produite et retransmet la trame. Après un nombre prédéfini de tentatives sans acquiescement, le support est considéré comme instable, et l'émission est abandonnée.

Comme il se fonde sur CSMA/CA, qui est une méthode probabiliste, le protocole DFC est efficace dans un réseau peu ou moyennement chargé, mais se comporte de manière moins efficace lorsque la charge du réseau est importante. Surtout, la qualité de service n'est pas gérée avec DCF, puisque, par nature, CSMA/CA est un protocole avec contention équitable, qui ne permet pas de garantir un délai pour l'acheminement des paquets.

Optionnelle, la méthode d'accès sans contention PCF (Point Coordination Function) proposée par le groupe 802.11 n'est utilisable que dans une configuration de réseau fondée sur une infrastructure, au sein d'une cellule appelée BSS (Basic Service Set), et est optimisée pour les transmissions à rythme régulier (isochrone), ce qui est particulièrement utile pour les flux temps réels tels que la téléphonie.

Dans ce mécanisme, les stations ne peuvent émettre ou recevoir des données que si elles y ont été invitées par une station spéciale, appelée Point Coordinator (PC). En pratique, le PC est implémenté au niveau d'un point d'accès. Il est le seul à avoir le contrôle sur le droit d'émission. Son rôle est d'arbitrer l'accès au support en interrogeant tour à tour les terminaux pour savoir si ces derniers veulent émettre et de leur donner l'accès. Cette méthode est appelée scrutation, ou polling.

Concrètement, le temps est divisé en intervalles de temps, appelés supertrames, qui se composent d'une période de contention CP (Contention Period) et d'une période sans contention CFP (Contention Free Period). La période avec contention utilise la méthode d'accès DCF. La période sans contention introduit un temporisateur PIFS, plus petit que le DIFS. Pendant cette période CFP, le PC envoie aux stations destination des trames de données, appelées CF-Down, après un délai PIFS. Les stations envoient quant à elles leur trame de données, appelée CF-Up, après un délai SIFS. La période sans contention se termine par une trame particulière CF-End, précédée d'un délai PIFS.

La méthode PCF laisse donc entrevoir la possibilité d'introduire de la qualité de service dans un réseau sans fil, mais ses contraintes deviennent vite limitatives. À chaque scrutation des stations par le coordinateur, une seule trame peut être envoyée. En outre, la scrutation des stations est un mécanisme peu scalable, compte tenu du temps offert systématiquement à toutes les stations, alors que nombre d'entre elles n'ont pas de trames à émettre. De surcroît, dans les trames émises, rien n'indique leur priorité par rapport à d'autres, si bien que le coordinateur ne peut favoriser un type de trafic en l'absence de profils types définis. Aucune garantie de délai ni de gigue n'est possible.

Par ailleurs, la méthode PCF n'est pas efficace dans la situation de la station cachée. Une station située en bordure d'un BSS est contrôlée par le PC de sa BSS. Si son PC l'autorise à émettre, rien n'indique que sa transmission ne va pas entrer en collision avec une station d'un BSS voisin.

Le frein le plus lourd à l'utilisation de PCF est sans doute le fait que cette méthode ne fonctionne qu'en mode infrastructure, alors même qu'elle est, dans la pratique, très rarement implémentée au niveau des points d'accès.

### La norme IEEE 802.11e

Aucune des méthodes DCF ni PFC ne permettant de gérer de façon satisfaisante la qualité de service, la norme 802.11e s'est assigné l'objectif de spécifier la gestion au niveau MAC de la qualité de service de façon à permettre aux paquets de téléphonie de passer en priorité.

Ce mécanisme repose sur le principe de différenciation de service au niveau du contrôle d'accès. Les stations gèrent les priorités de leurs flux et y affectent des temporisateurs. Un flux temps réel a de la sorte un temporisateur d'émission plus court qu'un flux moins sensible au délai, ce qui le favorise pour émettre. Deux mécanismes de contrôle d'accès au support, EDCF et HCF, sont proposés sur ce principe.

EDCF (Enhanced Distributed Coordination Function) est une amélioration de DCF pour la gestion de la qualité de service. Il s'agit d'un mécanisme à contention qui reprend l'algorithme CSMA/CA. EDCF repose en fait sur une différenciation de services au niveau des flux. Ces derniers sont répartis en catégories en fonction de leur priorité selon huit classes de trafic, de la valeur 7 (priorité la plus importante) à la valeur 0 (priorité la plus basse).

La valeur de ces priorités est donnée à la fois en fonction de celle affectée à un utilisateur, appelée UP (User Priority), et de celle affectée à l'application utilisée, appelée TSPEC (Traffic SPECification), qui peut être dynamique. Par exemple, la priorité d'un flux associé à la voix est plus importante que celle associée à l'envoi d'un e-mail.

Un champ TID (Traffic Identifier) de l'en-tête 802.11 porte la mention de la classe de trafic affectée à la trame. Ces priorités ont été répertoriées en quatre catégories de classes 0 à 4 : 0 pour les services best-effort, correspondant à l'absence de priorité des flux associés, 1 pour les flux vidéo non interactifs, 2 pour les flux vidéo interactifs (vidéoconférence) et 3 pour les flux audio (téléphonie notamment).

Chacune de ces classes est gérée à l'aide de files d'attente qui possèdent un ensemble de paramètres définissant la manière dont l'accès au médium va se faire. On distingue essentiellement quatre paramètres caractérisant une file d'attente. En premier lieu, la variable AIFS (Arbitration Interframe Space) définit un temporisateur avant émission. Elle joue un rôle analogue au DIFS dans la fonction DFS, mais sa valeur est ici arbitrairement fixée pour chaque classe, ce qui en favorise certaines aux dépens d'autres. Sa valeur minimale est DIFS (*voir figure 7.14*). Les files d'attente sont d'autant plus prioritaires que la valeur de leur variable AIFS est faible.

Les deux variables suivantes sont définies par le couple de variables  $CW_{\min}$  et  $CW_{\max}$ , qui caractérise une taille de fenêtre de contention respectivement minimale et maximale. Le back-off time choisi dans l'algorithme de back-off est compris entre ces deux seuils. En cas de collision, la fenêtre de contention croît toujours selon l'algorithme de back-off en utilisant la même formule :

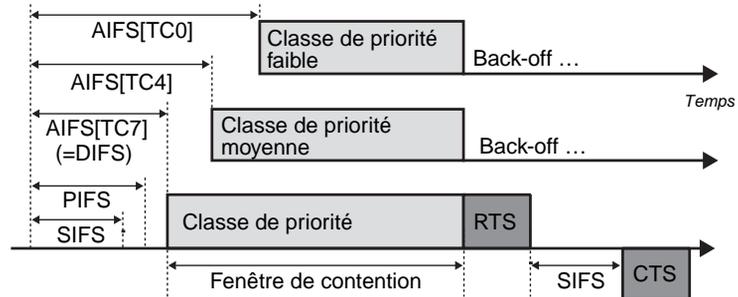
$$CW_{\text{nouveau}} = (CW_{\text{ancien}} + 1) \times FP[i] - 1$$

mais cette fois le facteur de persistance  $FP$  n'est plus fixé à la valeur 2 mais varie selon la priorité de la file d'attente numéro  $i$ . Plus ces deux seuils sont faibles, plus le back-off time est court, et plus la station est privilégiée.

Le quatrième paramètre est l'intervalle de transmission permis, appelé TxOP (Transmission Opportunity), qui définit pour chaque station un droit d'émettre ultérieurement. Cette variable définit le moment précis où une station est autorisée à émettre, ainsi que la durée maximale accordée pour l'émission. Selon ce concept, les stations ne sont plus équivalentes, même lors de l'émission de trames, certaines ayant un délai alloué plus long que d'autres. De surcroît, la contention sur le médium sans fil est effective au niveau du point d'accès qui distribue les valeurs des variables TxOP pour chaque station. Chaque file d'attente dispose ainsi d'un quadruplet de valeurs, instauré par défaut dans chaque station ou imposé par le point d'accès lors du polling, selon les choix d'implémentation.

Figure 7.14

Les AIFS dans la méthode EDCF



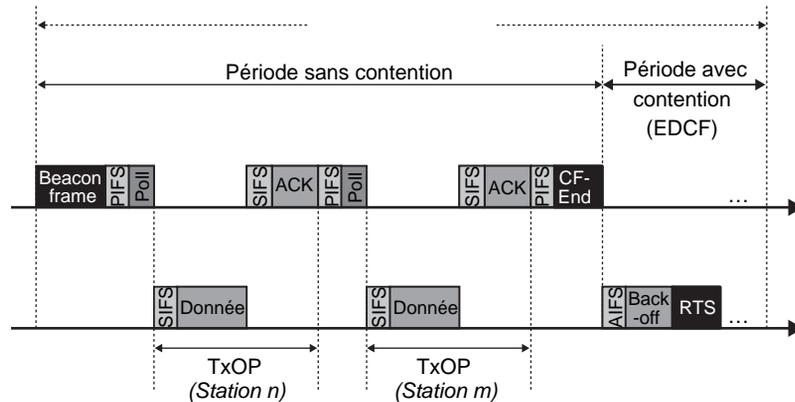
La méthode HCF (Hybrid Coordination Function), aussi appelée EPCF (Enhanced PCF), est une amélioration de PCF qui offre une gestion déterministe de l'accès au support sans collision. Le mécanisme est centralisé par un coordinateur, appelé HC (Hybrid Coordinator), en charge de l'allocation du support pour chaque station.

Comme pour PCF, le temps est divisé en supertrames composées chacune d'une période de contention et d'une période sans contention. Durant la période avec contention, c'est la méthode EDCF qui est utilisée. Durant la période sans contention, la station coordinatrice interroge les stations par polling.

La transmission des données s'effectue alors selon deux modèles : soit le coordinateur émet en transmettant les trames reçues aux stations concernées, soit il laisse les stations émettre, en leur distribuant des intervalles TxOP associés à la priorité qu'elles ont requise lors du polling. Autrement dit, le coordinateur contrôle l'accès au support et alloue le début et la durée d'émission des stations en fonction de la priorité de leurs flux. En outre, le coordinateur peut accéder au support en un temps SIFS (quel que soit le type des trames qu'il émet) inférieur au temps PIFS utilisé dans PCF, ce qui lui confère une priorité plus grande que n'importe quelle station de son BSS.

Le scénario fonctionnel de la méthode HCF est illustré à la figure 7.15.

Figure 7.15  
La méthode HCF



En complément d'EDCF et de HCF, le mécanisme DLP et la méthode d'accusés de réception cumulatifs ont été ajoutés à la norme 802.11e.

DLP (Direct Link Protocol), ou protocole de lien direct, consiste à envoyer des trames d'une station à une autre sans passer par le point d'accès. Cela permet de communiquer directement avec son interlocuteur sans transiter par un intermédiaire. Cette solution n'impose pas de passer dans un mode particulier, le mode *ad hoc*, et assure automatiquement la mise en relation des deux interlocuteurs, à conditions toutefois qu'ils se trouvent dans une même cellule. Le temps de transmission s'en trouve réduit d'autant et l'efficacité de l'utilisation de la bande passante optimisée.

La méthode d'accusés de réception cumulatifs (*burst acknowledgement*) est une technique optionnelle de la norme 802.11e qui améliore l'efficacité du canal en agrégeant plusieurs accusés de réception en un seul. L'émetteur n'est pas contraint de se mettre en attente de réception d'accusé de réception à chaque trame qu'il envoie, mais peut envoyer plusieurs trames de suite et attendre un accusé de réception global.

Malgré les potentialités de cette norme, elle est loin d'être parfaite et n'apporte qu'une priorité relative. Si deux stations risquent d'entrer en collision, elles vont être séparées par le mécanisme IEEE 802.11e, et la station la plus prioritaire passera avant la station non prioritaire. Si une première collision est évitée et que la station la moins prioritaire tire un long temporisateur, celui-ci peut arriver à échéance juste avant l'échéance d'un temporisateur beaucoup plus court d'une station hautement prioritaire qui aurait été enclenché bien après le temporisateur lent. Globalement, plus un point d'accès est chargé et plus ce risque est grand.

On peut en déduire que la norme IEEE 802.11e n'apporte qu'une amélioration relative et que la meilleure solution est encore de s'assurer que les points d'accès ne sont pas chargés du tout ou encore que le nombre de point d'accès est suffisant pour que les terminaux soient toujours connectés à la vitesse la plus haute et que le débit total soit très inférieur au débit maximal réel.

## En résumé

Pour les opérateurs comme pour les entreprises, le passage à la ToIP représente un coût relativement important, surtout s'il s'agit d'étendre la couverture aux réseaux sans fil. Mais ce ticket d'entrée peut s'avérer rentable et bénéfique si la technologie est à la hauteur de ce qu'elle remplace.

Simplicité, interopérabilité, sécurité, mobilité et surtout qualité de service sont les composants de base pour convaincre les industriels comme les utilisateurs de franchir le pas. Tels sont les défis actuels que la ToIP dans les réseaux sans fil doit relever.

Un standard unique s'imposerait plus facilement que l'addition de tous les standards proposés, la difficulté étant de préserver les promesses mises bout à bout de ces technologies. Une fois le standard approuvé, l'interopérabilité entre équipements à une large échelle imposerait aux constructeurs de respecter les mêmes normes d'implémentation.

Alors même que la ToIP n'est pas encore suffisamment déployée pour être pleinement concurrentielle, les opérateurs de téléphonie classique se livrent à une guerre des prix. Il est donc probable que les avantages financiers des solutions de ToIP paraîtront moins pertinents à l'avenir, d'autant que le ticket d'entrée reste conséquent.

Une fois alignée sur les autres tarifs, la ToIP devra se démarquer par la richesse de services à valeur ajoutée qui restent encore à inventer.

En constatant l'intérêt suscité par Wi-Fi aujourd'hui, on peut imaginer que le sans-fil devienne l'atout différenciateur pour l'émergence des technologies de ToIP. En cas de doute, la convergence des données vers un modèle tout-IP, pour la voix, la vidéo et les données à la fois, deviendra une évidence pour favoriser dans l'avenir l'essor de la ToIP sans fil.

## La téléphonie sur WiMax

L'initiative WiMax est née du désir de développer des liaisons hertziennes concurrentes des techniques xDSL terrestres. Après de longues années d'hésitation, son vrai démarrage a été favorisé par l'arrivée de la norme IEEE 802.16.

### WiMax fixe

Le groupe de travail 802.16 a mis en place des sous-groupes, qui se sont attaqués à des problèmes distincts.

Le groupe de travail de base a normalisé un accès métropolitain dans la bande des 10-66 GHz avec une vue directe des antennes entre elles et un protocole point-à-point. Finalisée en 2001, cette norme a été complétée en 2002 par la norme 802.16c, qui introduit des profils système WiMax, et par une partie de la norme 802.16d de 2004, qui apporte des correctifs et des fonctionnalités supplémentaires autorisant une stabilité de quelques années pour la norme WiMax.

La référence qui sert de base aux équipementiers est appelée IEEE 802.16 2004.

La norme 802.16e (voir figure 7.16) a pour objectif d'étendre WiMax à des machines terminales mobiles, impliquant la possibilité de réaliser des connexions xDSL vers des mobiles. Les fréquences utilisées se situeront entre 2 et 3 GHz.

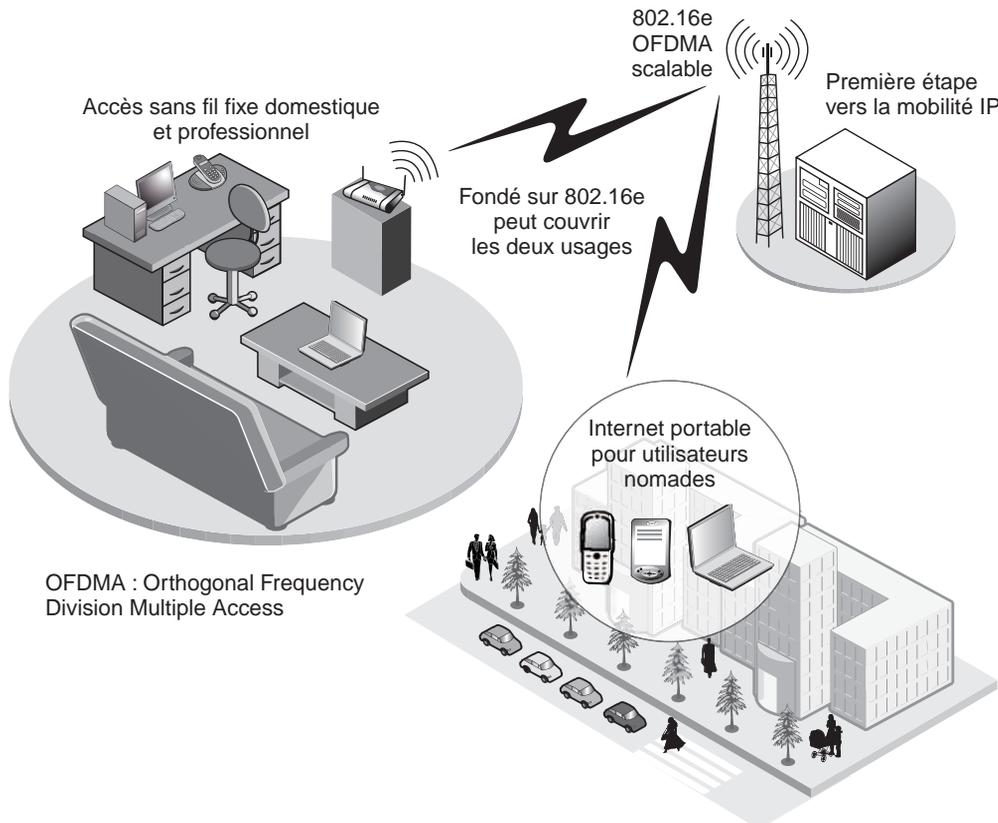


Figure 7.16

Réseau WiMax 802.16e

Les différences entre les normes 802.16 et 802.11 sont nombreuses. La couverture est beaucoup plus importante pour les premières, puisqu'elle peut dépasser 10 km, contre quelques dizaines à quelques centaines de mètres pour les secondes. La technologie 802.16 est moins sensible aux effets multitrajet et pénètre mieux à l'intérieur des bâtiments. Elle est de plus mieux conçue pour assurer le passage à l'échelle, ou scalabilité, sur de grandes surfaces, c'est-à-dire sur des cellules de plusieurs kilomètres carrés au lieu de plusieurs centaines de mètres carrés.

Pour un même canal de 20 MHz, WiMax permet de faire passer un peu plus de débit. La qualité de service est aussi plus facile à garantir. Les avantages de 802.11 par rapport à 802.16 résident essentiellement dans son prix de revient faible, une forte réutilisation et un large succès commercial.

## **WiMax-Mobile**

WiMax-Mobile, ou Wi-Mobile, ou encore Universal WiMax, correspond à la norme IEEE 802.16e. Son objectif est de concurrencer les normes de réseau de mobiles 3G, comme l'UMTS ou le CDMA 2000.

WiMax-Mobile s'adresse aux réseaux métropolitains sans fil mais adaptés aux connexions d'utilisateurs mobiles. Le standard WiMax-Mobile a été finalisé en décembre 2005, et les premiers équipements sont arrivés sur le marché dans le courant de l'année 2007. Le retard important par rapport aux normes déjà en place dans le monde des mobiles, comme l'UMTS, ne pourra être comblé que par un coût très inférieur de ces nouveaux équipements.

La raison du prix de revient potentiellement très bas de WiMax-Mobile est simple à expliquer. S'appuyant sur Ethernet et IP, ces réseaux seront totalement IP et gérés par les protocoles du monde IP, à commencer par les protocoles de gestion de la mobilité, tels que IP Mobile, voire IP cellulaire ou autre.

La sécurité des réseaux WiMax-Mobile sera globalement celle des réseaux sans fil. Le protocole IPsec sera le protocole de base pour la confidentialité des données transitant dans le réseau.

### **Performances des réseaux WiMax-Mobile**

Un réseau WiMax-Mobile donnera la possibilité de se connecter en se déplaçant jusqu'à des vitesses de 130 km/h avec gestion de la mobilité, c'est-à-dire avec handover pour passer d'une cellule à une autre. La taille des cellules pourrait être de l'ordre de 1 km.

Les antennes seront intelligentes. Elles se composeront d'une série d'antennes dont les signaux seront combinés de manière variable afin de contrôler la réception et la transmission. Une antenne principale sera accompagnée de plusieurs antennes secondaires sectorielles projetant un grand nombre de faisceaux étroits vers l'abonné. Les différents signaux émis ou reçus permettront de reconstituer de façon précise le signal original.

Une nouveauté à remarquer par rapport aux autres réseaux issus du groupe 802 de l'IEEE est la garantie de service incluse dans le protocole lui-même. Cette dernière permettra à un utilisateur se trouvant dans une cellule et bénéficiant d'un débit de 1 Mbit/s au moment de l'ouverture de sa connexion de maintenir ce débit pendant toute la durée de sa communication, indépendamment du nombre d'utilisateurs connectés. Dans les autres réseaux IEEE 802, au contraire, les ressources sont partagées entre les utilisateurs de manière statistique. Lorsqu'un grand nombre d'utilisateurs se partagent un point d'accès, chacun d'eux a un débit faible.

Dans un réseau WiMax-Mobile, un client peut se voir refuser l'accès à la borne de connexion s'il n'y a plus de ressources suffisantes dans le réseau, un peu à la manière du signal d'occupation dans le réseau téléphonique traditionnel. C'est là un changement d'orientation important dans la réflexion du groupe de travail IEEE 802.16.

### Classes de services WiMax pour la ToIP

Le réseau WiMax est doté de quatre classes de services pour la téléphonie, et le réseau WiMax-Mobile de cinq.

Les quatre classes de WiMax sont les suivantes :

- UGS (Unsolicited Grant Service), dévolu à la téléphonie grâce à une forte garantie de la qualité de service.
- rtPS (Real-time Packet Service), qui correspond à des applications ayant de fortes contraintes temporelles, mais avec des débits qui peuvent être variables.
- nrtPS (Non-real-time Packet Service), qui correspond à des applications sans contraintes temporelles mais avec des contraintes de débit.
- BE (best-effort), qui ne possède aucune garantie.

Un mécanisme dit de Request-Grant a été défini par le groupe IEEE 802.16 pour gérer la transmission montante. Les stations doivent en tout premier lieu être capables d'envoyer une demande de bande passante dans la trame en cours avant de pouvoir transmettre dans la trame suivante dans un slot réservé.

L'accès au support physique de WiMax est de type TDMA (Time Division Multiple Access). Il permet l'émission de trames successives, elles-mêmes découpées en tranches, comme l'illustre la figure 7.17.

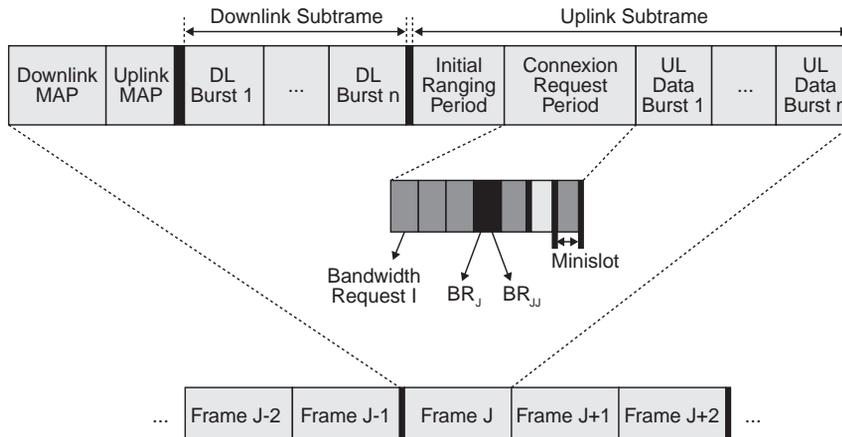


Figure 7.17

Format de la trame WiMax

La voie descendante ne pose pas de problème particulier puisque la station de base gère des transmissions de façon centralisée. Une voie de téléphonie est émise régulièrement dans des slots synchrones.

Le problème se complique avec la voie montante, puisque l'algorithme d'allocation de la bande passante doit être appliqué de façon distribuée. La trame est divisée en trois parties temporelles. La première, appelée Initial Ranging Period, est composée d'un nombre fixe de mini-slots. Ces mini-slots sont alloués aux stations par la station de base afin qu'elles puissent indiquer leur portée, la bande passante demandée et si elles ont des paquets prêts à être émis. La seconde, appelée Contention Request Period, est réservée aux demandes des stations qui n'ont pas de contrainte temporelle. La troisième et dernière est composée d'un ensemble de tranches de temps allouées aux différentes stations par la station de base à la suite des informations recueillies dans les deux parties précédentes.

Dans la norme IEEE 802.16, les applications sensibles au délai et les applications qui doivent éviter les collisions réclament de la bande passante dans les mini-slots de la première partie et transmettent dans les tranches de temps qui leur sont allouées dans la trame suivante. C'est le cas des flots de téléphonie sur IP. D'un autre côté, les applications qui présentent une tolérance à l'égard du délai et les applications qui peuvent perdre du temps dans des collisions éventuelles réclament de la bande passante par l'intermédiaire des mini-slots de la période de contention, la deuxième trame. Une fois leur demande réalisée avec succès, c'est-à-dire sans qu'une autre demande vienne entrer en collision avec la leur, une tranche de temps leur est affectée dans la trame suivante.

Dans la version Universal WiMax une cinquième classe de service est ajoutée pour la parole téléphonique compressée. Dans WiMax, la classe de service mise au point pour la téléphonie, appelée UGS (Unsolicited Grand Service), ne propose qu'un flux constant, de telle sorte que si le flux téléphonique est compressé et donne un débit variable, une perte de bande passante importante peut avoir lieu. La nouvelle classe s'appelle ertPS (enhanced real-time Packet Service).

## La sécurité

Plus qu'une évolution technologique, la téléphonie sur IP est aujourd'hui une application majeure du monde des réseaux. Son atout le plus important provient en partie de son intégration dans le réseau de données, qui rend son coût très compétitif.

Si les communications téléphoniques classiques sont généralement facturées à la durée et à la distance, les communications sur Internet sont forfaitaires. Il n'existe aucune contrainte de distance et aucune limite de temps. La facture est uniquement fondée sur le débit offert. Ainsi, toutes les applications dont les flux transitent sur le réseau Internet sont comprises dans le prix.

La technologie de ToIP est apparue il y a plus de dix ans. Elle a subi de multiples standardisations internationales, qui, sans la mettre à l'abri des évolutions permanentes, inhérentes aux technologies réseau, la rendent désormais suffisamment mature pour envisager un

déploiement à grande échelle. À condition toutefois de maîtriser la sécurité et son intégration au monde du sans-fil.

Les vulnérabilités dont les attaques peuvent tirer parti peuvent avoir cinq origines :

- les protocoles ;
- les logiciels ;
- le système d'exploitation ;
- l'infrastructure physique ;
- l'erreur humaine.

Chacune d'elles est une source potentielle de faille, qu'il convient d'étudier avec précaution dans la mise en place d'une solution de ToIP.

Une attaque peut avoir trois objectifs :

- Acquisition de service. L'objectif d'une telle attaque est de s'approprier des droits et fonctionnalités qui n'ont pas véritablement été attribués à l'attaquant.
- Interception de service. Cette attaque compromet la confidentialité du service et vise à en analyser ou modifier le contenu.
- Interruption de service. L'objectif est purement de nuire au bon déroulement du service en cherchant à le mettre hors d'usage.

La sécurité de l'application de téléphonie sur IP n'est pas vraiment différente de celle des autres applications du monde IP. Nous allons dans un premier temps examiner les attaques possibles.

## Les attaques

Les attaques de sécurité réseau sont divisées en attaques passives et actives. Ces deux classes sont elles-mêmes divisées en sous-classes. Les conséquences de ces attaques sont le coût légal et de recouvrement et la perte d'informations propriétaires, d'image ou de services réseau.

Une attaque est dite passive lorsqu'un individu non autorisé obtient un accès à une ressource sans modifier son contenu. Les attaques passives peuvent être des écoutes ou des analyses de trafic, parfois appelées analyses de flot de trafic.

Ces deux attaques passives présentent les caractéristiques suivantes :

- Écoute clandestine, ou *eavesdropping*. L'attaquant écoute les transmissions pour récupérer le contenu des messages. Par exemple, une personne écoute les transmissions sur un réseau LAN entre deux stations ou bien écoute les transmissions entre un téléphone sans fil et une station de base. Des outils tels que VoIPong, Vomit, Cain and Abel, Oreka ou Angst permettent de reconstituer des communications téléphoniques.
- Analyse de trafic, ou *sniffing*. L'attaquant obtient des informations en surveillant les transmissions pour détecter des formes ou des modèles classiques dans la communication.

Une quantité considérable d'information est contenue dans la syntaxe des flots de messages transitant entre des parties communicantes. Des programmes tels que Siprogue, Sipsak, SiVuS, Dsniff ou Wireshark permettent d'effectuer ce genre d'interception.

Une attaque est dite active lorsqu'un parti non autorisé apporte des modifications aux messages et flux de données ou de fichiers. Il est possible de détecter ce type d'attaque.

Les attaques actives peuvent prendre la forme d'un des quatre types suivants, seul ou en combinaison :

- Mascarade. L'attaquant usurpe l'identité d'un utilisateur autorisé et obtient ainsi certains privilèges d'accès. Des outils tels que Registration Hijacker, Registration Eraser, Registration Adder ou RedirectPoison permettent d'effectuer ces manipulations.
- Rejeu. L'attaquant surveille les transmissions (attaque passive) et retransmet les messages à un utilisateur légitime. Ce type d'attaque n'est pas possible dans le cas de la téléphonie puisque le rejeu n'est pas envisageable dans le cadre d'une application interactive. Le rejeu pourrait éventuellement s'appliquer au cours de l'authentification. Un programme tel que AuthTool permet de réaliser ce genre d'exploit.
- Modification de message. L'attaquant altère un message légitime en supprimant, ajoutant, modifiant ou réordonnant du contenu. Là encore la téléphonie sur IP n'est pas concernée directement, mais des programmes dédiés existent, tels RTP InsertSound ou RTP MixSound.
- Déni de service, ou DoS (Deny of Service). L'attaquant prévient ou interdit l'usage normal ou la gestion des moyens de communication. Par exemple, en envoyant des messages en masse, les ressources d'un équipement peuvent saturer. Des logiciels tels que UDP flooder, RTP flooder, INVITE flooder ou IAX flooder sont conçus à des fins d'audit, avec les risques d'utilisation que cela suppose.

Ce dernier type d'attaque est une source de menace redoutable pour les solutions de sécurité logicielle, puisque la sécurité est facilement mise en cause en cas de modification malveillante des programmes chargés d'appliquer les protocoles et les règles de contrôle.

L'attaque par déni de service est une des plus simples à mettre en œuvre et est généralement très difficile à parer dans les réseaux sans fil. Le déni de service est obtenu lorsque l'élément que l'on attaque est submergé de messages et ne peut répondre à la demande. Dans le cas classique, les pirates occupent un grand nombre de postes de travail et leur font émettre des flots ininterrompus de messages qui convergent vers l'élément attaqué. La parade est difficile puisque l'attaque peut être soudaine et qu'il n'est pas évident de prévoir cette convergence.

Dans un réseau sans fil, un déni de service consiste à émettre un grand nombre de requêtes d'attachement vers le point d'accès jusqu'à le faire tomber. Il est pour le moment impossible d'empêcher un utilisateur d'émettre ce flot de requête, même s'il n'est pas autorisé à se connecter. À chaque requête, le point d'accès doit exécuter une suite

d'instructions avant d'effectuer le refus. La seule parade connue consiste à déterminer le point d'où provient l'attaque et à lancer une intervention humaine de neutralisation.

De nombreuses attaques de déni de service peuvent s'effectuer par l'intermédiaire du protocole ICMP (Internet Control Message Protocol). Ce protocole est utilisé par les routeurs pour transmettre des messages de supervision permettant, par exemple, d'indiquer à un utilisateur la raison d'un problème. Une attaque par déni de service contre un serveur consiste à générer des messages ICMP en grande quantité et à les envoyer au serveur à partir d'un nombre de sites important.

Pour inonder un serveur, le moyen le plus simple est de lui envoyer des messages de type ping lui demandant de renvoyer une réponse. On peut également inonder un serveur par des messages de contrôle ICMP d'autres types.

Dans l'attaque par cheval de Troie, le pirate introduit dans la station terminale un programme qui permet de mémoriser le login et le mot de passe. Ces informations sont envoyées vers l'extérieur par un message destiné à une boîte à lettres anonyme. Diverses techniques peuvent être utilisées pour cela, allant d'un programme qui remplace le gestionnaire de login, jusqu'à un programme pirate qui espionne ce qui se passe dans le terminal.

Ce type d'attaque est assez classique dans les réseaux sans fil puisqu'un client peut s'immiscer, *via* le point d'accès, dans un PC et y installer un logiciel espion lui permettant de prendre la place de l'utilisateur.

Beaucoup de mots de passe étant choisis dans le dictionnaire, il est très simple pour un automate de les essayer tous. De nombreuses expériences ont démontré la simplicité de cette attaque et ont mesuré que la découverte de la moitié des mots de passe des employés d'une grande entreprise pouvait s'effectuer en moins de deux heures.

Une solution simple pour remédier à cette attaque consiste à complexifier les mots de passe en leur ajoutant des lettres majuscules, des chiffres et des signes comme !, ?, &, etc.

L'attaque par dictionnaire est l'une des plus fréquentes dans les réseaux sans fil qui ne sont protégés que par des mots de passe utilisateur.

Il est possible de compromettre une communication en diffusant de faux messages RTP, conçus par un attaquant comme s'ils faisaient partie d'une communication mais qui perturbent en réalité la réception du message audio pour le rendre incompréhensible.

En considérant le protocole de signalisation SIP, on peut imaginer des attaques à la fois par des requêtes et par des réponses, notamment les suivants :

- Un message BYE correctement forgé par un attaquant peut mettre fin à la communication d'autres personnes.
- Un message REGISTER contenant un enregistrement d'un utilisateur fictif est envoyé en masse vers un serveur d'enregistrement afin de saturer les capacités de stockage de ce dernier ou de le surcharger.

- Un message REGISTER contenant un enregistrement d'un utilisateur réel vers une localisation inexacte risque de détourner la communication vers la localisation mentionnée.

Des attaques similaires sont envisageables avec les autres protocoles de signalisation.

## Les sécurités à mettre en place

La sécurité dans les réseaux téléphoniques classiques est particulièrement forte. La disponibilité du réseau y atteint les 5 « neuf », c'est-à-dire que le système marche 99,999 % du temps. Avec la téléphonie sur IP, il faut introduire des éléments de sécurité supplémentaires puisque le support est partagé et qu'une écoute peut se produire.

Classiquement, la sécurité s'appuie sur cinq services de base : l'identification, l'authentification, la confidentialité, l'intégrité des données et la non-répudiation.

L'utilisateur d'un système ou de ressources quelconques possède une identité, sorte de clé primaire d'une base de données, qui détermine ses lettres de crédits (*credentials*) et autorisations d'usage. Cette identité peut être vérifiée de multiples manières, par exemple par la saisie d'un compte utilisateur (login) ou au moyen de techniques biométriques, telles que les empreintes digitales ou vocales, les schémas rétiniens, etc.

L'authentification a pour objectif de vérifier l'identité des personnes en communication ou des processus remplaçant ces personnes. Plusieurs solutions simples sont mises en œuvre pour cela, comme l'utilisation d'un identificateur (login) et d'un mot de passe (password), d'une méthode de défi fondée sur une fonction cryptographique et d'un secret partagé. L'authentification peut s'effectuer par un numéro d'identification personnel, comme le numéro inscrit dans une carte à puce, ou code PIN (Personal Identification Number).

Des techniques d'authentification beaucoup plus sophistiquées, comme la vérification d'une identité par les empreintes digitales ou rétiniennes, se sont développées de façon industrielle au début des années 2000. Les mécanismes permettant cette vérification sont toutefois assez complexes et ne permettent cette solution pour vérifier l'identité d'un individu que dans des contextes particuliers.

L'authentification peut être simple ou mutuelle. Dans le cadre d'une communication téléphonique, une authentification mutuelle est conseillée. Elle consiste essentiellement à comparer les données provenant de l'utilisateur qui se connecte à des informations stockées dans un site protégé. Les attaques sur les sites mémorisant les mots de passe représentent une part importante du piratage.

La confidentialité désigne la garantie que les données échangées, les paroles téléphoniques, ne sont compréhensibles que par les deux entités qui partagent un même secret. Cette propriété implique la mise en œuvre d'algorithmes de chiffrement en mode flux, c'est-à-dire octet par octet, ou en mode bloc, par exemple par série de 8 octets.

Les algorithmes de chiffrement permettent de transformer un message écrit en clair en un message chiffré, appelé cryptogramme. Cette transformation se fonde sur une ou plusieurs clés. Le chiffrement le plus simple est celui où une clé unique et secrète est partagée par les seuls émetteur et récepteur.

Les systèmes à clés secrètes sont caractérisés par une transformation  $f$  et une transformation inverse  $f^{-1}$ , qui s'effectuent à l'aide de la même clé. C'est la raison pour laquelle on appelle ce système « à chiffrement symétrique ».

Les algorithmes de chiffrement à clé publique sont asymétriques. Le destinataire est le seul à connaître la clé de déchiffrement. La sécurité s'en trouve accrue puisque même l'émetteur ne connaît pas cette clé. L'algorithme le plus classique et le plus utilisé est RSA (Rivest, Shamir, Adleman), qui utilise la quasi-impossibilité d'effectuer la fonction d'inversion d'une fonction puissance.

### **Les infrastructures de sécurité**

Des mécanismes tels que la confidentialité ou l'intégrité des données peuvent être supportés à différents niveaux de l'architecture et sur les différents tronçons, ou arcs, qui composent le réseau. La gestion des clés de chiffrement peut être, par exemple, réalisée manuellement.

L'identification, l'authentification, la non-répudiation et les autorisations sont des procédures mises en œuvre dans le réseau d'accès, le réseau de transport IP et le réseau de destination, un intranet, par exemple. Ces services peuvent également être offerts au niveau applicatif.

Schématiquement, les infrastructures de sécurité des réseaux peuvent être classées en cinq catégories :

- Chiffrement au niveau physique. Dans la cryptographie optique (PMD), le saut de fréquences pseudo-aléatoire ou le chiffrement du flux d'octets (une méthode couramment déployée par les banques), les clés sont distribuées manuellement.
- Confidentialité, intégrité de données, signature de trames MAC. La distribution des clés est réalisée dans un plan particulier, décrit par la norme IEEE 802.1x. Dans ce cas, on introduit la notion de contrôle d'accès au réseau LAN. C'est une notion juridique importante, dont le rôle est d'interdire le transport des informations à des individus non authentifiés, et donc potentiellement dangereux.
- Confidentialité, intégrité des données, signature des paquets IP ou TCP. C'est typiquement la technologie IPsec en mode tunnel. Un paquet IP chiffré et signé est encapsulé dans un paquet IP non protégé. En effet, le routage à travers Internet implique l'analyse de l'en-tête IP par les passerelles traversées. IPsec crée un tunnel sécurisé entre le réseau d'accès et le domaine du fournisseur de services. On peut déployer une gestion manuelle des clés ou des protocoles de distribution automatisés, tels que ISAKMP (Internet Security Association and Key Management Protocol). La philosophie de ce protocole s'appuie sur la libre utilisation du réseau d'accès, qui ne va pas sans soulever des problèmes juridiques. Par exemple, si des utilisateurs mal intentionnés protègent leurs échanges téléphoniques, il est impossible aux réseaux traversés de détecter leur complicité dans le transport d'informations illégales.

- Insertion d'une couche de sécurité additive. Le protocole SSL (Secure Sockets Layer) fondé sur un chiffrement asymétrique assure la protection d'applications telles que la navigation Web ou la téléphonie IP. SSL réalise généralement une simple authentification entre serveur et client et négocie un secret partagé (Master Secret), à partir duquel sont dérivées les clés de chiffrement utilisées par l'algorithme de chiffrement négocié entre les deux parties. Une fois le tunnel sécurisé établi, le client s'authentifie à l'aide d'un login et d'un mot de passe. Il obtient alors une identité temporaire associée à un simple cookie.

### **La sécurité dans la téléphonie par Wi-Fi**

Pour sécuriser une communication téléphonique dans un réseau sans fil, il faut doter l'environnement d'un certain nombre de fonctions, qui peuvent être prises en charge soit par l'infrastructure achetée pour réaliser le réseau lui-même, soit par de nouveaux éléments de réseau à ajouter.

De façon plus précise, il faut intervenir auprès de quatre grands types d'éléments d'infrastructure, l'infrastructure qui permet l'authentification des clients et des équipements de réseau, le matériel et le logiciel nécessaires pour réaliser la sécurité sur l'interface radio, les éléments de réseau nécessaires pour filtrer les paquets et détecter les attaques et enfin les machines nécessaires pour gérer les accès distants lorsque les utilisateurs se déplacent :

- Infrastructure d'authentification. La norme IEEE 802.1x recommande l'usage de serveurs RADIUS (Remote Authentication Dial-In User Server). L'authentification peut être réalisée par un serveur situé dans le domaine visité ou à l'extérieur de ce dernier. Cette architecture établit un cercle de confiance, grâce auquel un message d'authentification est relayé par plusieurs serveurs liés les uns aux autres par des associations de sécurité.
- Sécurité radio. La sécurité radio vise à assurer la confidentialité, l'intégrité et la signature des paquets. Ces services sont délivrés par des protocoles tels que WEP (Wired Equivalent Privacy), TKIP (Temporal Key Integrity Protocol) ou CCMP (Counter with CBC MAC Protocol), normalisés par le comité IEEE 802. Ils utilisent des clés déduites d'une clé maître au terme de la procédure d'authentification.
- Filtrage des paquets. La fiabilité de cette opération repose sur la signature des paquets à l'aide des clés déduites de l'authentification. Grâce à ce mécanisme, les trames qui pénètrent dans le système de distribution sont sûres (pas de risque de *spoofing*). Des systèmes de filtrage (point d'accès ou portail) gèrent les privilèges des paquets IP (destruction des paquets illicites) et permettent de réaliser et de facturer des services de QoS (Quality of Service).
- Accès aux services distants (roaming). L'accès à des services distants peut être désigné de façon générique sous l'appellation de services VPN (Virtual Private Network). Par exemple, on peut mettre en œuvre des liens sécurisés interdomaines à l'aide des protocoles IPsec ou SSL.

## Conclusion

Dans ce chapitre nous avons introduit les différentes architectures de réseaux qui peuvent être utilisées pour transporter de la téléphonie sur IP.

Nous nous sommes intéressés aux réseaux terrestres, en commençant par les réseaux d'entreprise de type Ethernet puis les réseaux d'opérateur de type Relais de trames et ATM. Ces réseaux doivent être bâtis pour introduire de la qualité de service.

Nous avons détaillé les environnements hertziens et décrit des solutions à déployer pour réaliser de la ToIP dans les réseaux Wi-Fi et WiMax. Il est à noter que réaliser de la téléphonie sur Wi-Fi est plus complexe que sur WiMax, qui possède des classes de service permettant de prendre en charge directement la téléphonie sur IP.

En fin de chapitre, nous avons abordé la sécurité de la ToIP. En fait, la sécurité de la ToIP n'est pas vraiment différente de la sécurité des applications Internet en général. L'aspect temps réel est évidemment important puisqu'un déni de service, par exemple, ne permet plus le temps réel et fait tomber l'application de téléphonie plus facilement qu'une autre application.

Globalement, on peut conclure que la ToIP est une application difficile à mettre en place et que les architectures nécessaires demandent à être bien conçues pour que la qualité de service soit obtenue à un coût très bas.

Le succès des réseaux DiffServ en entreprise en est une parfaite illustration. En utilisant des routeurs DiffServ, pas tellement plus chers que des routeurs classiques, il est assez simple d'obtenir de la qualité dans un environnement d'entreprise.

En ce qui concerne les réseaux d'opérateurs, la solution statistique apportée par DiffServ n'est plus forcément acceptable. Si tous les clients se mettent à téléphoner à la suite d'un sinistre important, les flots de paquets de parole peuvent devenir trop importants par rapport à l'infrastructure prévue. Ce que l'on peut faire dans une entreprise, dans laquelle le nombre d'employés est limité, ne peut être reproduit lorsque plusieurs centaines de milliers, voire de millions d'utilisateurs peuvent se connecter simultanément. Dans ce cas, nous avons vu que les solutions consistaient à utiliser le relais de trames ou la version AAL2 de l'ATM, qui permettent d'optimiser les ressources du réseau.

## Partie II

# Pratique de la ToIP

De la théorie à la pratique, il y a parfois des écarts considérables. L'objectif de cette partie est de proposer un panorama assez large des applications pratiques de la téléphonie sur IP.

Les quatre premiers chapitres s'intéressent aux softphones grand public. Parce qu'il nous paraissait difficile de les ignorer, compte tenu de leur succès, nous avons fait le choix d'analyser les tendances actuelles du marché, même si, bien souvent, elles ne se conforment pas au modèle théorique exposé dans la première partie. Car souvent, des acteurs de renom, choisissent d'utiliser une technologie propriétaire. Le logiciel ainsi conçu s'appuie sur la notoriété de ces acteurs pour isoler ses clients au sein de son réseau, à l'exclusion de tout autre réseau concurrent.

Les softphones que nous présentons ne sont pas forcément dédiés à la téléphonie et ont parfois davantage vocation à servir de messagerie instantanée que de téléphone. Mais la convergence des données vers un réseau entièrement fondé sur le protocole IP s'accélère, et les outils de communication tendent à s'uniformiser pour s'imposer. Dans la mesure où les softphones sont les reflets de cette tendance, nous avons privilégié une présentation générale de cette thématique, en nous penchant sur des softphones bien connus afin de les analyser, même si la ToIP n'est parfois que le pendant de ces logiciels.

Le chapitre 8 introduit la problématique des softphones de manière générale. Nous détaillons les services que les softphones sont en mesure de fournir en plus de la ToIP et évoquons quelques offres du marché.

Le chapitre 9 explore les caractéristiques de Skype, le softphone grand public le plus célèbre. Nous essayons de comprendre les raisons de ce phénomène, qui a été le premier à percer véritablement, générant un vif intérêt du public et une émulation importante parmi les principaux acteurs d'Internet.

Le chapitre 10 se penche sur les caractéristiques des softphones WLM de Microsoft et Yahoo! Messenger de Yahoo!. Conscients du potentiel des services de téléphonie, Microsoft comme Yahoo! ont tout naturellement fait évoluer leur logiciel de messagerie instantanée en softphone. Nous détaillons les principales fonctions de ces logiciels.

Le chapitre 11 est consacré à la plate-forme Jabber et au softphone GTalk de Google. Là encore, c'est la messagerie qui sert d'appui au service de téléphonie. Nous détaillons les caractéristiques principales de la plate-forme et montrons comment cette

approche, en suivant la normalisation et en cherchant à la faire évoluer, se distingue remarquablement de ces concurrentes.

Les deux chapitres suivants analysent des aspects annexes de la téléphonie sur IP, qui sont des champs d'application prépondérants aujourd'hui.

Le chapitre 12 présente un PBX d'un genre nouveau, Asterisk, solution logicielle gratuite (libre sous licence GPL), ouverte et innovante. Ce programme constitue une solution de rechange aux traditionnels PABX physiques d'entreprise, lesquels sont souvent des entités particulièrement fermées, complexes et onéreuses. Asterisk fournit un niveau de service et de fiabilité comparable, sans investissement préalable. Nous montrons comment ce logiciel peut révolutionner la téléphonie d'entreprise et pourquoi il peut aussi s'introduire chez les particuliers.

Le chapitre 13 explique comment les opérateurs de ToIP, ainsi que les FAI, qui se positionnent de plus en plus en opérateurs téléphoniques, peuvent offrir leurs services de téléphonie à leurs clients. Nous indiquons les différents réseaux d'accès déployés et leurs caractéristiques principales.

Le dernier chapitre de cette partie se penche sur une difficulté d'application de la ToIP dans les réseaux : l'adaptabilité des flux conjointement à d'autres fonctionnalités.

Le chapitre 14 examine un problème délicat et récurrent de la ToIP, qui réside dans la traversée des réseaux déployant du NAT ou se situant derrière un pare-feu. Ces deux éléments d'usage relativement courants posent problème, et nous indiquons les techniques et protocoles qui permettent de les contourner.

## La ToIP sur softphone

---

Le temps est-il révolu où l'on décrochait son téléphone pour appeler ? Avec un ordinateur connecté à Internet, il suffit d'un microcasque pour téléphoner à moindre coût, parfois même gratuitement, avec une qualité audio potentiellement supérieure à celle de la téléphonie traditionnelle et en disposant de fonctionnalités infiniment plus évoluées. C'est devenu possible grâce aux logiciels de téléphonie sur IP, appelés softphones. Les plus grands éditeurs de logiciels se livrent une bataille acharnée pour imposer le leur.

Bien des personnes sont toutefois réticentes à l'idée d'utiliser les fonctionnalités évoluées de leur ordinateur, méfiantes face aux mystères de l'informatique et parfois craintives devant les aléas de leur connexion Internet. Pour téléphoner par le biais d'un softphone, il faut d'abord allumer son ordinateur, vérifier que la connexion Internet est fonctionnelle, lancer le softphone et enfin composer le numéro de son correspondant. Mieux vaut ne pas être pressé pour communiquer de cette façon ! À première vue, on n'y trouve ni la modernité technologique, ni le confort d'utilisation, ni même l'intérêt.

La différence est qu'ici la communication n'est pas exactement celle que l'on connaît. Il s'agit d'un nouvel usage, plus riche et fondamentalement différent, de la téléphonie. Ce nouvel usage s'appuie sur le constat que le développement des technologies réseau et des offres des fournisseurs d'accès accroît sans cesse le nombre de personnes connectées en permanence, ou presque, au réseau Internet. Cette masse déjà conséquente d'utilisateurs est de plus en plus rodée aux technologies réseau. À partir du moment où les internautes disposent d'une connexion à Internet permanente, il n'est pas plus difficile de composer un numéro de téléphone sur un ordinateur que sur un téléphone.

Avec les nouveaux services qu'offre le logiciel de téléphonie, il devient possible d'ajouter de la vidéo à la voix, d'établir des conférences avec plus de deux intervenants, d'envoyer des photos de vacances ou des documents de travail en même temps que l'on parle à son interlocuteur, de gérer un répertoire virtuel avec plusieurs dizaines, voire

centaines de contacts. Le tout gratuitement ou à moindre coût, sur un unique support, l'ordinateur, et avec un logiciel convivial, qui permet de communiquer indifféremment avec d'autres ordinateurs ou avec des téléphones traditionnels. Le marché potentiel de ce nouvel usage est gigantesque.

Parmi les prétendants au titre, Skype a lancé la marche et s'est rapidement bâti une renommée de leader. Son succès a ouvert les portes à la concurrence. Sur les traces du fameux logiciel, des sociétés parmi les plus importantes d'Internet ont lancé leur propre logiciel de téléphonie sur softphone, aux fonctionnalités très analogues, tels Google, Yahoo! et Microsoft.

Ce chapitre propose un tour d'horizon des principales offres existantes et à venir.

## Introduction aux softphones

La téléphonie sur softphone présente des avantages indéniables sur la téléphonie par ADSL, et ce pour les raisons suivantes :

- En moyenne, moins d'un tiers des abonnés ADSL disposent de la téléphonie dans leur forfait. Compte tenu de la progression du dégroupage, ce nombre devrait augmenter, mais ceux qui n'en bénéficient pas resteront tout de même majoritaires et devraient être intéressés par les services des softphones.
- Propre au réseau Internet, le service de présence permet aux utilisateurs de connaître en permanence la disponibilité de leur interlocuteur.
- Avec l'abonnement ADSL, une fois le fournisseur d'accès choisi, il est bien difficile d'en changer, tant les délais et les coûts sont importants. Avec les softphones, par ailleurs souvent moins chers que l'ADSL, on ne rencontre pas de tels problèmes.
- Tous les médias, audio, vidéo, textes et applicatifs, peuvent transiter sur un même support, alors que la visiophonie reste marginale dans le réseau téléphonique traditionnel.
- L'utilisateur devient nomade puisque les connexions IP sont facilement disponibles partout dans le monde, et aux mêmes coûts.

## Les services proposés

Les logiciels de téléphonie sur IP visent avant tout à se démarquer des autres offres de téléphonie en prenant appui sur des services ayant déjà séduit les internautes. Leur principale cible est la messagerie instantanée, un service à la mode, qui s'est d'abord imposé chez les plus jeunes, avant de s'étendre à une utilisation beaucoup plus large.

La téléphonie grand public n'a pas toujours été le cœur de cible des acteurs du monde des softphones. Windows Live Messenger, de Microsoft, et le logiciel de Yahoo! n'avaient pas pour vocation initiale de faire de la téléphonie sur IP, mais simplement de la messagerie instantanée. À l'inverse, Skype se cantonnait à la téléphonie sur IP. Aujourd'hui,

tous ces acteurs proposent des fonctionnalités complètes, qui englobent la téléphonie au travers de services complémentaires.

La plupart de ces logiciels proposent des solutions pour téléphoner sur Internet, mais aussi pour appeler des téléphones fixes ou des mobiles, faire de la vidéoconférence, partager des fichiers et de la musique et utiliser une messagerie instantanée. Globalement, la communication s'enrichit et offre de multiples facettes complémentaires. C'est ce qu'on appelle la convergence des services.

## La téléphonie

Le débit requis pour la téléphonie sur IP est très faible pour une qualité audio satisfaisante. En outre, les codecs utilisés par les logiciels de téléphonie sont souvent meilleurs que ceux utilisés dans la téléphonie traditionnelle.

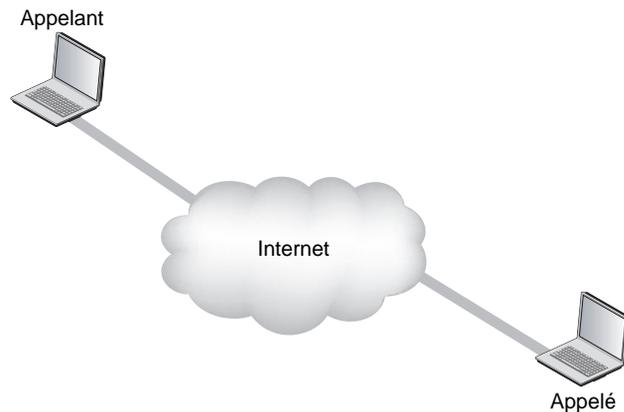
Le service de téléphonie se décline sous deux formes. Une offre gratuite, purement Internet, et une offre payante, qui permet la convergence avec les réseaux téléphoniques traditionnels.

### L'offre gratuite

L'offre gratuite ne permet aux utilisateurs du logiciel que de communiquer entre eux. Il s'agit d'une communication purement IP, qui sollicite exclusivement le réseau Internet, comme l'illustre la figure 8.1.

**Figure 8.1**

*Le modèle de téléphonie tout-IP*



Ce service est systématiquement proposé gratuitement dans toutes les offres. La gratuité est justifiée puisque le coût de connexion au réseau IP est directement imputable à l'internaute, lequel s'acquitte d'une somme le plus souvent forfaitaire pour disposer d'une connexion Internet. La communication ne requiert donc pas de surcoût en soi.

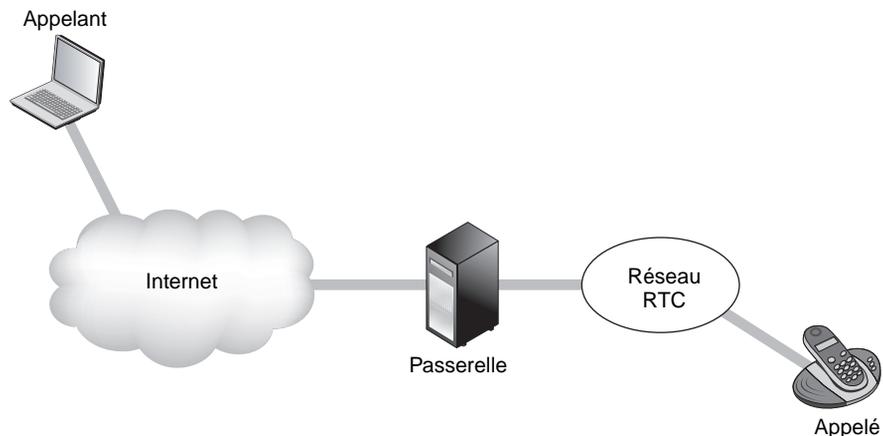
### L'offre payante

L'offre payante met en relation un utilisateur utilisant un softphone avec un utilisateur du réseau de téléphonie traditionnel. Les utilisateurs de softphone ne sont donc pas confinés à des communications entre internautes mais encouragés à acheter des crédits leur permettant d'appeler les téléphones traditionnels.

Ce sont donc à la fois le réseau Internet et le réseau téléphonique commuté qui sont utilisés dans ce mode de communication. Le service n'est plus gratuit puisque le réseau RTC est généralement facturé à l'usage et non forfaitairement. Grâce à ces crédits, les utilisateurs peuvent communiquer partout dans le monde, à des tarifs très avantageux, une bonne partie de la communication transitant sur le réseau IP, y compris la partie qui relie l'abonné appelant à son opérateur (voir figure 8.2).

Figure 8.2

Le modèle de téléphonie IP/RTC



Pour rendre la communication possible, une passerelle est nécessaire afin de transformer les flux IP en flux RTC, et réciproquement, car les protocoles de signalisation et de transport ne sont pas les mêmes dans les deux mondes.

L'appelant exploite le réseau IP de son fournisseur d'accès, puis passe par la passerelle pour communiquer avec des utilisateurs du réseau téléphonique RTC. Plus cette passerelle est proche de l'appelé, moins la communication est onéreuse.

La liaison entre l'appelant et l'appelé ne peut être réalisée pratiquement que sur Internet, sauf pour la liaison terminale avec l'appelé, c'est-à-dire les dernières centaines de mètres qui le relient au commutateur RTC le plus proche. Cette liaison est la seule facturation réelle à considérer, car en utilisant le réseau IP prioritairement, il n'est plus nécessaire de sous-louer aux opérateurs de téléphonie leur réseau pour transporter les flux de parole téléphonique, ce qui allège considérablement la facturation des appels.

Globalement, la facturation se fait soit au forfait, incluant un grand nombre de pays sans limitation de durée, soit, selon le pays de la personne appelée, à des tarifs variables en

fonction des solutions utilisées, mais souvent moins chers que ceux proposés par les opérateurs de téléphonie ou les fournisseurs d'accès à Internet.

### **Liste de contacts, présence et disponibilité**

Afin de simplifier la gestion des contacts, les logiciels de ToIP offrent la possibilité de lister l'ensemble de ses contacts, c'est-à-dire d'avoir un répertoire organisé en groupes de personnes (amis, famille, relations professionnelles, etc.). Cela permet de disposer d'un répertoire numérique, comportant numéros de téléphones portables et fixes, professionnels et personnels, adresses de messagerie, sites Internet, blogs, etc.

Cette liste simplifie les appels puisqu'il n'est pas nécessaire de connaître les identifiants ou les numéros d'appel d'un contact. Tous sont virtuellement visibles et joignables par simple clic à partir de leur véritable nom ou du pseudonyme qu'ils se donnent.

À cette liste est associé un indicateur de présence et de statut, qui permet d'informer les autres contacts de la disponibilité de chaque interlocuteur. Cet indicateur permet de savoir si un correspondant est joignable, c'est-à-dire s'il a connecté son logiciel de téléphonie au réseau, et s'il est disponible ou occupé.

Le statut peut être géré manuellement ou automatiquement. Il est géré manuellement pour indiquer explicitement qu'une activité est en cours chez le correspondant. Par exemple, un utilisateur informe ses contacts qu'il est au téléphone et ne souhaite pas être contacté. Il peut aussi être géré automatiquement. Par exemple, si le logiciel ne relève aucune activité sur le poste de travail d'un utilisateur, il peut modifier son statut pour notifier cette inactivité. Les gestions manuelles et automatiques peuvent se combiner.

### **Messagerie instantanée**

La messagerie instantanée, aussi appelée IM (Instant Messaging) ou IMP (Instant Messaging and Presence) ou plus couramment *chat*, est un mode de communication textuel et interactif qui connaît un énorme succès. C'est un peu l'art de discuter avec plusieurs personnes à la fois sur des sujets d'importance variable, tout en pouvant effectuer une autre activité en parallèle, comme écouter de la musique ou passer un appel téléphonique.

Le premier système de messagerie a été l'IRC (Internet Chat Relay). Conçu par Jarkko Oikarinen et Darren Reed, il a été standardisé en mai 1993 par la RFC 1459, remplacée ensuite par les RFC 2810 à 2813. Depuis, des efforts considérables ont été apportés à l'ergonomie et aux services. L'IRC n'en reste pas moins toujours aussi populaire aujourd'hui, en parallèle des systèmes de messagerie instantanée.

La messagerie instantanée n'a en elle-même aucun rapport avec la téléphonie sur Internet. C'est toutefois un outil complémentaire en ce qu'elle permet une certaine réactivité, voire interactivité, avec l'avantage de communiquer de manière moins intrusive que la téléphonie.

## Le SPAM

On note cependant un problème persistant avec la messagerie instantanée, que partagent les différents softphones : le SPAM, c'est-à-dire des messages non sollicités pouvant contenir des virus. Par le biais du réseau IP, le monde entier devient joignable à moindre coût, ce qui incite les spammeurs à utiliser ce canal.

On distingue deux types de SPAM :

- Le Call Spam, qui désigne des tentatives d'appels en masse. Lorsque le correspondant accepte l'appel, le spammer lui diffuse immédiatement un message audio, à caractère commercial le plus souvent. C'est l'équivalent du télémarketing massif.
- Le SPIM (SPam over Instant Messaging), c'est-à-dire le Spam par messagerie instantanée. Dans ce type de SPAM, une publicité est envoyée soit par messagerie instantanée, soit en annonçant un sujet pour un prétendu appel. Dans ce dernier cas, lorsque l'appelé lit le sujet de l'appel, le message publicitaire est passé, que l'appelé réponde ou non à l'appel. C'est l'équivalent du Spam par e-mail.

Lorsqu'on sait qu'en utilisant Skype, une simple recherche sur un prénom renvoie la liste des personnes qui portent ce prénom et leur adresse e-mail associée, on comprend pourquoi le SPIM peut se répandre aussi facilement et rapidement dans les réseaux. C'est l'un des risques inhérents à la facilité de mise en relation, prônée par tous les softphones.

Les parades possibles consistent à limiter les contacts en définissant une liste de contacts prohibés au fur et à mesure que des spammeurs sont détectés (liste noire) ou en utilisant des listes de contacts autorisés (liste blanche). Un filtrage des contenus de flux s'avère difficilement praticable, en particulier pour le Call Spam.

## Vidéo et transfert de fichiers

Les softphones permettent de superposer au son téléphonique une image capturée par une webcam. Les communications sont ainsi plus riches et se prêtent parfaitement à un cadre familial.

Le transfert de fichiers s'effectue sans passer par des serveurs centralisés, mais directement d'un poste à l'autre. Cela permet de ne solliciter que les machines qui participent effectivement au transfert, et ainsi de ne pas saturer les serveurs centraux.

Dans ce mode de communication, l'interactivité est perdue, contrairement à la téléphonie, la vidéo ou la messagerie instantanée. Mais cette possibilité se combine souvent aux précédentes pour compléter et enrichir une communication avec différents documents, qu'il s'agisse de documents de travail ou de fichiers multimédias.

Le transfert de fichiers se fait de manière non bloquante, en tâche de fond, laissant la place à toute autre forme de communication, messagerie instantanée ou communication audio/vidéo.

## Les softphones en entreprise

Bien souvent, les installations de softphones se font par les utilisateurs eux-mêmes, sans contrôle des administrateurs réseau de l'entreprise, ce qui ne va pas sans poser problème. D'autant que ces derniers sont parfois dépourvus des moyens d'assurer une gestion efficace de ces logiciels. Skype comme WLM utilisent des protocoles propriétaires difficiles à filtrer et qui contournent les politiques de sécurité de l'entreprise en se faisant passer pour des flux licites.

Les risques sont bien réels, à commencer par les attaques par virus ou chevaux de Troie. Il n'en reste pas moins vrai que tous les logiciels sont potentiellement sujets à ces infections. Ce qui pose problème ici, c'est surtout l'opacité des protocoles utilisés et la volonté de s'intégrer dans un réseau en contournant les politiques de sécurité définies par l'entreprise. Les déploiements se font donc de manière anarchique, sans que quiconque maîtrise les risques de détournement d'appel ou d'écoute passive des communications.

Reste pour les administrateurs à utiliser des outils de filtrage spécifiques à ces protocoles, qui sont nombreux, mais pour la majorité payants. L'autre solution, beaucoup plus radicale, mais souvent utilisée, consiste à ne donner aucun droit d'administrateur aux utilisateurs, de manière à contrôler toute nouvelle installation logicielle.

## Les autres softphones

Les chapitres suivants de l'ouvrage présentent en détail les softphones les plus réputés. Nous ne mentionnons ici que les principales caractéristiques de quelques autres logiciels afin d'offrir un panorama assez large des offres actuelles.

### *WengoPhone*

Wengo est une société française spécialisée dans les services de ToIP, qui s'est recentrée, depuis 2007, sur les services d'aide entre particuliers par téléphone. Filiale de l'opérateur Neuf Cegetel, l'entreprise s'est fait une place sur le marché des solutions de téléphonie en proposant un logiciel complet et performant, WengoPhone.

La société ne dispose pas d'une force de frappe semblable à celle de ces concurrents, Microsoft ou Yahoo!. Elle ressemble un peu à Skype à ses débuts, pour qui le bouche-à-oreille a remplacé le marketing direct. Pour s'imposer, WengoPhone se présente comme un outil fédérateur face aux logiciels propriétaires.

L'un des atouts innovants de WengoPhone est l'ouverture de son code source. Chacun est libre de consulter l'implémentation du logiciel et d'y apporter sa contribution dans une version particulière du logiciel, appelée WengoPhoneNG.

WengoPhone utilise le protocole SIP pour la signalisation d'appel, respectant ainsi les standards protocolaires, ce dont peu d'autres logiciels peuvent se targuer.

Le logiciel peut être installé sur la plupart des systèmes d'exploitation existants : MacOS X, Linux (multiplate-forme), Windows 2000, XP ou Vista et Windows Mobile 5 et 6 pour Pocket PC et SmartPhone.

Téléchargeables sur le site de l'éditeur ([www.wengophone.fr/index.php](http://www.wengophone.fr/index.php)), les différentes versions du logiciel permettent d'adresser un maximum d'utilisateurs potentiels. Nous nous intéressons dans ce qui suit à la version Windows.

Une fois le programme d'une quinzaine de mégaoctets téléchargé et lancé, il est nécessaire de s'authentifier avec un compte Wengo. La création d'un compte est extrêmement rapide. Elle ne nécessite que trois informations de base : un identifiant, un mot de passe et une adresse e-mail (pour activer le compte).

Pour effectuer un premier appel de test, permettant de calibrer son matériel audio, il suffit de composer le 333 et de se laisser guider.

Peu de fonctionnalités de personnalisation sont disponibles, mais signalons tout de même la possibilité de dialoguer avec les logiciels de messagerie concurrents.

### Dialoguer entre logiciels concurrents

WengoPhone est le premier logiciel à proposer une solution de téléphonie avec les services associés, tout en assurant la compatibilité de sa messagerie instantanée avec les logiciels concurrents.

WengoPhone gère les communications textuelles avec des utilisateurs appartenant aux réseaux de messagerie les plus utilisés : WLM/MSN, Yahoo!Messenger, AIM/ICQ, Jabber et GoogleTalk.

Figure 8.3

*Ajout de comptes de messagerie externes*



Pour ajouter un contact appartenant à l'un de ces réseaux, il suffit de sélectionner les menus Outils et Compte de messagerie (ou Outils, Configuration et Comptes). La fenêtre illustrée à la figure 8.3 s'affiche alors. Les boutons situés au bas de la fenêtre permettent d'ajouter des contacts en mentionnant leur identifiant et le réseau auquel ils appartiennent.

Le plus étonnant est que cette compatibilité ne s'appuie sur aucun partenariat. Protégeant jalousement leur pré carré, les autres éditeurs de logiciels sont dépourvus de cette possibilité de converser indépendamment du réseau auquel appartiennent les contacts et ne diffusent généralement pas leur code source.

Seule la réglementation sur le droit à l'interopérabilité a permis cette fonctionnalité. Pour cela, Wengo a dû analyser en profondeur les communications réseau de chacun des logiciels tiers, en reproduisant le schéma protocolaire des communications. Malheureusement, ni l'audio ni la vidéo n'offrent encore cette souplesse. De plus, l'analyse fastidieuse des données peut être remise en cause chaque fois qu'un éditeur modifie ses couches protocolaires. En outre, le logiciel Skype a échappé à ces analyses.

Le logiciel WengoPhone est également fourni en marque blanche du fournisseur d'accès et opérateur téléphonique Neuf Cegetel pour son offre de téléphonie mobile, c'est-à-dire sans que le nom du logiciel ne soit mentionné dans la commercialisation des produits. C'est ainsi que le cadre du projet Beautifull Phone, destiné à promouvoir la convergence des communications fixes et mobiles, le téléphone Twin, embarque une version du logiciel WengoPhone.

Le combiné hybride proposé par de Neuf Cegetel, à la fois GSM et Wi-Fi, tente prioritairement de se raccorder au réseau domestique IP de l'opérateur, en utilisant la technologie Wi-Fi et la signalisation SIP. Ainsi, le combiné va chercher par Wi-Fi n'importe quel réseau IP sur lequel il est autorisé à se connecter, par exemple celui d'un abonné de l'opérateur ou plus généralement d'une zone dans laquelle l'opérateur a conclu des accords. Si cette connexion est possible, la communication téléphonique s'effectue en IP et est facturée aux conditions tarifaires de la ligne fixe de l'abonné, avec notamment, la gratuité vers les postes fixes partout en France, donc d'une manière très avantageuse par rapport aux conditions tarifaires d'une ligne mobile. À défaut de connexion au réseau IP, le mobile Twin se connecte au réseau GSM, aux conditions tarifaires du contrat mobile.

L'offre commerciale Unik de France Télécom propose un service équivalent.

Avec quelques milliers d'inscriptions par jour, WengoPhone a beaucoup de chemin à parcourir pour rattraper les millions de téléchargements de Skype. Mais, étant fondé sur des protocoles standardisés, ouverts et compatibles avec les systèmes de messagerie concurrents, WengoPhone possède des atouts indéniables, dont ne disposent pas toujours les logiciels concurrents.

## Téléphoner gratuitement d'un PC vers un téléphone fixe

Parmi les dizaines de sites qui proposent de téléphoner gratuitement d'un PC vers des téléphones fixes sur des dizaines de destinations, citons notamment les suivants :

- Netappel ([www.netappel.fr](http://www.netappel.fr))
- VoIPBuster ([www.voipbuster.com](http://www.voipbuster.com))
- VoIPStunt ([www.voipstunt.com](http://www.voipstunt.com))
- VoIPCheap ([www.voipcheap.com](http://www.voipcheap.com))
- VoIPDiscount ([www.voipdiscount.com](http://www.voipdiscount.com))
- InternetCalls ([www.internetcalls.com](http://www.internetcalls.com))
- SIPDiscount ([www.sipdiscount.com](http://www.sipdiscount.com))

En dépit de la profusion de ces sites, ce modèle ne semble guère viable à première vue. En réalité, derrière tous ces sites se cache une seule et même société, Finarea SA, basée à Lugano en Suisse. La multiplicité des offres a pour unique objectif de rabattre de nouveaux clients vers cette société.

### Multiplication des offres volatiles

Créée en avril 2000, Finarea SA ([www.finarea.ch](http://www.finarea.ch)) est spécialisée dans la téléphonie discount et s'est fait connaître par des tarifs ultra-agressifs, parfois difficilement rentables, mais progressivement revus à la hausse.

Tous les logiciels proposés par Finarea reposent sur une offre d'appel généralement similaire : les communications sont offertes vers un grand nombre de destinations, quelle que soit la durée de la communication. La seule contrainte est que l'utilisateur qui ne détient qu'un crédit nul ne dispose que d'une durée de communication de quelques minutes, le plus souvent deux. Au-delà de cette durée, l'appel se termine automatiquement.

Si l'utilisateur crédite son compte, ne serait-ce que de quelques euros, le crédit n'est pas décrétementé pour les communications gratuites, et les communications n'ont plus aucune restriction de durée. En clair, le modèle incite à acheter des crédits, lesquels ne seront pas forcément utilisés, mais donneront droit à des appels illimités.

Les conditions d'utilisation sont moins claires quant à la liste des pays que l'on peut joindre gratuitement. Le logiciel proposé étant en version Bêta, cette liste varie en permanence et n'est pas garantie. Autrement dit, un utilisateur peut s'acquitter d'un crédit pour téléphoner à un certain tarif vers une destination et découvrir du jour au lendemain que le tarif a été modifié sans préavis.

La vigilance est donc de mise, y compris pour la liste des destinations vers lesquelles les appels sont gratuits. Certains pays apparaissent, tandis que d'autres deviennent subitement payants. Petit à petit, et une fois le service connu, l'opérateur change sa grille tarifaire pour faire place à des tarifs qui n'ont plus rien de gratuit. En attendant, un nouveau

service similaire ouvre sur un autre site, avec un nom différent, mais une même ergonomie du site et du logiciel, et une nouvelle offre tarifaire agressive est proposée.

Signalons pour finir que les crédits ont une durée de vie limitée à quelques mois. Par conséquent, l'appelant doit les recharger régulièrement pour pouvoir appeler « gratuitement ».

## **Les clients de messagerie Web**

Meebo ([www.meebo.com](http://www.meebo.com)) illustre le concept de messagerie multiprotocole sur le Web. Avec un simple navigateur, il est possible de se connecter simultanément sur ses comptes WLM, Yahoo!, Jabber/GoogleTalk et AIM/ICQ.

En se connectant à la page d'accueil de Meebo, l'utilisateur est invité à saisir les identifiants et mots de passe de tous les comptes de messagerie sur lesquels il souhaite se connecter. Une fois authentifié, il retrouve sur l'interface de son navigateur l'ensemble des contacts de ses différents comptes, et peut communiquer avec eux. En outre, il dispose de la possibilité de réorganiser la page en redimensionnant ou déplaçant les fenêtres de dialogue à sa guise. C'est la technologie Ajax (Asynchronous JavaScript and XML) qui permet cette liberté d'action. Ajax s'inscrit dans la mouvance de ce que l'on appelle le Web 2.0, exprimant une nouvelle forme d'ergonomie grâce à laquelle les pages Web, en plus d'être dynamiques, deviennent personnalisables, à la manière de véritables applications.

Ce service est limité à la messagerie, et il n'est pas possible d'effectuer des appels téléphoniques ou des vidéoconférences ni d'échanger des fichiers. Meebo est du reste encore en version Bêta, donc en cours de développement.

## **La téléphonie Free**

Comme bon nombre de ses concurrents, l'opérateur Free, filiale du groupe Iliad, a développé une offre de téléphonie fixe qui se base sur le protocole de signalisation SIP. Mais, contrairement à beaucoup d'autres, il ne s'est pas arrêté là et a étendu son offre commerciale avec deux possibilités très intéressantes :

- Le compte SIP : Free fournit à tous ses abonnés un login et un mot de passe associés à un compte SIP. Ce compte permet à l'abonné de se connecter en dehors de son domicile, à partir d'une simple connexion Internet, pour accéder aux mêmes services de téléphonie et services que ceux auxquels il a accès de chez lui. Autrement dit, l'abonné n'a pas besoin d'être chez lui pour profiter de la gratuité des appels en France et dans des dizaines de destinations dans le monde, ainsi que des conditions avantageuses de tarification de son forfait. Il lui suffit de disposer, d'une part, d'une connexion à Internet et, d'autre part, d'un logiciel SIP correctement configuré.
- La Freephonie : la Freephonie propose une extension du domaine d'accès de l'utilisateur, de son domicile à un vaste réseau constitué de toutes les Freebox compatibles. Autrement dit, l'utilisateur qui est abonné chez Free et qui se déplace en dehors de sa

résidence peut accéder à toutes les Freebox qui l'entourent, appartenant à d'autres abonnés. Bien sûr, la connexion n'est accessible qu'à partir d'un compte privé. Donc si les appels ne sont pas couverts par l'abonnement, ils seront naturellement facturés à l'appelant, et non à celui qui laisse sa Freebox à disposition.

La Freephonie étend les possibilités du compte SIP : alors qu'avec un compte SIP, il est nécessaire d'avoir un accès Internet ouvert, le réseau Freephonie fournit à l'abonné un ensemble de points d'accès à Internet, qui peuvent être sécurisés et fermés pour les utilisateurs en général, mais seront accessibles et ouverts pour cet usage en particulier.

Dans ce qui suit, nous allons décrire brièvement la manière d'accéder à ces deux services.

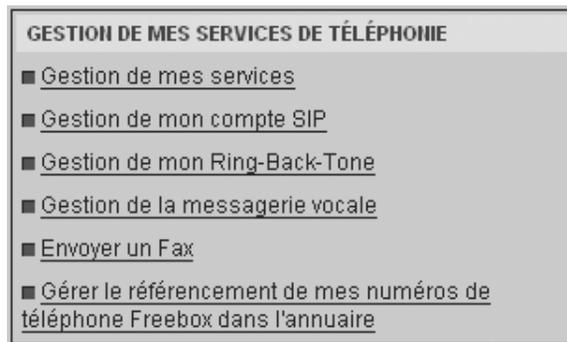
### Le compte SIP

Pour accéder aux services téléphoniques de Free à partir de n'importe quelle station (PC, MAC, PDA ou SmartPhone, comme l'iPhone d'Apple), il est nécessaire d'avoir un accès à Internet et un logiciel SIP configuré avec ses paramètres personnels.

Pour connaître ses paramètres personnels, l'internaute doit se rendre dans son interface Web de gestion de compte, accessible à la page <http://subscribe.free.fr/login/>. Une fois son identification faite, l'internaute accède à sa console de gestion, lui permettant de gérer toutes les fonctionnalités de son compte. La section « Gestion de mon compte SIP » (voir *figure 8.4*) permet d'afficher les paramètres SIP de son compte.

Figure 8.4

Accès au compte SIP



En cliquant sur ce lien, on repère trois paramètres importants (voir *figure 8.5*) :

- L'identifiant du compte : c'est le numéro de téléphone de l'abonné, constitué de dix chiffres (identiques au paramètre nommé *Utilisateur*).
- Le mot de passe : constitué de dix caractères au moins, il est laissé au libre choix de l'utilisateur.
- Le domaine : son nom (*freephonie.net*) est celui du serveur de Free que l'abonné va solliciter pour l'établissement de sa connexion.

On remarque également deux options : la première permet, lors de la réception d'appel, de faire sonner tout combiné ou logiciel configuré avec les paramètres SIP précédents, ou bien (cas par défaut) de faire sonner le téléphone relié à la Freebox ; la seconde permet d'activer ou de désactiver ce service. Il est indispensable que cette case soit cochée pour que l'abonné puisse utiliser son compte SIP. Nous verrons à la section suivante l'utilité des certificats mentionnés en bas de la page de configuration.

Une fois validé (par le biais du bouton Envoyer), le service est automatiquement et immédiatement disponible.

**Figure 8.5**  
*Paramètres du  
compte SIP*

**Merci de remplir les champs suivants**

Utilisateur : 0954545454

Identifiant : 0954545454

Mot de passe (10 caractères min.) :

Domaine : **freephonie.net**

Rediriger les appels entrants vers le SIP

Rediriger les appels entrants vers la FreeBox

Activer le service

**ENVOYER**

[Afficher les certificats pour  
votre Pocket PC /  
SmartPhone](#)

Pour y accéder, n'importe quel logiciel ou téléphone compatible avec le protocole SIP convient, pourvu qu'il soit disponible sur sa plate-forme (PDA, SmartPhone, portable ou ordinateur fixe). On pourra par exemple, choisir d'utiliser WengoPhone, SJphone, GoSIP (tous trois disponibles sous Windows, Mac, Linux et Windows Mobile), Xlite et Gizmo Project (tous deux disponibles sous Windows, Mac, Linux), Ekiga (Windows, Linux, BSD) et twinklephone (Linux), parmi les très nombreux logiciels SIP disponibles gratuitement, généralement sous licence GPL.

Pour accéder à son compte SIP et pouvoir téléphoner librement, comme si on se trouvait chez soi, il suffit de configurer les paramètres susmentionnés, à savoir l'identifiant et le mot de passe (privés) ainsi que le nom de domaine et l'adresse du serveur proxy, pour lesquels on entrera dans les deux cas l'adresse *freephonie.net*. Ces paramètres sont

généralement accessibles dans les options du logiciel. On évitera de modifier les paramètres dont la fonction n'est pas clairement identifiée.

Pour un exemple, reportez-vous à la section de configuration du logiciel XLite détaillée au chapitre 12, dédié au PBX Asterisk.

Free a récemment développé un logiciel permettant à tous les possesseurs d'un téléphone iPhone d'Apple débloqué de se connecter pour accéder à leur compte SIP de Free et de profiter des mêmes avantages.

Ce logiciel est disponible à la page <http://sip.free.fr>. La démarche est analogue à celle décrite ici, si ce n'est qu'elle implique le téléchargement du logiciel développé par Free et spécifique à la plate-forme de l'iPhone.

### Le réseau Freephonie

Le réseau Freephonie autorise l'abonné à se connecter sur n'importe quelle Freebox compatible (à partir des versions v5 ou HD) pour y établir ses communications téléphoniques. Seules les communications téléphoniques sont possibles, à l'exclusion de tout autre service, comme la navigation sur Internet. Ces connexions pourraient en effet consommer une bande passante importante de l'abonné chez lequel on se connecte, au détriment de la qualité de la connexion de ce dernier.

Pour cela, l'abonné doit télécharger et installer sur son terminal les certificats qui lui sont fournis tout en bas de la page de gestion du compte SIP que nous avons présentée précédemment, avec le lien « Afficher les certificats pour votre Pocket PC/Smartphone » (*voir figure 8.5*).

En cliquant sur ce lien, trois certificats sont disponibles (*voir figure 8.6*) : une clef privée, une clef publique et un certificat racine (dit root).

Si le téléchargement de ces fichiers ne pose pas le moindre problème, leur installation requiert toutefois quelques manipulations. En fait, à l'origine, l'opérateur Free commercialisait des téléphones SIP préconfigurés avec les paramètres nécessaires. Depuis peu, les paramètres ont été décryptés et diffusés sur Internet. La société Free semble avoir encouragé ce mouvement, en fournissant à chaque abonné les paramètres privés de son compte (les certificats vus plus haut), mais sans toutefois documenter le moyen de les mettre en œuvre.

Voici la méthode à suivre pour la génération et l'installation de ces certificats. Cette démarche s'oriente, pour l'exemple, vers un appareil (PDA ou SmartPhone) disposant du système d'exploitation Windows Mobile 6, mais la méthode est équivalente sous Symbian, si ce n'est que les menus diffèrent.

La première étape consiste à sauvegarder ces clefs et certificats dans des fichiers textuels de son ordinateur. Avec un simple éditeur de texte, comme Notepad, on recopie le contenu de chacune de ces deux clefs et du certificat dans un fichier (l'intégralité, débutant et se terminant par les premiers et derniers tirets, sans y insérer d'espace en début, ni fin de fichier). On nommera ces fichiers respectivement `free_clef_priv.txt`, `free_clef_pub.txt` et `free_certif_racine.txt`.

## AFFICHER LES CERTIFICATS POUR VOTRE ACCES SIP

Vous trouverez ci-dessous les certificats destinés à votre compte SIP permettant de configurer votre pocket PC / GSM compatible SIP :

## Clef Privée :

```
-----BEGIN RSA PRIVATE KEY-----
bZ1XI29BnrZ0zk0ZE4KY+pI380oPPbRTVW/LkPSjt3B4E50amEUX+hbEFQ1evwvG
wQADgYEAxvrv5kJKMU3tu8/aRAGjGDrIeD3yHgG96hDLczvGGf6mjgDYorpTvpLs1
ZTReFw0wODR0MTMsdMQ1MTRaFw0wODR2MTMsdMQ1MTRaME4xCzAJBgNVBAYTAkZS
MQ8wDQYDVQQIEwZGcmFuY2UsdJAMBGNVBAcTBVBhcm1zMQ0wCwYDVQQKEwRGMVl
BAGTBkZyYW5jZTEOMGwGA1UEBjMVFUGFyaXMsDTALBgNVBHoT7XAqIqDg6E+CMXgc
vTELMakGA1UEE2uvcoallNdos1pC4w8F2uvco+btEktf7A=
-----END RSA PRIVATE KEY-----
```

## Clef Publique :

```
-----BEGIN CERTIFICATE-----
MIICAjCCRWsCCQCPCb95/8o0PDANBgkqhkiG9w0BAQQFAAD9MQswCQYDVQQGEwJG
UjEPMA0GA1UECBMGRnJhbnNlMQ4wDgYDVQQHEwVQYXJpczENMAsGA1UEChMERnJl
ZTReFw0wODR0MTMsdMQ1MTRaFw0wODR2MTMsdMQ1MTRaME4xCzAJBgNVBAYTAkZS
Pv8m5dp6UV5V5jji7eDTjU0yDsnVaRt6IrvjdrxvhrAgMBAREwDQYJKoZIhvcNAQEE
BQADgYEAxvrv5kJKMU3tu8/aRAGjGDrIeD3yHgG96hDLczvGGf6mjgDYorpTvpLs1
x0ap9drdrgdDICUleHQArztCaA6Ln/TxjVaytomPd8vvT1u51k/IzEq4Xk75snpPu
jUTy7T/C6pPdVfGqAX6g/ymkRfUDuHNYgp1pC4w8F2uvcoeaJM=
-----END CERTIFICATE-----
```

## Certificat root freephonie :

```
-----BEGIN CERTIFICATE-----
MIICmDCCAgGgAwIBAgTBkZyYW5jZTELMakGA1UEBjMVFUGFyaXMsDTALBgNVBHoT
V BAYTAkZSMQ8wDQYDVQQIEwZGcmFuY2UsdJAMBGNVBAcTBVBhcm1zMQ0wCwYDVQQK
EwRGMVlBAGTBkZyYW5jZTEOMGwGA1UEBjMVFUGFyaXMsDTALBgNVBHoTALBgNVBHoT
njE7uY1BCD+RnAK/utK+ztmj1ek0WRupoQ9x29MLSWV1RboWY8qrYwP+GWDJ2yLb
hn2Lq7psS35ZJQpTELMakGA1UEPwj2/PhZQpSIS+5w/sqXy8A8aY+xdHuP+g=
-----END CERTIFICATE-----
```

Figure 8.6

Paramètres du compte SIP

La seconde étape consiste à installer OPENSSSL et à l'utiliser pour la génération de certificats valides. OPENSSSL peut être téléchargé, par exemple, à l'adresse <http://www.sproweb.com/products/Win32OpenSSL.html>. Une fois installé, on se rend dans le répertoire que l'on a choisi durant l'installation et qui contient l'exécutable openssl.exe. On y déplace les trois fichiers textes précédemment créés. Puis, on lance le programme openssl.exe et, dans le terminal qui s'ouvre, on saisit les deux lignes suivantes :

```
enc -d -base64 -in free_certif_racine.txt -out free_certif_racine.cer
```

```
pkcs12 -export -in free_clef_pub.txt -inkey free_clef_priv.txt -out free_certif.pfx
```

La première encode simplement le fichier sur la base 64. Elle n'est parfois pas indispensable, et une simple modification de l'extension .txt en .cer sur le fichier free\_certif\_racine suffit souvent, selon le système d'exploitation.

La ligne suivante génère, à partir de la clef publique et de la clef privée, un unique certificat au format PKCS 12. Un mot de passe (optionnel) de son choix est requis lors de la création de ce second certificat. Il permet de sécuriser ce fichier en limitant son utilisation à son concepteur.

Dès lors, dans le même répertoire que celui où se trouve la commande `openssl.exe` et les fichiers `free_clef_pub.txt` et `free_clef_priv.txt`, nous trouvons le certificat généré, dont le nom a été fixé à `free_certif.pfx`.

La quatrième étape consiste à déplacer les deux fichiers générés `free_racine.cer` et `free_certif.pfx` sur son PDA ou SmartPhone, et de les exécuter tous les deux en cliquant dessus. Le mot de passe, choisi précédemment, est requis lors de l'installation du fichier `free_certif.pfx` sur son terminal. Les deux installations doivent être validées par un message de succès.

La cinquième étape consiste à régler les paramètres de la connexion Wi-Fi. Pour cela, après avoir activé le Wi-Fi sur son terminal et sélectionné un réseau dont le nom est Freephonie, on sélectionne les paramètres de la connexion Wi-Fi. On y choisit une authentification WPA et un cryptage TKIP et, sur la page de configuration d'authentification, on sélectionne « Carte à puce ou certificat » comme type EAP. Puis, on clique sur le bouton Propriétés et, dans les certificats, on sélectionne celui référencé comme émis par Free par un clic prolongé permettant d'accéder aux propriétés de ce certificat. Dans la fenêtre qui s'affiche, on note simplement le numéro affiché sous la ligne « Octroyé à ». Il est utile dans la dernière étape. On valide cette étape après s'être assuré que le certificat Free est bien sélectionné.

La dernière étape consiste à valider la connexion sur le réseau Freephonie. Quelques secondes plus tard, une fenêtre s'ouvre, sollicitant un nom d'utilisateur et un nom de domaine. On entre pour le premier le numéro noté précédemment, et on laisse vide le champ domaine. La connexion doit alors s'effectuer avec succès : l'utilisateur est connecté au réseau Freephonie.

Le logiciel `sofphone` de son choix peut ensuite être utilisé, à condition de remplacer l'adresse du serveur proxy à contacter par l'adresse 172.17.20.241 (à la place de `freephonie.net`). On configure donc son logiciel SIP avec pour paramètres son identifiant, son mot de passe, le nom de domaine (`freephonie.net`) et l'adresse du proxy (172.17.20.241).

Les certificats ainsi créés ont une durée de vie de deux mois. Passé ce délai, il faut répéter la procédure. Cette limitation permet de restreindre les possibilités d'abus et de fraude. Cette fonctionnalité offre à l'utilisateur beaucoup de souplesse dans son exploitation, mais elle demeure liée à un contrat d'abonnement privé avec le fournisseur d'accès selon lequel une diffusion de ces certificats au-delà des termes du contrat serait illégale.

L'opérateur Neuf Télécom utilise lui aussi le protocole SIP pour les connexions téléphoniques de ses abonnés et leur permet de manière semblable à Free de se connecter au réseau constitué de toutes les NeufBox des abonnés, ainsi que dans différents hotspots.

## Conclusion

Le téléphone traditionnel a probablement encore une longue vie devant lui, mais les évolutions technologiques et les nouveaux usages des internautes jouent en la faveur du réseau IP.

Pour simuler une utilisation conforme au traditionnel réseau RTC, certains services proposent d'utiliser un combiné habituel en utilisant de manière transparente le réseau IP. C'est le cas, par exemple, du service Jajah (<http://www.jajah.com>), qui propose de saisir sur une interface Web le numéro de téléphone de l'appelant et celui de l'appelé. Le service se charge de contacter tour à tour les téléphones des deux correspondants en exploitant le réseau IP, en dehors de la boucle locale d'accès des correspondants, afin de bénéficier d'un tarif attractif.

Opérateurs et équipementiers proposent plusieurs équipements téléphoniques qui repoussent la contrainte de devoir allumer son ordinateur pour téléphoner et simplifient la convivialité des appels dans un réseau IP devenu presque transparent. En particulier, il existe des téléphones IP directement raccordables sur une prise Ethernet, sans nécessiter l'utilisation d'un ordinateur. De même, les boîtiers, ou box, fournis par les fournisseurs d'accès Internet disposent d'un branchement pour un téléphone non-IP, dont les flux sont automatiquement convertis par le boîtier pour passer dans le réseau IP.

Le service n'est toutefois pas toujours garanti, puisque la maîtrise du réseau ne peut passer que par l'opérateur, et non pas par l'éditeur de logiciels. En outre, les appels d'urgence ne sont possibles que si l'utilisateur dispose d'un crédit non nul. La qualité audio s'avère en revanche globalement bonne, parfois meilleure qu'en téléphonie circuit classique. Il est probable que les tarifs extrêmement compétitifs des softphones vont modifier les comportements des utilisateurs, les incitant à passer plus de temps au téléphone.

Les chapitres suivants de l'ouvrage détailleront les logiciels de téléphonie grand public les plus répandus que proposent les poids lourds de l'industrie, comme Skype, Microsoft, Yahoo! ou Google.

Au terme de ce panorama nécessairement simplificateur, une question se pose, à laquelle nous n'avons pu ou voulu répondre : parmi tous ces clients, lequel faut-il privilégier ? Le problème est trop complexe pour pouvoir y apporter une réponse définitive. Tout dépend des besoins et préférences.

Il convient de noter quelques caractéristiques fondamentales de ces logiciels, qui permettront selon les usages de choisir au cas par cas. Dans notre cadre, le logiciel doit impérativement posséder les fonctionnalités de téléphonie sur IP. C'est le cas de toutes les solutions que nous allons présenter, qui proposent cette possibilité, avec une qualité audio globalement excellente. Les prix étant variables selon les destinations, et plus intéressantes au cas par cas avec certains softphones qu'avec d'autres, le critère tarifaire s'avère difficilement distinctif *a priori*.

Restent les services complémentaires disponibles. L'ergonomie des logiciels est souvent affaire de goût, les uns préférant la simplicité d'une interface de type Google Talk, les

autres s'attachant davantage à la large palette de services et de personnalisation d'une interface de type Yahoo! Messenger. La compatibilité et l'évolutivité des logiciels sont d'autres paramètres à prendre en compte. Enfin, les services proposés varient selon les logiciels. En proposant un numéro d'appel entrant associé à une messagerie, le logiciel Skype possède indéniablement une longueur d'avance sur ces concurrents au niveau de la téléphonie.

# 9

## Skype

---

Avec plus de 100 millions d'utilisateurs (dont 4 millions de Français) dans près de 225 pays, Skype, leader mondial et pionnier des offres de téléphonie grand public sur Internet, bouleverse l'industrie des télécommunications en modifiant en profondeur les habitudes des consommateurs.

Chaque jour, 150 000 nouveaux utilisateurs téléchargent cette solution de peer-to-peer (P2P) qui propose deux services différents : une offre gratuite entre utilisateurs équipés du logiciel pour une exploitation purement Internet et une offre payante, qui permet de joindre et d'être joint *via* Internet tandis que les correspondants utilisent la téléphonie traditionnelle RTC.

Skype est l'un des premiers logiciels grand public à avoir permis la jonction entre la téléphonie du monde Internet et celle du monde RTC. C'est sans doute là la clé de son succès. Grâce à une qualité d'écoute excellente, une facilité d'utilisation ne nécessitant généralement aucune configuration (y compris dans les infrastructures réseau déployant des pare-feu), une mobilité accrue, une gamme de services complémentaires et un prix incomparablement moins cher que la téléphonie traditionnelle, Skype s'est répandu de manière virale.

Skype a été lancé le 29 août 2003 à l'initiative de Niklas Zennström, un Suédois de 36 ans, et Janus Friis, un Danois de 26 ans, tous deux experts en technologies de peer-to-peer puisqu'ils avaient fait frémir l'industrie de l'entertainment au début des années 2000 avec le logiciel Kazaa, qu'ils avaient conçu.

Zennström et Friis ont ensuite lancé Altnet, l'un des premiers services légaux de vente sécurisée de contenus commerciaux, également fondé sur le peer-to-peer. Encouragés par ces expériences réussies, plusieurs partenaires ont soutenu le projet Skype en y investissant près de 20 millions d'euros.

La société Skype a vu le jour au Luxembourg et s'est tout de suite imposée sur le marché naissant de la téléphonie logicielle. Grâce à Skype, les internautes pouvaient communiquer sur Internet par la voix, sans considération de distance, ni de durée. Skype n'est pas le seul à proposer un service de ce type, mais la qualité de son service demeure inégalée, tant par le soin apporté à l'ergonomie de l'interface que par la simplicité de configuration du logiciel ou la qualité de la communication.

Microsoft et Google d'abord, puis Yahoo! et News Corporation (la société de Rupert Murdoch), s'intéressent à Skype, mais, en 2005, ses deux fondateurs créent la surprise en vendant la société à eBay, un acteur pour le moins inattendu dans le domaine des télécoms.

Le géant américain de la vente aux enchères sur Internet s'offre la jeune société de ToIP logicielle pour une somme record, comprise entre 2,1 et 3,3 milliards, qui laissera plus d'un analyste financier perplexe. Quelles motivations ont-elles pu pousser une entreprise prospère à se risquer dans un domaine qu'elle ne maîtrise pas, qui n'est pas son secteur d'activité et qui la place en concurrent direct de mastodontes tels que Microsoft, Google ou Yahoo! ?

L'objectif affiché par les dirigeants d'eBay était de créer une synergie de son activité grâce à la communication. Puisque le contact physique avec les objets mis en vente était impossible, la possibilité de discuter de vive voix des caractéristiques des produits, de leur état général ou des conditions de vente, voire de négocier pouvait devenir un véritable atout.

En facilitant l'acte d'achat, eBay s'ouvrait de nouveaux vecteurs de vente. De surcroît, fort de ses millions d'utilisateurs, Skype était susceptible d'apporter une clientèle importante à sa nouvelle maison mère. En retour, Skype bénéficiait du gigantesque réseau de clients payants d'eBay. Pour que la boucle soit bouclée, ce dernier a aussi racheté la société PayPal, dont le site de vente aux enchères utilisait le système de paiement sécurisé.

## Architecture de Skype

Comme expliqué précédemment, Skype fonctionne selon un mode décentralisé et une architecture peer-to-peer (P2P), c'est-à-dire de poste à poste, ou point-à-point, ou encore de pair en pair ou d'égal à égal, dans lequel chaque poste intermédiaire est susceptible de jouer le rôle de relais et de participer de manière dynamique au processus d'acheminement des paquets.

Le client logiciel n'est pas seulement utilisé par le possesseur du logiciel. Il est mis à contribution pour les besoins d'autres utilisateurs et sert de support de transmission aux flux de ces derniers. Chaque élément du réseau (on parle de nœuds) constitue à la fois un client, qui peut demander un service, et un serveur, qui peut agir pour le compte d'un autre client. Ce modèle distribue ainsi totalement ses traitements, à l'opposé du traditionnel modèle client-serveur, dans lequel chaque entité joue exclusivement le rôle de serveur ou de client, ce qui nécessite de centraliser les flux vers des centres de contrôle.

Le terme peer-to-peer est parfois utilisé pour désigner toute communication directe entre un poste et un autre, indépendamment du mode de routage des données utilisé. C'est là un abus de langage, puisque le routage caractérise la technologie P2P et définit un moyen de transporter des informations faisant intervenir des terminaux intermédiaires de proche en proche jusqu'au véritable destinataire. Skype est d'ailleurs le seul softphone parmi ceux que nous présentons dans les chapitres suivants à pouvoir utiliser un mode de routage de type peer-to-peer.

Prenons un exemple concret. Si l'utilisateur Albert souhaite communiquer avec Béatrice, pour quelle raison ses communications doivent-elles être acheminées vers le poste de Claude ? Le terminal de Claude n'a *a priori* rien d'un routeur. De plus, ses ressources privées vont être mises à contribution sans qu'il en ait conscience. Le chemin le plus court étant évidemment le lien direct Albert-Béatrice, pourquoi passer par Claude et établir le chemin Albert-Claude-Béatrice ? Il y a deux raisons principales à cela, la volonté de limiter les ressources utilisées et celle de traverser les pare-feu.

### **Limiter les ressources**

Le modèle décentralisé peer-to-peer proposé par Skype fait reposer l'intelligence du réseau sur les utilisateurs eux-mêmes, et non sur des serveurs centraux. Dès lors, le passage à l'échelle est permis à moindres frais, puisque chaque nouvel utilisateur est potentiellement une source de traitement pour l'ensemble du réseau.

Skype a ainsi pu s'étendre sur toute la planète sans avoir à s'intéresser directement aux ressources de traitement de la montée en charge. C'est l'un des secrets de sa réussite.

### **Traverser les pare-feu**

Une condition essentielle de la réussite de la ToIP est la possibilité de traverser les pare-feu. Les communications de ce type exploitent des ports dynamiques, qui ne sont généralement pas ouverts par ces pare-feu. Par ailleurs, le réseau sur lequel se trouve l'utilisateur peut mettre en œuvre un mécanisme de NAT (Network Address Translation), ou translation d'adresse réseau, qui donne à l'utilisateur une adresse IP non routable sur Internet. Pour ces deux raisons, la communication directe entre correspondants est impossible.

Skype a trouvé la parade en exploitant différentes techniques. L'une d'elles consiste en l'utilisation de ports standards, qui sont étrangers à la téléphonie sur IP, mais qui présentent l'avantage d'être le plus souvent ouverts par les pare-feu. C'est le cas du port 80, associé généralement au Web pour le protocole HTTP.

Skype permet en outre d'utiliser des ressources situées à l'extérieur de la zone protégée par le pare-feu. Cette ressource peut être un utilisateur parmi d'autres, choisi pour accomplir cette tâche selon un algorithme propriétaire. Les flux IP de Skype suivent ainsi un chemin détourné lorsque le chemin direct est impossible. Ce sont de tels chemins

qu'empruntent les communications entre utilisateurs de Skype, lesquels se prêtent à la fonctionnalité de routage sans en avoir conscience et pour les besoins d'autres clients.

Ce type de connexion s'effectue aux dépens des utilisateurs intermédiaires, mais à un débit très faible, de 0,5 Ko/s, qui ne perturbe que très peu ces derniers. Ceux-ci sont en outre sélectionnés en fonction de la bande passante dont ils disposent afin d'assumer la charge supplémentaire induite par ces communications. L'idée du transfert relayé est d'avoir une communication, fût-elle de qualité médiocre, plutôt que pas de communication du tout.

Si l'architecture de Skype est globalement décentralisée, il existe cependant des serveurs centralisés, qui assurent un ensemble de fonctionnalités annexes indispensables à la communication. Par exemple, pour savoir si un utilisateur est connecté ou non, le logiciel se connecte à l'un de ces serveurs, qui informe de la disponibilité de tous les contacts.

## Les offres Skype

Pour l'appel entre internautes, le service Skype est totalement gratuit, ainsi que la visio-phonie, la messagerie instantanée, la vidéoconférence et le transfert de fichiers. Cette formule ne permet de communiquer qu'entre internautes, ce qui représente une restriction importante.

Outre cette offre d'appel, plusieurs services payants sont proposés, tels que les possibilités de communiquer avec les utilisateurs du réseau téléphonique commuté ou de disposer d'un numéro d'appel entrant et d'une messagerie vocale associée.

Deux options payantes sont proposées, :

- SkypeOut permet d'appeler à partir du logiciel et donc d'Internet vers le réseau téléphonique traditionnel (fixe ou mobile). Cette option est le fruit de partenariats mis en place au niveau mondial avec des opérateurs de téléphonie tels que Colt, Level 3 Communications, iBasis et Teleglobe, assurant une très large couverture de la zone Europe, États-Unis et Asie.
- SkypeIn permet d'être appelé à partir de téléphones fixes ou mobiles. L'utilisateur de Skype dispose pour cela d'un numéro de téléphone, qui se présente sous la forme d'un numéro géographique standard. Lorsqu'un appelant souhaite joindre un abonné ayant souscrit au service SkypeIn, il compose simplement son numéro d'appel. Cette communication est facturée par l'opérateur de l'appelant au prix d'une communication dépendante du numéro géographique de l'appelé. Autrement dit, si l'appelé possède un numéro géographiquement lié à la région de l'appelant, ce dernier sera facturé au prix d'une communication locale, quel que soit le lieu où se trouve physiquement l'appelé. Ce service est couplé avec une messagerie vocale. Les messages vocaux enregistrés peuvent ensuite être envoyés par e-mail en fichier audio à l'abonné appelé.

## Partenariats technologiques et commerciaux

Skype s'ouvre progressivement à des éditeurs et intégrateurs tiers. L'une des premières étapes dans cette direction a été la publication d'une API diffusée sous le nom de Skype-Net permettant l'intégration du logiciel directement depuis un site Internet. Auparavant, seule une fenêtre applicative du logiciel permettait aux utilisateurs de téléphoner entre eux. Depuis la publication de cette API, tout concepteur et développeur de pages Web peut exploiter les fonctionnalités de Skype sur une page Web. Sans être une nouveauté technologique, cet outil constitue une forme d'ouverture du logiciel, auparavant protégé et fermé, qui lui permet de renforcer sa présence sur Internet.

Les partenariats se sont ensuite étendus à de grands constructeurs de matériels de téléphonie, tels Siemens, Panasonic, Creative, Linksys ou US Robotics, avec lequel Skype a conçu des combinés téléphoniques utilisant son logiciel pour permettre d'appeler aussi bien sur Internet que sur le réseau téléphonique classique.

Ces téléphones utilisent une connexion filaire ou sans fil, de type DECT ou encore Wi-Fi, et ressemblent en tout point à des téléphones traditionnels, le logiciel Skype et la connectivité IP en prime (voir figure 9.1). Pour garantir le bon fonctionnement des appareils, qu'il s'agisse de la gamme de combinés téléphoniques ou des accessoires tels que les microcasques, la firme s'est même dotée d'un label commercial, appelé Skype Ready, qui fait office de norme d'interopérabilité afin de rassurer les consommateurs.



**Figure 9.1**  
*Téléphones Skype*

## La sécurité

Si aucune attaque ou vulnérabilité critique concernant le logiciel n'a encore été recensée, de nombreux spécialistes déplorent la facilité avec laquelle le logiciel parvient à traverser et se jouer des pare-feu censés bloquer les flux non autorisés. De ce fait, ces experts recommandent d'interdire l'utilisation du logiciel dans un cadre professionnel.

De leur côté, les autorités s'inquiètent du manque de transparence du logiciel, qui se comporte comme une boîte noire. Il n'est donc pas possible de savoir s'il contient une porte dérobée, accessible à partir d'Internet, pas plus qu'il n'est possible de savoir si des données sensibles sur les utilisateurs ne sont pas envoyées à leur insu.

Quant au fondement même du logiciel, le peer-to-peer, puisque les communications peuvent être acheminées *via* des ordinateurs intermédiaires, elles peuvent faire l'objet d'écoutes clandestines par ces mêmes intermédiaires.

De même, les pièces jointes transmises par l'outil de transfert de fichiers de Skype ne sont pas soumises à des contrôles d'antivirus. Même si l'on peut supposer que les interlocuteurs sont dignes de confiance, il n'en va pas forcément de même des fichiers qu'ils transfèrent, qui peuvent avoir été corrompus.

Selon un raisonnement paranoïaque, on pourrait imaginer que toutes les communications soient relayées vers des serveurs centraux qui les enregistreraient, agissant comme un système automatisé de profiling des utilisateurs. Techniquement, ce serait parfaitement réalisable. Dans le doute, et dans la mesure où Skype refuse d'ouvrir les spécifications de son protocole, on comprend la méfiance de certains.

Chez Skype, on garantit que le logiciel est parfaitement sécurisé et ne présente aucun risque pour l'internaute. Le cryptage se fait de bout en bout, au moyen de l'algorithme de chiffrement symétrique AES (Advanced Encryption Standard), le standard utilisé par les organisations gouvernementales aux États-Unis. AES utilise un cryptage sur 256 bits. La négociation des clés symétriques AES s'effectue par l'algorithme RSA (Rivest Shamir Adleman), avec des clés de 1 536 à 2 048 bits.

Si les spécifications générales du protocole ne sont pas rendues publiques, explique-t-on chez Skype, c'est uniquement pour offrir au logiciel une protection supplémentaire et éviter de donner aux pirates l'occasion d'y chercher des failles.

En entreprise, même si des administrateurs souhaitent bloquer l'utilisation de Skype chez les utilisateurs de leur réseau, dans la pratique il est très difficile de mettre en place une politique de sécurité qui prenne en compte les spécificités du logiciel. Le cryptage rendant impossible le filtrage, il n'est même pas possible de protéger le logiciel d'attaques malicieuses ou de le rendre compatible avec un système de détection d'intrusion IDS (Intrusion Detection System) puisque les échanges ne sont pas analysables.

Plusieurs sociétés proposent des logiciels qui détectent et bloquent les flux de Skype. Appelés SkypeKiller, ces logiciels poussent la reconnaissance de l'analyse protocolaire jusqu'au niveau applicatif ou comportemental, et pas uniquement en se fondant sur les protocoles de transport ou les ports, afin d'empêcher les flux Skype de traverser le réseau.

## Utiliser Skype

Skype est aujourd'hui le logiciel de ToIP le plus utilisé au monde. Même si l'arrivée dans ce secteur d'acteurs aussi importants que Google, Yahoo! et Microsoft promet des évolutions majeures, Skype demeure une référence pleinement fonctionnelle.

Nous allons détailler la richesse du logiciel, à travers ses fonctionnalités les plus courantes. Quoique simple d'utilisation, nous verrons qu'il offre une gamme de possibilités très étendue.

### Prérequis

Skype est disponible gratuitement en téléchargement. Il est disponible pour les quatre plates-formes suivantes :

- Windows 2000 , XP et Vista;
- MacOS X ;
- Linux, avec notamment des installeurs déjà disponibles pour les distributions Suse, Mandriva et Fedora, ainsi qu'une version binaire pouvant être installée sur d'autres types de distributions Linux ;
- Pocket PC, sous Windows Mobile 2003, Windows Mobile 5.0, Windows Mobile 6.0, permettant une utilisation nomade, pour une connexion en Bluetooth, Wi-Fi ou 3 G, par exemple.

Pour communiquer, un microcasque est nécessaire, ainsi qu'une webcam si l'on souhaite profiter des appels avec vidéo.

La machine sur laquelle on souhaite installer le logiciel doit avoir au minimum un processeur cadencé à 400 MHz, 15 Mo de disponible sur disque dur, ainsi qu'une mémoire vive de 128 Mo. Si l'on se limite à une connexion audio, une connexion Internet par modem à 56 Kbit/s est suffisante. Dans ce cas, il importe de ne pas avoir de téléchargements en parallèle, qui pourraient perturber le service de téléphonie.

Pour une connexion vidéo, il faut disposer d'une connexion haut débit. En moyenne, pour l'audio uniquement, la bande passante doit être comprise entre 3 et 16 Ko/s. Le codec utilisé pour une communication est automatiquement sélectionné d'après la bande passante disponible chez l'interlocuteur et les conditions intermédiaires du réseau. En l'absence d'appel, Skype utilise une bande passante inférieure à 500 o/s, essentiellement pour mettre à jour les informations de présence et de statut sur les contacts. La bande passante peut aussi être utilisée en petite quantité pour d'autres utilisateurs si un utilisateur sert de relais à d'autres communications.

Dans ce qui suit, nous nous intéressons uniquement à la version Windows de Skype, qui est la plus utilisée. Les autres versions ont une interface différente, mais on y retrouve des fonctionnalités semblables.

## Appeler

Après avoir téléchargé et installé le logiciel Skype, il est nécessaire de créer un compte Skype qui permettra de se connecter au réseau Skype.

Dès lors, les premiers appels avec le logiciel sont possibles. Pour tester le bon fonctionnement du logiciel, un contact spécial, nommé « echo123 », qui se trouve dans l'onglet des contacts du logiciel, est automatiquement ajouté lors de l'installation du logiciel. Il suffit de cliquer sur ce contact pour initier un appel de test. La fenêtre illustrée à la figure 9.2 s'affiche alors, avec notamment l'indication de la durée de l'appel. Un message vocal invite l'appelant à enregistrer un message audio de 10 secondes qui lui est immédiatement retransmis.

Si la langue choisie lors de l'installation est le français, le message audio est diffusé en français. La connectivité est assurée si l'appel est correctement réalisé. La réception du son est validée si l'appelant entend le message d'accueil de manière satisfaisante. De même, l'émission est validée si le message que l'on a transmis est correctement rendu.

Figure 9.2

*Appel de test*



Pour connaître la disponibilité d'un contact avant même de l'appeler, il faut l'ajouter à sa liste de contacts. Pour cela, on peut indifféremment utiliser le bouton Ajouter un contact ou sélectionner Ajouter un contact dans le menu Contacts.

Comme illustré à la figure 9.3, dès la première tentative de communication, l'appelant reçoit ce message personnalisé, lui indiquant qu'un utilisateur tente de l'ajouter à sa liste de contacts et l'invite à en faire autant. L'appelant peut soit accepter la demande, soit la refuser. Les communications vocales sont possibles entre les deux intervenants dans le premier cas, et bloquées dans le second.



**Figure 9.3**

*Accepter ou refuser un contact*

En acceptant ce contact, son identifiant est automatiquement ajouté à la liste des contacts. Dorénavant, sa présence et sa disponibilité sont immédiatement visibles, et il n'est plus nécessaire pour initier une communication avec lui de spécifier son pseudonyme, son nom ou son numéro de téléphone.

D'autres informations sont affichées par défaut dans la zone relative au contact, comme l'image qu'il s'est choisie, son pays ou son heure locale (*voir figure 9.4*).

**Figure 9.4**

*Nouveau contact de la liste*



En double-cliquant sur un contact, la communication audio est initiée. L'appelé est immédiatement informé par une sonnerie ainsi qu'une alerte visuelle (*voir figure 9.5*) qu'un correspondant cherche à le joindre. L'alerte lui précise les nom et prénom de l'appelant, en lui laissant choisir d'accepter ou non l'appel.

**Figure 9.5**

*Réception d'un appel*



L'absence prolongée de réponse est naturellement considérée comme une absence du correspondant. Si l'appelé choisit de rejeter l'appel, l'appelant est informé que la communication n'a pas été acceptée. Si l'appelé accepte l'appel, la communication débute instantanément.

## Outils

Parce qu'il se veut à la fois convivial et complet, Skype fournit une gamme d'outils facilitant son utilisation au quotidien. Ces outils sont progressivement devenus des classiques, que l'on retrouve sous différentes formes dans les autres logiciels grand public de téléphonie et de messagerie instantanée.

### Indicateur de présence

Skype dispose d'un indicateur de présence symbolisant la disponibilité de chacun des contacts de la liste.

Chaque utilisateur peut afficher son statut, c'est-à-dire son activité actuelle, selon les possibilités suivantes : Déconnecté, Connecté, Absent, Indisponible, Ne pas déranger, Skypez-moi, Invisible.

Ces indicateurs peuvent être définis de plusieurs manières (voir figure 9.6) :

- en sélectionnant Statut de connexion dans le menu Fichier ;
- en cliquant sur le logo à gauche de son nom, sous barre d'outils ;
- en cliquant sur le logo en bas à gauche de l'interface du logiciel (partie droite de la figure) ;
- en cliquant sur l'icône de la zone d'état système puis en sélectionnant « changer de statut ».

Figure 9.6

Configuration du statut



Le statut Skypez-moi permet à tout internaute, même s'il ne figure pas dans la liste de contacts, d'indiquer sa disponibilité. Le statut Invisible permet de voir la liste des personnes connectées sans être soi-même visible de ces contacts.

L'inactivité prolongée d'un poste, sans mouvement de la souris ni utilisation du clavier, peut amener le logiciel à se positionner automatiquement sur le statut Absent.

## Salons de discussions

Un nouveau service, nommé Skypecast, permet de créer ses propres salons de discussions ou de rejoindre des salons existants. Ce service gratuit permet de gérer jusqu'à 100 participants et est bien adapté aux conférences, même professionnelles.

L'initiateur du salon en est le modérateur. Il est donc habilité à autoriser ou bloquer les intervenants à parler. Deux modes de gestion de la conférence sont proposés :

- Mode libre, dans lequel chacun peut communiquer quand il le souhaite et intervenir sans demander l'accès. Le modérateur se contente de s'assurer que les échanges ne dérivent pas des règles générales d'utilisation du service *Skypecast*.
- Mode régulé, dans lequel les intervenants doivent se faire connaître auprès du modérateur pour avoir la possibilité de prendre la parole tour à tour. En plus de son rôle de contrôle des règles, le modérateur a celui de donner la parole aux intervenants.

Le modérateur peut décider dynamiquement de basculer dans un mode ou un autre.

## Transfert de fichiers

Il n'est pas surprenant de la part des concepteurs de Kazaa d'avoir inclus dans Skype la possibilité de transférer des fichiers en peer-to-peer.

Plusieurs manières permettent de réaliser un transfert de fichiers :

- En déplaçant les fichiers à envoyer sur l'icône du contact destinataire ou dans la fenêtre de ce contact, si elle est ouverte.
- En sélectionnant un contact dans la liste puis en cliquant sur le bouton d'envoi de fichier.
- En sélectionnant un contact par clic droit puis en sélectionnant l'option d'envoi dans le menu contextuel qui s'affiche.

Le transfert de fichiers est sécurisé par un cryptage analogue à celui utilisé pour les communications et autorise la reprise d'un téléchargement interrompu. Si une déconnexion intervient alors que le fichier n'est pas totalement transféré, le transfert peut reprendre où il a été interrompu, et non depuis le début.

Notons que les documents envoyés par ce moyen ne sont soumis à aucun test d'antivirus, ni par un serveur intermédiaire (il n'y a pas de serveur centralisé pour les transferts) ni par le logiciel lui-même (ce dernier ne dispose pas d'antivirus).

## Aller plus loin avec Skype

Certaines options évoluées de Skype permettent d'améliorer la convivialité et l'utilisation du logiciel.

Il est ainsi possible d'ouvrir plusieurs instances de Skype, de travailler en ligne de commande, de créer des commandes textuelles ou d'intégrer Skype dans ses pages Web et courriers électroniques.

## Ouvrir plusieurs instances de Skype

Si l'on souhaite partager son poste de travail avec des amis ou des membres de sa famille, il peut être utile de lancer plusieurs instances du logiciel. Un même utilisateur peut aussi avoir plusieurs comptes, par exemple en dissociant compte privé et professionnel. Skype empêche par défaut le lancement de deux instances du programme, mais il est possible de contourner cette contrainte.

Deux courtes étapes sont nécessaires pour cela. La première, effectuée une fois pour toutes, consiste à créer un second compte utilisateur sous Windows, et la seconde à lancer le programme sous le compte utilisateur que l'on vient de créer.

Nous allons détailler ces deux opérations. Nous supposons que nous disposons de deux comptes Skype, qu'une première instance de Skype est déjà lancée sur le poste de travail avec le premier compte Skype et que nous souhaitons ouvrir une nouvelle instance avec le second compte.

Pour la création d'un nouveau compte utilisateur sous Windows, il faut se rendre dans la section Comptes d'utilisateurs du Panneau de configuration et sélectionner Créer un nouveau compte. Il suffit alors d'entrer un nom associé à ce compte et de lui attribuer les droits d'administrateur. En cliquant sur l'icône créée pour ce compte, on peut attribuer un mot de passe au compte.

Pour exécuter Skype sous le compte qui vient d'être créé, il faut se rendre à l'emplacement où le programme a été installé. Ce chemin a été indiqué lors de l'installation du logiciel Skype. Par défaut, le répertoire utilisé se trouve dans **C:\Program Files\Skype\Phone**. Il suffit d'ouvrir ce répertoire puis, par un clic droit sur l'icône de Skype, de choisir l'option « Exécuter en tant que... » dans le menu contextuel (voir figure 9.7).

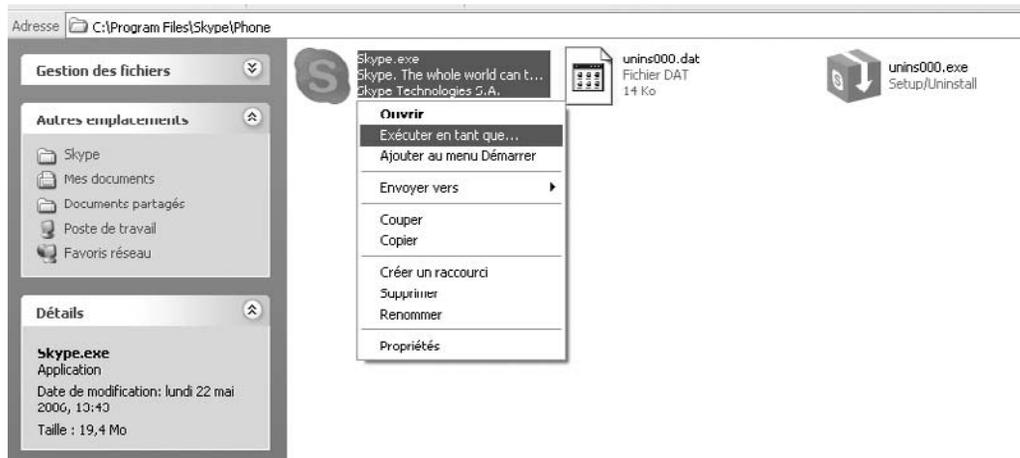


Figure 9.7

Menu contextuel Windows

Une nouvelle fenêtre propose de s'identifier sous un autre nom, comme illustré à la figure 9.8. Il suffit de sélectionner la case « L'utilisateur suivant », pour spécifier le nom et le mot de passe associés au nouveau compte Windows.

**Figure 9.8**  
*Exécuter Skype sous un autre compte Windows*



Après avoir renseigné les informations fournies à l'étape de création du compte Windows, Skype peut se lancer sous une nouvelle instance, et il devient possible de s'identifier sous un autre compte Skype à partir de cette instance.

On obtient ainsi deux exécutions différentes du même programme, comme l'illustre la figure 9.9.

**Figure 9.9**  
*Affichage simultané de deux instances de Skype*



Les deux icônes affichées dans la zone d'état système (voir figure 9.10) n'étant pas dissociables l'une de l'autre, il est recommandé de ne pas les utiliser par clic droit mais de se contenter de les activer par double-clic pour faire apparaître la fenêtre principale permettant d'activer le compte désiré.

Figure 9.10

*Icônes système des deux instances de Skype*



De cette manière, on peut lancer autant d'instances du programme Skype que l'on souhaite et ainsi recevoir les appels destinés à chacun des comptes associés.

Il est possible d'optimiser le lancement de la seconde étape en créant un raccourci vers Skype (par clic droit sur l'icône de Skype puis en sélectionnant Créer un raccourci. Une fois le raccourci créé, il suffit de le sélectionner par clic droit et de cliquer sur Propriétés. Dans l'onglet Raccourci, il faut ensuite cliquer sur le bouton Avancés puis sur la case « Exécuter en utilisant d'autres informations d'identification » avant de valider (voir figure 9.11). Désormais, chaque fois que l'on clique sur ce raccourci, on affiche la fenêtre de saisie de l'identifiant et du mot de passe du compte Windows avec lequel on souhaite exécuter le logiciel Skype.

Figure 9.11

*Propriétés avancées de Skype*



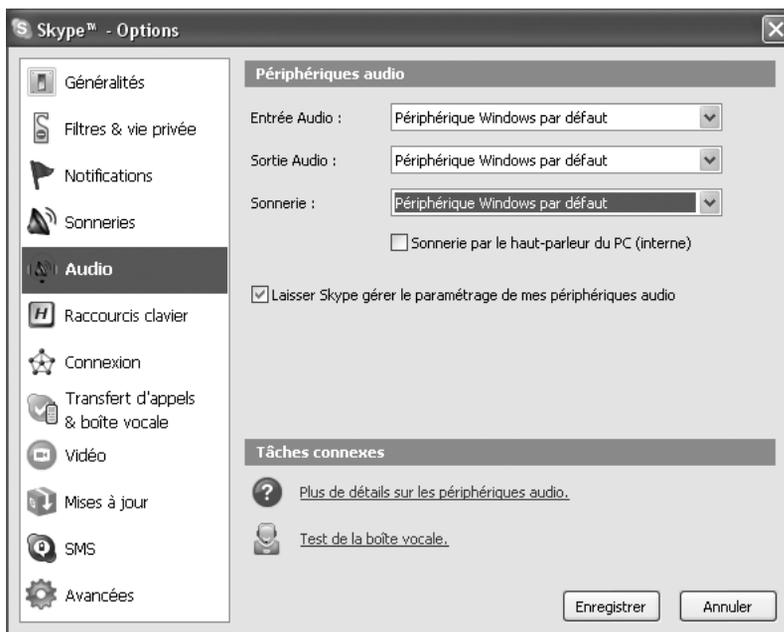
Une méthode encore plus rapide consiste à utiliser des programmes tels que FireDaemon Windows Service (gratuit, mais en anglais). FireDaemon Windows Service a pour vocation d'automatiser la saisie des paramètres d'authentification Windows lors du lancement de programmes.

Après avoir téléchargé, installé et exécuté l'application, une interface invite à saisir le nom de l'application à lancer et les login/mot de passe associés. De cette manière, il n'est même plus nécessaire de renseigner les informations d'authentification Windows, et le lancement de la seconde instance de Skype est immédiat.

Il existe cependant une limitation à l'utilisation de ces méthodes, qui vient de ce qu'une seule carte son ne permet pas de converser simultanément sur plusieurs comptes Skype lancés sur un même PC. La solution à ce problème consiste tout simplement à installer plusieurs cartes son et à associer à chaque compte un périphérique différent.

Comme illustré à la figure 9.12, la section Options du menu Outils de Skype propose un onglet Audio qui permet de réaliser cette association. Si l'on dispose d'un terminal téléphonique dédié à Skype, comme on en trouve dans le commerce, il est possible de le spécifier dans cette section et de s'en servir comme périphérique complémentaire à la carte son du PC.

**Figure 9.12**  
*Sélection du  
périphérique audio*



## Options en ligne de commande

Skype peut être exécuté par le biais d'options complémentaires exécutables en ligne de commande.

Les quatre options suivantes sont disponibles :

- */nosplash*. Par défaut, Skype affiche son logo avant le lancement du logiciel. Cette option interdit l'affichage du logo et lance immédiatement le logiciel.
- */minimized*. Par défaut, Skype ouvre sa fenêtre principale en premier plan. Cette option permet de minimiser le logiciel directement dans la zone d'état système lors de son lancement.
- */callto:numéro\_ou\_identifiant\_à\_appeler* (en remplacement de la partie *numéro\_ou\_identifiant\_à\_appeler* par un identifiant Skype ou un numéro de téléphone). Cette option permet d'appeler automatiquement le numéro ou l'identifiant Skype correspondant par simple clic sur l'icône de ce raccourci. Elle suppose bien sûr que les crédits soient suffisants pour cet appel, s'il est payant. Dans le cas contraire, un message d'erreur est retourné. Cette option peut se révéler utile pour des contacts très fréquemment appelés.
- */shutdown*. Cette option ferme directement le programme Skype.

Pour spécifier ces options, il faut créer un raccourci vers Skype (par clic droit sur l'icône de Skype puis en sélectionnant Créer un raccourci) puis sélectionner ses propriétés par clic droit à nouveau et en choisissant Propriétés dans le menu contextuel.

Dans l'onglet Raccourci, la case Cible permet de personnaliser le lancement avec les options désirées en ajoutant après les guillemets les options qui seront activées lorsqu'on cliquera sur cette icône (voir figure 9.13).

Figure 9.13

Options de raccourci



## Commandes textuelles

Au cours d'une conversion par messagerie instantanée, il est possible d'utiliser un certain nombre de commandes textuelles dans la zone de saisie de texte. Toutes ces commandes doivent être précédées d'un slash.

En saisissant la commande */help*, on obtient le récapitulatif des commandes disponibles, dont voici le détail :

- */add identifiant\_skype*. Permet d'inviter un contact à se joindre à la conversation en cours (en remplaçant *identifiant\_skype* par l'identifiant du contact à inviter). Par exemple, */add Nathalie* ajoute l'utilisateur Nathalie à la conversation en cours.
- */topic sujet*. Permet de spécifier ou modifier le thème de la conversation. Dans une conversation avec de nombreux participants, il est ainsi possible d'indiquer clairement en haut de la fenêtre le thème de la discussion. C'est une manière d'orienter ou d'animer le dialogue. Par exemple, */topic Bon Anniversaire Simon !* spécifie en haut de l'écran le sujet « Bon anniversaire Simon ! ».
- */me message*. Permet de mettre l'accent sur une action par un message particulier qui sera mis en relief. Cette commande est suivie d'un message textuel. Par exemple, si Rebecca saisit la commande */me est au téléphone avec Henri !*, chaque utilisateur avec qui Rebecca est en conversation verra s'afficher dans sa fenêtre le message « Rebecca est au téléphone avec Henri ! » encadré et centré.
- */history (sans argument)*. Affiche l'historique des messages.
- */find texte\_à\_chercher*. Recherche dans la communication en cours le texte indiqué. La première occurrence est sélectionnée et surlignée. Les suivantes sont accessibles en appuyant sur la touche F3.
- */fa* ou *l(sans argument)*. Répète la recherche précédemment effectuée. La commande a le même effet que la touche F3.
- */leave (sans argument)*. Permet de fermer la fenêtre de la messagerie instantanée.
- */alertson texte*. Active une alerte qui sera déclenchée en mettant en surbrillance le texte repéré dans une conversation. C'est notamment utile si l'on ne souhaite pas suivre la totalité d'une conversation avec de multiples intervenants mais que l'on désire être alerté par des éléments susceptibles de nous intéresser. Ces alertes peuvent être annulées en saisissant simplement la commande sans argument textuel.
- */alertsoff (sans argument)*. Désactive les alertes d'avertissement de messages, ce qui évite que la fenêtre de messagerie instantanée clignote dans la barre des tâches à chaque nouveau message.
- */call identifiant\_skype*. Permet d'appeler au téléphone (en remplaçant *identifiant\_skype* par l'identifiant du contact à inviter). Par exemple, */call Marc* lance un appel téléphonique à l'utilisateur Marc.

## Intégrer Skype dans ses pages Web et ses e-mails

Si un utilisateur maintient un site Web, il peut insérer dans ses pages des liens invitant les internautes à le contacter par Skype pour obtenir des informations complémentaires ou simplement converser.

Il suffit pour cela d'insérer des balises spécifiques dans ses pages Web. Ces balises s'insèrent comme un code HTML dans un lien hypertexte.

La syntaxe générale d'une balise Skype respecte la forme suivante :

```
<a href= "skype:contact(s)?action">
Message textuel
</a>
```

Cette balise comporte trois parties à remplacer selon les usages :

- *contact(s)*. Spécifie le contact Skype concerné par le lien hypertexte. Le contact peut être indiqué par un identifiant Skype ou par un numéro de téléphone standard. Il peut être unique ou concerner plusieurs contacts séparés par des points-virgules.
- *action*. Spécifie l'action à enclencher lors d'un clic sur le message textuel. Les actions les plus classiques sont les suivantes :
  - *call* : appelle le ou les contacts mentionnés ; c'est l'action par défaut.
  - *add* : ajoute le ou les contacts mentionnés à sa liste de contacts.
  - *userinfo* : affiche le profil du ou des contacts mentionnés.
  - *chat* : lance une conversation par messagerie instantanée avec le ou les contacts mentionnés.
  - *sendfile* : envoie un fichier aux contacts mentionnés.
- *Message\_textuel*. C'est le message textuel affiché sur la page Web à l'internaute.

Par exemple, pour lancer une conférence audio avec les utilisateurs *guy.laurent* et *dupont.durand* en cliquant sur un lien hypertexte de sa page Web, il suffit de placer la balise suivante dans le code source de la page, à l'emplacement où l'on souhaite afficher un message textuel invitant l'utilisateur à cette conférence :

```
<a href="skype:guy.laurent;dupont.durand?call">
Téléphoner à Guy Laurent et Dupont Durand
</a>
```

En cliquant sur ce lien, le navigateur de l'internaute instancie automatiquement l'application Skype et initie l'appel vers ces correspondants. Bien évidemment, cela suppose que l'application Skype soit installée sur le poste de l'internaute. Dans le cas contraire, le navigateur de l'internaute ne parvient pas à interpréter ce code Skype et lui indique une erreur.

De la même manière, les coordonnées Skype peuvent être indiquées en signature d'un e-mail, pour permettre au correspondant de lancer un appel d'un simple clic sur l'e-mail.

### Utiliser une image de statut

Ces balises permettent aux internautes de contacter les auteurs des pages Web ou des courriers électroniques qui les comportent, mais pas d'être informés de la disponibilité de ces derniers. Pour faciliter la mise en relation, Skype propose des scripts qui testent la présence des utilisateurs et l'affichent sous la forme d'une image dynamique. Ce service est appelé SkypeWeb.

L'image permet non seulement de spécifier les coordonnées (identifiant ou numéro de téléphone) d'un contact Skype, mais aussi de détecter et d'afficher son statut de disponibilité. Ainsi, les internautes peuvent savoir avant d'appeler si leur correspondant est en ligne.

Un exemple d'image de statut Skype est illustré à la figure 9.14. Il est possible de personnaliser cette image.

**Figure 9.14**  
*Image de statut Skype*



Pour afficher son statut, il faut tout d'abord avoir autorisé la diffusion de cette information dans le logiciel Skype lui-même. Pour cela, il faut sélectionner Options dans le menu Outils, puis, dans la fenêtre des options qui s'ouvre, cocher la case « Permettre que mon statut soit visible depuis le web » de la section Filtres & Vie privée.

Il ne reste qu'à ajouter dans sa page Web ou comme signature de ses e-mails l'icône affichant ce statut.

Nous allons utiliser une icône disponible sur le site de Skype, mais il est possible d'en choisir une autre (à l'adresse <http://www.skype.com/go/skypebuttons>) ou d'en générer une soi-même.

Dans le corps du code de la page Web (après la balise `<body>` et à l'emplacement où l'on désire ajouter l'icône Skype), il suffit d'insérer la section de code suivante (ce code affiche une icône récupérée sur le site de Skype et lui donne l'apparence correspondant à son propre statut) :

```
<script type="text/javascript" src="http://download.skype.com/share/skypebuttons/
  ↳ js/skypeCheck.js">
</script>

<a href="skype:guy%2Elaurent?call">

</a>
```

Dans les trois premières lignes, on charge une page de script en langage JavaScript disponible sur le site de Skype afin de vérifier les statuts des utilisateurs. Comme l'indique la balise suivante `<a href>`, un clic sur l'image dont le code suit appelle l'utilisateur *guy.laurent* (le code `%2E` correspond au point). L'image est renseignée ensuite. Son emplacement se trouve directement sur le site de Skype, à l'adresse `http://mystatus.skype.com/bigclassic`. Il n'est donc pas nécessaire de la fournir dans sa page Web.

En remplaçant le mot *bigclassic* dans l'adresse précédente par *balloon*, *smallclassic*, *smallicon* ou *mediumicon*, on obtient d'autres images utilisables de la même manière. Toutes ces images utilisent un paramètre qui est l'identifiant de l'utilisateur Skype et qui doit être fourni à sa suite. En utilisant le script inséré juste au-dessus, ce paramètre permet d'afficher l'image selon le statut réel de l'utilisateur dont l'identifiant est fourni. Périodiquement, le statut de l'utilisateur est testé, et l'image correspondante actualisée au besoin.

Indépendamment des balises, mentionnons l'existence d'un module additionnel pour les navigateurs Internet Explorer et Mozilla Firefox. Ce module se présente sous la forme d'une barre d'outils appelée Skype Web Toolbar, qui s'intègre au navigateur une fois son installation effectuée. Dès lors, tous les numéros de téléphone et identifiants Skype diffusés sur le Web sont automatiquement reconnus et exploitables par le logiciel en un simple clic.

## Recommandations et résolution de problèmes

Il est vivement recommandé d'utiliser un microcasque de bonne qualité, faute de quoi la communication s'en ressent.

Dans la majorité des cas, les problèmes rencontrés se résolvent facilement en vérifiant un certain nombre de points élémentaires.

Voici quelques recommandations simples en cas de problèmes rencontrés dans l'utilisation de Skype :

- Vérifier sa connexion Internet en ouvrant une page Web avec son navigateur.
- Éviter d'effectuer des transferts en parallèle d'une communication, surtout si le débit est limité. Skype nécessite un débit minimal pour fonctionner, et il revient à l'utilisateur de veiller à en disposer. Dans la pratique, un débit insuffisant peut provoquer des hachures lors de la communication ou des interruptions désagréables. Elles peuvent provenir du poste de travail de l'appelant comme de celui de l'appelé.
- Un simple redémarrage résout bien des problèmes mystérieux.
- Tester une communication en se connectant sur son compte à partir d'un autre poste de travail. Cela permet de savoir si le problème provient du compte Skype ou du poste de travail utilisé. Si la communication demeure possible sur un autre poste de travail, le problème peut être localisé sur le premier poste. Si l'on ne dispose pas d'un autre poste de travail, on peut toujours utiliser le même poste, mais avec un compte Skype différent.

- S'assurer que d'autres utilisateurs arrivent à communiquer. Si le problème est situé au niveau des serveurs de Skype, il faut attendre le rétablissement du service.
- Vérifier (en le saisissant à nouveau) que le mot de passe du compte est correct et n'a pas été modifié.
- Tester une communication avec le contact test « echo123 ». Cela indique de manière fiable si le microcasque fonctionne correctement. Au besoin, mettre à jour le pilote de sa carte son.
- Utiliser les guides, FAQ et aides en ligne de Skype. Un grand nombre de ces documents sont disponibles en français.

Si ces conseils de base ne suffisent pas à la résolution de problèmes plus complexes, les forums spécialisés sur Skype sont généralement assez réactifs.

## Conclusion

En proposant une solution de ToIP logicielle simple, fiable et bon marché, Skype s'est fait un nom et a perturbé un équilibre que l'on croyait établi. Skype grandit chaque jour au point de devenir incontournable dans le grand public et de s'installer progressivement dans les entreprises.

Les forfaits de téléphonie illimitée proposés dernièrement par de nombreux opérateurs remettent cependant en cause le modèle même de Skype. Reste qu'avec Skype, l'utilisateur peut être mobile et disposer d'un indicateur de présence, de la vidéo en plus du texte, des échanges de fichiers et des conférences à plusieurs.

À sa défaveur, Skype ne repose sur aucun protocole normalisé et n'est pas ouvert aux protocoles concurrents. Si Skype ne s'ouvre pas à la compatibilité, il risque de s'enliser au moment même où ses concurrents se rassemblent, à l'image du rapprochement des systèmes de messagerie de Microsoft et de Yahoo!.

De plus, en utilisant un protocole propriétaire, incompatible aussi bien avec H.323 que SIP, Skype empêche ses utilisateurs de migrer facilement vers des solutions concurrentes. C'est là un frein majeur à son développement.



# 10

## Windows Live Messenger et Yahoo! Messenger

---

Acteurs majeurs du monde de l'Internet, Microsoft et Yahoo! proposent une vaste gamme de services similaires : moteur de recherche, messagerie Web, mais aussi messagerie instantanée.

Utilisée par plusieurs millions de personnes dans le monde, la messagerie instantanée n'a pas échappé à la déferlante de la ToIP et permet désormais à ses utilisateurs de téléphoner vers le réseau RTC traditionnel, en plus des conversations audio et vidéo sur le réseau IP.

Ce chapitre présente les principales caractéristiques de ces deux logiciels et montre comment les mettre en œuvre, à la fois pour la téléphonie mais aussi pour les autres fonctionnalités remarquables dans cette convergence des services.

### Windows Live Messenger

Microsoft aborde de manière très générale la gestion des communications. S'appuyant sur sa puissante capacité à développer des logiciels couvrant les besoins les plus communs, il a opté pour une stratégie reposant sur l'unification la plus large possible. Sa suite logicielle Microsoft Office System peut même être intégrée directement dans les applications propriétaires grand public de la firme.

Cette offre n'a pas vocation à servir uniquement les besoins des particuliers, mais s'étend aux besoins professionnels, notamment pour le travail collaboratif en temps réel, en mettant l'accent sur la garantie de transmission des données sans interruption intempestive.

Son outil phare pour les communications temps réel en entreprise est LCS (Live Communications Server), un sous-ensemble de Microsoft Office System. Parallèlement, Microsoft poursuit et amplifie son offensive dans les services unifiés, avec sa gamme Live.

### **La gamme de services unifiés Live**

En 1996, la stratégie commerciale de Microsoft n'était pas vraiment tournée vers le réseau Internet. Il croyait encore en un réseau purement Windows, The Microsoft Network, ayant pour vocation de connecter tous les ordinateurs sous plate-forme Windows. Ce n'est donc pas le protocole IP qui fut choisi pour fédérer les communications réseau, mais spécifiquement et exclusivement le système d'exploitation propriétaire Windows, très majoritairement implémenté sur les postes de travail. Lorsqu'il installait Windows 95 sur son poste, l'utilisateur voyait s'afficher systématiquement l'icône The Microsoft Network, censée permettre de relier tous les utilisateurs Windows. Finalement, ce réseau Microsoft n'a jamais sérieusement concurrencé Internet.

De façon analogue, cette même année 1996, ce n'était pas le navigateur de Microsoft, Internet Explorer, qui dominait le marché, mais Netscape. Ce dernier, pourtant payant, occupait une part de marché d'environ 75 %. Microsoft a alors changé radicalement de stratégie en mettant toute sa puissance technique et commerciale sur Internet. Intégrée à Windows 98, la version 4 du navigateur Internet Explorer a pris de vitesse Netscape et s'est imposée comme le nouveau navigateur grand public. Parallèlement, le service The Microsoft Network disparaissait.

Le réseau purement Microsoft a laissé place à un service réseau à l'échelle d'Internet, dont la dénomination MSN (MicroSoft Network) rappelle son prédécesseur. Ce service réseau regroupe différents services, dont les quatre plus importants sont les suivants :

- MSN : site Internet, portail et moteur de recherche, qui demeure l'un des plus visités au monde.
- Hotmail : messagerie Internet (webmail), qui s'est également vite imposée comme une référence.
- Msn Messenger : client de messagerie instantanée, devenu emblématique pour les millions d'internautes qui l'utilisent.
- Msn Spaces : espace de stockage, ou blog, pour tenir un journal personnel à caractère humoristique ou informatif, appuyé par des sons, photos ou vidéos.

L'ensemble de ces services est accessible aux utilisateurs par le biais d'une authentification unique, appelée Passport .Net.

En 2006, la firme de Redmond, désireuse de rapprocher tous ces services dans une gamme unique, a décidé de les renommer et d'axer uniformément leur développement vers le Web 2.0. Le nom de code de la nouvelle stratégie Microsoft est *Live*, pour symboliser l'échange, l'interactivité, la rapidité et le temps réel. Le bouquet de services a donc pris le nom de WLS (Windows Live Services), et s'est enrichi d'une cinquantaine de

nouveaux services estampillés Live (par exemple Windows Live Drive, un outil de stockage de données en ligne, ou Windows Live Academic Search, un moteur de recherche pour universitaires). Quant à l'ancien système d'identification Passport .Net, il a été renommé Windows Live ID.

Les services fédérés par WLS ont également été tous renommés : MSN est devenu Live, Hotmail est devenu Windows Live Mail, Msn Spaces est devenu Windows Live Space, et MSN Messenger est devenu WLM (Windows Live Messenger). C'est à ce dernier outil que nous nous intéressons dans la suite de cette section.

### **LCS (Live Communications Server)**

Dans le domaine de l'entreprise, l'outil de communication temps réel LCS (Live Communications Server) est un serveur logiciel qui assure la connectivité aux utilisateurs et offre un ensemble de services. Il est couplé avec un logiciel client, Microsoft Office Communicator.

L'originalité de LCS est de couvrir tous les moyens de communication classiques intégrés au sein d'une même interface. La gestion de présence des utilisateurs est introduite afin de fournir en temps réel un indicateur de disponibilité des interlocuteurs, accolé à un mécanisme d'alertes sur les changements de statut.

LCS utilise le standard SIP pour unifier l'ensemble de ces services et permettre l'interopérabilité avec des solutions externes. Windows Web Live Meeting, dédié aux conférences en temps réel, offre la possibilité d'organiser des réunions interactives, avec jusqu'à deux mille intervenants, en partageant voix, vidéo, texte, fichiers et même tableau blanc. Pour être le plus large possible, la plate-forme logicielle est extensible et prévoit la compatibilité avec les services de messagerie les plus courants.

### **WLM (Windows Live Messenger)**

Avec près de 300 millions d'utilisateurs, WLM est l'un des clients de messagerie les plus utilisés au monde. La grande nouveauté de WLM réside dans la possibilité d'appeler des utilisateurs sur le réseau téléphonique traditionnel (fixe ou mobile). Pourvu de cette fonctionnalité, l'affrontement avec Skype, qui, de son côté, revendique le plus grand nombre d'utilisateurs à sa solution de ToIP logicielle, semble inévitable.

Les sections qui suivent présentent quelques spécificités notables de WLM.

#### **Un protocole propriétaire et fermé**

Le système protocolaire utilisé dans WLM est propriétaire. Appelé MSNP (Mobile Status Notification Protocol), il fonctionne sur une couche de transport TCP avec le port 1863.

Publié sous forme de draft en 1999 dans sa version 2, MSNP a beaucoup évolué et en est à sa version 14. Ce système est totalement fermé, et seuls des analyses des flux entrants et sortants ou des partenariats technologiques permettent d'envisager des interactions.

L'architecture générale de WLM est centralisée. Les serveurs Microsoft assurent à eux seuls l'enregistrement de tous les abonnés, avec leurs statuts et leurs pseudonymes. En changeant d'ordinateur, ces derniers conservent à la fois leur liste de contacts et le dernier pseudonyme qu'ils ont choisi.

Toutes les communications textuelles sont également centralisées et transitent par les serveurs Microsoft. Les utilisateurs ne sont donc pas mis en relation directement, sauf s'ils décident d'initier un transfert de fichier ou une communication audio ou vidéo, auquel cas les flux de données sont trop volumineux pour transiter par les serveurs Microsoft et ces derniers se contentent de faire connaître aux correspondants leur adresse IP respective.

Longtemps disponible uniquement sous Windows, WLM est depuis peu implémenté sur MacOS X. Cette version ne dispose cependant pas de toutes les fonctionnalités de son équivalent Windows, et la vidéo fait notamment défaut.

## Utiliser WLM

Pour utiliser WLM, il faut télécharger le logiciel à l'adresse [www.windowslivemessenger.fr](http://www.windowslivemessenger.fr). À l'exécution, l'interface du logiciel invite l'utilisateur à s'authentifier en saisissant son identifiant unique et le mot de passe associé.

Si l'utilisateur ne dispose pas encore d'un compte, il lui suffit de cliquer sur Créer un nouveau compte. Ce compte est appelé Windows Live ID.

Avant même d'être connecté, il est possible de sélectionner le statut avec lequel on souhaite se connecter. C'est notamment utile pour le statut Invisible, qui permet de ne se connecter que pour savoir qui est en ligne et éventuellement de discuter avec certains contacts tout en restant invisible des autres.

Une fois l'authentification effectuée, on accède à la fenêtre principale (voir figure 10.1), plus sobre et épurée que celle de Skype.

Figure 10.1

Fenêtre principale de WLM



Pour personnaliser cette interface, il faut faire apparaître la traditionnelle barre de menus. Dans un souci d'ergonomie, cette barre est masquée par défaut et s'affiche en cliquant sur la flèche descendante, en haut à droite de l'interface. En sélectionnant les menus Outils et Options, on accède à la configuration des préférences.

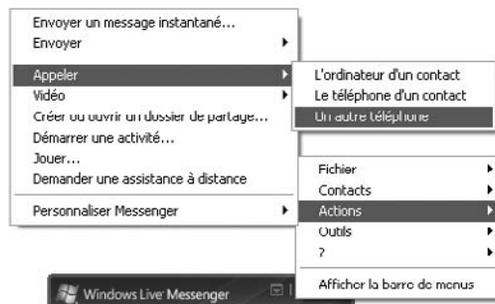
### Le service de téléphonie Windows Live Call

WLM inclut la possibilité de téléphoner de façon purement IP entre internautes, mais sa principale nouveauté réside dans la possibilité de passer des appels d'un PC vers un téléphone fixe ou mobile. Pour proposer ce service, Microsoft s'est associé à l'opérateur américain MCI, qui a été racheté par Verizon Communications, le deuxième opérateur téléphonique américain.

L'accès au service se fait en sélectionnant Appeler puis Un autre téléphone dans le menu Actions (voir figure 10.2).

**Figure 10.2**

*Fonctionnalités de WLM*



L'interface du service Windows Live Call invite alors l'utilisateur à acheter des crédits de communication lui permettant d'émettre des appels. Une fois l'achat effectué, la fenêtre illustrée à la figure 10.3 permet de composer le numéro du correspondant à joindre.

**Figure 10.3**

*Appel avec Windows Live Call*



L'utilisateur a la possibilité d'émettre des appels à destination de plus de 220 pays à des tarifs variables, plus avantageux que ceux proposés par les opérateurs de téléphonie, et avec une excellente qualité sonore.

À la différence de Skype, Windows Live Call ne fournit pas d'option pour disposer d'un numéro de téléphone standard aux clients. Autrement dit, seuls les appels sortants sont envisageables avec la téléphonie standard, et il n'est pas possible de recevoir des appels provenant de téléphones standards. Par ailleurs, en téléphonie purement IP, il n'existe pas de messagerie vocale associée au compte.

### **Périphériques compatibles**

Pour étendre davantage son réseau de diffusion, Microsoft a mis en place des partenariats intégrant le logiciel WLM dans des téléphones traditionnels. Sans avoir besoin d'allumer son ordinateur, il devient possible grâce à ces combinés d'appeler et de recevoir des appels avec des contacts de sa liste WLM comme avec n'importe quel autre contact du réseau RTC. Philips est devenu le premier partenaire européen à distribuer un tel téléphone.

Il existe une version de WLM pour le système d'exploitation Windows Mobile de Microsoft, destiné aux téléphones mobiles intelligents, ou SmartPhones, et aux assistants personnels (PDA). Grâce à la convergence entre mobiles et PC, les opérateurs de téléphonie proposent également WLM sur des téléphones portables, notamment France Télécom, avec son offre spécifique Orange Messenger by Windows Live, Bouygues Telecom, avec le service iMode, ou l'opérateur virtuel Ten.

Dans un souci d'extension, Microsoft commence également à ouvrir ses API en proposant des kits de développement pour les programmeurs indépendants souhaitant faire évoluer le logiciel WLM.

## ***Aller plus loin avec WLM***

Pour se distinguer de ses concurrents, WLM va plus loin que la messagerie instantanée et la téléphonie sur IP, en proposant un ensemble de fonctionnalités spécifiques. Cette section présente quelques-unes d'entre elles.

### **Intégrer WLM sur ses pages Web**

WLM permet l'intégration de ses principales commandes dans des pages Web grâce à des balises spécifiques, qui évitent d'avoir à saisir manuellement l'identifiant d'un correspondant. L'internaute ne peut toutefois exécuter ces commandes que s'il utilise le navigateur Internet Explorer. De plus, en cliquant sur un lien Web, l'internaute actionne une commande qui ne peut être prise en charge que par le logiciel WLM. Il doit donc nécessairement avoir installé et lancé le logiciel.

Le tableau 10.1 récapitule les principales commandes disponibles (la partie **identifiant\_email** doit être remplacée par l'identifiant réel de l'utilisateur WLM concerné).

**Tableau 10.1 – Balises HTML pour lancer une commande WLM sur une page Web**

Action	Code
Ajouter un contact dont l'identifiant est <code>identifiant_email</code>	<code>msnim:add?contact= <i>identifiant_email</i></code>
Démarrer une conversation textuelle avec un contact dont l'identifiant est <code>identifiant_email</code>	<code>msnim:chat?contact= <i>identifiant_email</i></code>
Envoyer une invitation pour démarrer une conversation audio avec un contact dont l'identifiant est <code>identifiant_email</code>	<code>msnim:voice?contact= <i>identifiant_email</i></code>
Envoyer une invitation pour démarrer une conversation vidéo avec un contact dont l'identifiant est <code>identifiant_email</code>	<code>msnim:video?contact= <i>identifiant_email</i></code>

Par exemple, pour générer un lien permettant d'ajouter un contact dont le nom est *mon\_identifiant@hotmail.fr*, il faut insérer dans le code Web de sa page la balise suivante :

```
<a href="msnim:add?contact= mon_identifiant@hotmail.fr">
  Cliquer ici pour m'ajouter dans votre liste de contacts WLM !
</a>
```

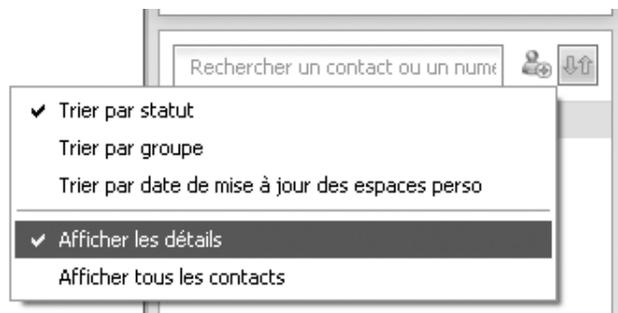
En cliquant sur ce lien, l'internaute déclenche un message du logiciel WLM lui demandant la confirmation d'ajout du contact.

### Afficher des images de ses contacts

Par défaut, l'interface n'affiche que de petites icônes à côté de chaque contact. Pour remplacer ces icônes par l'image choisie par ses correspondants, il suffit de cliquer sur le bouton Gérer vos contacts, à droite du champ de recherche des contacts (*voir figure 10.4*), puis de sélectionner Afficher les détails. L'image du contact s'affiche en taille réduite.

**Figure 10.4**

*Affichage de l'image d'un contact*



## Windows Live Contacts

WLM propose un outil de gestion avancée des contacts, appelé Windows Live Contacts, dont le fonctionnement repose sur l'enrichissement volontaire de chacun des utilisateurs.

Par clic droit sur la zone textuelle où figure son pseudonyme (juxtaposé à son image personnelle), un menu contextuel propose l'option Partager mes coordonnées. Dans la page Web qui s'affiche, l'utilisateur peut renseigner ses propres informations. Ces dernières étant confidentielles, il est possible de restreindre leur diffusion à des groupes de personnes ou à des personnes en particulier parmi la liste des contacts WLM affichée à la section Autorisation de la page Web.

Inversement, pour connaître les informations relatives à un contact, il suffit de sélectionner ce contact par clic droit puis de choisir Modifier un contact. Toutes les informations sur ce contact s'affichent alors.

## Ajouter un nom et mettre à jour les informations de ses contacts

Les contacts peuvent se présenter de façon ambiguë. Par exemple, deux personnes ayant le même prénom pour pseudonyme ne sont distinguables qu'en examinant leur adresse de messagerie. Pour permettre d'identifier plus simplement un contact, WLM propose une fonction d'alias.

Figure 10.5

Mise à jour d'un contact



Pour mettre en œuvre un alias, il suffit de sélectionner un utilisateur par clic droit et de choisir la fonction Ajouter un surnom dans le menu déroulant (voir figure 10.5). De cette manière, quels que soient les changements de pseudonyme effectués, les contacts sont facilement identifiables.

Lorsqu'un contact change son image, la modification n'est pas immédiatement reflétée chez les correspondants et n'est mise à jour qu'au moment où le contact se connecte. Il est possible de forcer cette mise à jour afin qu'elle s'effectue à intervalle régulier. Il suffit de sélectionner un contact par clic droit puis de sélectionner l'option « Recevoir les coordonnées mises à jour du contact », comme illustré à la figure 10.5.

### Exporter et importer sa liste de contacts

Pour diverses raisons, un utilisateur peut vouloir créer un nouveau compte. Pour cela, il doit obtenir un nouvel identifiant de connexion puis insérer un à un tous ses anciens contacts. Pour éviter cette saisie fastidieuse, WLM offre la possibilité d'exporter et d'importer sa liste de contacts.

Il suffit pour cela de sélectionner « Enregistrer les contacts Messenger » dans le menu Contacts. Un document portant l'extension **ctf** (pour contacts) est alors généré. Ce document doit être stocké afin de permettre l'importation future de ses contacts. L'exportation se fait de manière tout aussi simple dans le menu Contacts à l'aide de l'option « Importer des contacts Messenger ». La fenêtre qui s'ouvre invite l'utilisateur à spécifier le chemin où se trouve le fichier de contacts portant l'extension **ctf**.

Cette fonctionnalité permet en outre de partager sa liste de contacts avec d'autres personnes, par exemple un groupe d'amis communs ou un répertoire d'entreprise. En ce cas, l'importation ne peut être sélective, et il faut accepter tous les contacts de la liste ou aucun.

Il est néanmoins possible d'affiner ce processus et de ne sélectionner que certains contacts en recourant à une petite astuce. Le document portant l'extension **ctf** est en réalité un simple fichier texte éditable et modifiable dans n'importe quel éditeur de texte. Il suffit de le sélectionner par clic droit, de choisir Ouvrir avec dans le menu contextuel et de préciser l'éditeur de texte de son choix, WordPad ou Word. Il devient ainsi possible de supprimer des contacts ou d'en ajouter manuellement.

L'important dans l'édition de ce fichier est de respecter scrupuleusement sa syntaxe.

### Utilisateurs bloqués et hors connexion

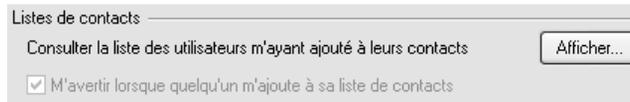
Un utilisateur ne peut jamais savoir si l'un de ses contacts l'a bloqué. Il est simplement vu comme étant hors connexion, rendant impossible toute tentative de communication. De la même manière, le statut d'un utilisateur invisible est confidentiel, et ses contacts ont seulement l'impression que celui-ci est déconnecté.

En revanche, un utilisateur bloqué par un autre utilisateur est souvent retiré de la liste des contacts de ce dernier. Or cette action peut être détectée manuellement à tout

moment. Il suffit de sélectionner le menu Outils, puis de choisir Options et d'ouvrir l'onglet Confidentialité, qui comporte la section Listes de contacts illustrée à la figure 10.6. Le bouton Afficher permet de voir la liste des contacts chez qui l'on figure en tant que contact. Par inversion, on peut en déduire les contacts chez qui l'on ne figure pas.

Figure 10.6

Option de la liste des contacts



### Ajouter un robot intelligent dans sa liste de contacts

La société Conversagent a conçu pour WLM un programme intelligent capable de répondre à des questions en exploitant la base de données encyclopédique Encarta de Microsoft. Ce robot, appelé Encarta Réponses Instantanées, a la particularité de répondre instantanément à des questions posées en langage naturel.

Ce type d'outil est décrit par Conversagent comme un ASA (Automatic Service Agent). Son originalité réside dans le fait qu'il est accessible sous la forme d'un contact WLM et que ses réponses sont fournies de manière concise dans une fenêtre de messagerie standard.

Pour disposer du robot en langue française, il suffit d'ajouter à sa liste de contacts l'identifiant *fr.encarta@botmetro.net*. En double-cliquant sur ce contact, la fenêtre de messagerie apparaît, et il est possible de poser toutes sortes de questions, même saugrenues. Quelques exemples de questions-réponses sont données à la figure 10.7.

Le service est prédisposé à répondre à plusieurs catégories de questions, mais il bloque bien souvent sur des questions *a priori* simples, ce qui le relègue davantage dans la catégorie gadget, du moins jusqu'à sa maturité, tout en méritant malgré tout un détour.

Le robot et ses réponses abrégées sont accessibles gratuitement. Pour compléter les réponses fournies par le robot, ce dernier invite l'utilisateur à utiliser optionnellement la version en ligne d'Encarta. Dans ce cas, un volet complémentaire à droite de la fenêtre de discussion s'ouvre pour afficher les réponses pertinentes du site Encarta. Cet accès est toujours gratuit, mais certains liens proposent l'accès à Encarta Premium, qui est la base la plus complète de l'encyclopédie, et nécessite un abonnement payant.

Il existe bien d'autres robots assignés à différentes fonctions, mais la presque totalité d'entre eux converse en langue anglaise. *smarterchild@hotmail.com* est probablement l'un des plus aboutis du moment. Évolutif, il apprend au fur et à mesure des conversations, en fonction de ce qui lui est dit. Pour inciter au développement de robots de ce type, Microsoft a lancé un concours récompensant les robots les plus performants (Robots Contest).

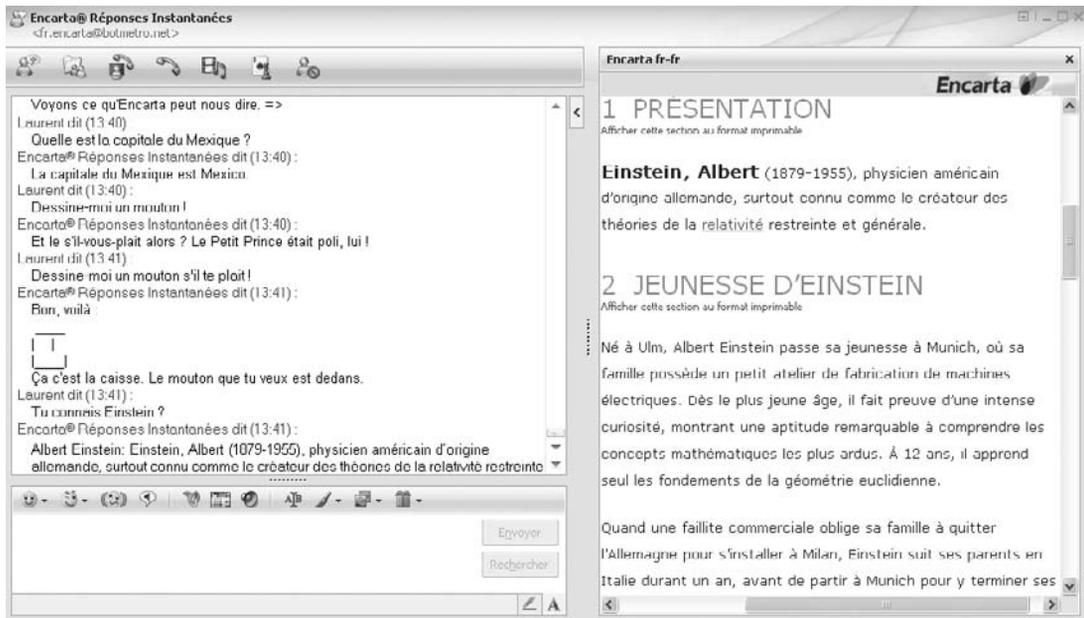


Figure 10.7

Exemples de questions-réponses au robot d'Encarta

### Censure automatique

Un peu par hasard, les utilisateurs de WLM ont découvert que certains mots étaient pros-crits lors des communications avec le logiciel. Il est par exemple impossible d'envoyer un message contenant les mots **download.php**, **gallery.php**, **profile.php** ou **.src** et **.pif**.

Même si le message contient d'autres mots, il est refusé par le serveur, et le récepteur pas plus que l'émetteur ne sont informés qu'un message n'a pu être transmis.

Dans le même genre, si, lors d'une conférence avec plusieurs intervenants, une personne utilise un de ces mots, elle se voit exclue de la conférence par un message pour le moins surprenant : « Vous avez momentanément des problèmes de réseau. Vous n'êtes plus dans la conversation. »

La raison invoquée par Microsoft pour justifier ce comportement étrange est la protection de l'utilisateur. Ce type de message est en effet fréquemment utilisé par les hackers pour inciter l'utilisateur à cliquer sur un lien, lequel lance insidieusement sur le poste client un virus ou un cheval de Troie. Pour protéger les postes utilisateur et éviter que le logiciel ne soit vecteur de telles attaques, Microsoft bannit l'utilisation des termes jugés sensibles. Reste que la contrainte subsiste pour les usages licites.

Il est en tout cas dommage que l'utilisateur ne puisse désélectionner cette option de protection envahissante. D'autant plus que Microsoft n'a pas communiqué la liste des mots interdits. En réalité, tous les flux de communication textuelle étant relayés vers les

serveurs de Microsoft, cette liste de mots peut être modifiée à tout moment. Par ailleurs, la centralisation des communications peut faire redouter aux plus sceptiques l'enregistrement de toutes les communications privées à des fins de profilage des utilisateurs, théoriquement possible, quoique totalement démenti par Microsoft.

### Web Messenger sur interface Web

Les fonctionnalités de base de WLM sont également disponibles sur une interface Web, appelée Web Messenger, sans avoir à télécharger le logiciel. Il suffit de se rendre à l'URL <http://webmessenger.msn.com>. Ce service est disponible sur Internet Explorer et Mozilla Firefox.

Pour en bénéficier, il est nécessaire d'autoriser les fenêtres pop-up pour lancer l'interface. Cette dernière ne propose que des fonctions réduites, essentiellement la messagerie instantanée entre utilisateurs, et il n'est pas possible de converser vocalement ni d'envoyer des fichiers.

Le service s'avère surtout utile dans les cas où l'installation du logiciel WLM est impossible. C'est le cas, par exemple, lorsqu'un utilisateur se trouve sur un ordinateur qui n'est pas le sien ou qu'il ne possède pas les droits d'administration sur ce poste ou encore qu'il n'utilise pas un système d'exploitation compatible avec le logiciel. Par ailleurs, dans un réseau filtrant et bloquant les connexions WLM, le Web Messenger a de fortes chances de passer, car il ne repose que sur des connexions Web classiques.

### Extensions et améliorations de WLM par des programmes tiers

Plusieurs tentatives d'amélioration du logiciel WLM ont été proposées par des particuliers, qui sont le plus souvent utilisateurs et développeurs et diffusent leurs créations gratuitement, en échange parfois de messages publicitaires.

Ces améliorations se présentent sous la forme de petits programmes à télécharger, qui étendent les fonctions de WLM et lui ajoutent des options. Il existe quantité d'extensions de ce type, chacune apportant son lot d'options complémentaires.

L'une de ces extensions parmi les plus connues est Windows Plus! Live. Gratuite, simple et disponible en français à l'adresse [www.msgpluslive.net](http://www.msgpluslive.net), elle existe depuis plusieurs années, et bien des améliorations qu'elle a apportées ont été progressivement intégrées dans la mouture officielle du logiciel de Microsoft. Parmi les fonctionnalités innovantes de cette extension, signalons la possibilité de lancer plusieurs instances de WLM, de regrouper toutes les conversations dans une fenêtre unique à onglets, de modifier l'apparence des fenêtres, de surveiller des événements, tels que le changement de statut d'un contact, d'effectuer des substitutions de messages textuels ou encore d'interroger un compte de messagerie externe POP3.

Lors de l'installation de ce logiciel, il est proposé à l'utilisateur de choisir l'option « Installer les sponsors ». Il est recommandé de désactiver cette option, car ces « sponsors » sont en réalité bien souvent des spywares, petits logiciels espions qui épient

l'activité des utilisateurs sur leur poste de travail et forcent l'affichage de bandeaux publicitaires.

Après l'installation du logiciel, une configuration de base est proposée automatiquement en redémarrant WLM. La configuration complète peut s'effectuer à tout moment *via* l'option Préférences du menu Plus!. Notons que les développeurs peuvent utiliser cette extension et l'enrichir par des scripts complémentaires (*via* le menu Général puis la section Scripts).

Parmi les autres extensions de ce type, signalons MessengerDiscovery Live, disponible à l'adresse <http://live.msgdiscovery.com> et similaire à Messenger Plus! Live ou, plus originale, Fake Webcam, qui remplace la diffusion d'une vidéo par webcam par celle d'une vidéo de son choix ([www.fakewebcam.com](http://www.fakewebcam.com)).

## Yahoo! Messenger

Avec près de 90 millions d'adeptes dans le monde, Yahoo! Messenger, la messagerie instantanée de Yahoo!, est très proche de la messagerie de Microsoft.

Tous deux ont fondé une grande partie de leurs développements sur le Web (en intégralité dans le cas de Yahoo!) et sont en concurrence sur plusieurs services. Leur moteur de recherche et leur messagerie Web respectifs ont été largement couronnés de succès. Dans les deux cas, le logiciel de messagerie instantanée fait office de portail vers leurs propres services mais aussi un ensemble de partenaires qui proposent des services souvent payants.

Plus étonnante, l'ergonomie des interfaces offertes dans les deux logiciels est frappante de ressemblance : barre de recherche de contacts, barre de recherche Web, barre d'outils, barre de publicité, liste de contacts et même photos sont situées aux mêmes emplacements.

### Utilisation

Chose rare, les utilisateurs d'ordinateurs Mac et UNIX n'ont pas été oubliés par le logiciel proposé par Yahoo!. Pour les possesseurs de PC sous Windows, le client de Yahoo! Messenger est disponible à l'adresse <http://fr.messenger.yahoo.com>. Pour les autres plateformes, il faut se tourner vers la version anglaise du site, aux adresses <http://fr.messenger.yahoo.com/mac.php>, pour MacOS 8/9/X, et <http://fr.messenger.yahoo.com/unix.php>, pour PC sous UNIX.

Nous ne détaillons dans les sections suivantes que la version Windows.

L'installation de Yahoo! Messenger est plus délicate que celle de WLM. Non seulement le logiciel est plus volumineux, mais il peut aussi se révéler particulièrement intrusif si l'on n'y prend garde.

En effet, après avoir téléchargé l'installeur du programme, il faut procéder avec prudence dans le choix des options. Deux possibilités sont offertes : une installation typique et une

installation personnalisée. Si l'on choisit l'installation typique, la page d'accueil de son navigateur Web devient celle du portail yahoo.fr, et le moteur de recherche par défaut (accessible par la touche F3 sur Internet Explorer) devient celui de Yahoo!. Il est donc vivement recommandé de préférer l'installation personnalisée, qui affiche clairement les éléments à installer et permet de désactiver les choix inopportuns.

Une fois l'installation effectuée, le logiciel s'ouvre sur la page invitant l'utilisateur à s'identifier avec son identifiant de compte Yahoo! et son mot de passe associé ; la connexion s'effectue alors, donnant accès au logiciel.

### Un logiciel portail

L'interface principale de Yahoo! Messenger est particulièrement riche en fonctionnalités. Hautement personnalisable, le logiciel se distingue de ses concurrents par l'utilisation d'un mécanisme de plug-in, permettant d'ajouter des fonctionnalités au produit de base.

La figure 10.8 illustre l'interface standard par défaut.

**Figure 10.8**

*L'interface principale de Yahoo! Messenger*



Les plug-in sont situés au bas de l'interface, juste au-dessus de la barre de recherche de Yahoo!. On y trouve l'accès à des brèves d'actualités, à de la musique en ligne, à des jeux en solo ou en réseau, à des blogs, ainsi qu'aux cours de la Bourse.

Ces plug-in sont accessibles par simple clic et s'ouvrent dans un nouveau panneau de contrôle, sur le côté du logiciel. Ils sont configurables (en cliquant sur le bouton Plugins). Le client logiciel de Yahoo! se distingue par cette personnalisation de l'interface, laquelle devient bien davantage qu'une plate-forme de communication et fait office de portail vers

l'information en continu, la météo, l'actualité spécialisée, en passant par la musique en ligne, les résultats sportifs, etc.

Le statut avec lequel l'utilisateur est connecté est affiché en haut, en regard de sa photo ou image. Pour le modifier, il suffit de cliquer sur la flèche descendante et de sélectionner celui qui convient ou d'en créer un nouveau.

Juste en dessous de la photo, une barre d'outils donne accès aux fonctions les plus courantes suivantes :

- Ajout de contacts.
- Affichage de la liste de contacts dans le panneau central du logiciel.
- Modification du panneau central pour le remplacer par le carnet d'adresses listant l'ensemble des contacts.
- Accès par lien direct à la messagerie Yahoo! associée au compte courant.
- Accès à son répondeur de messages audio.
- Accès à Yahoo! 360°, le service de blog de Yahoo!.

La barre de saisie alphanumérique située en dessous de la barre d'outils simplifie les mises en relation par une aide à la saisie. À défaut de parvenir à interpréter les caractères saisis, elle permet de lancer une recherche de texte dans le moteur de Yahoo!.

Dans la liste des contacts affichée dans la zone centrale, l'icône de chaque contact précède les pseudonymes correspondants. Le statut des contacts, s'il est autre que Disponible, est mentionné à la suite des pseudonymes.

### Fonctionnalités évoluées

En plus du mécanisme de plug-in, Yahoo! Messenger offre des fonctionnalités évoluées, parmi lesquelles les services gratuits suivants :

- Création d'avatars. Les avatars définissent l'image qui est affichée à côté de son pseudonyme et qui est diffusée aux contacts.
- Gestion des conférences. Plusieurs correspondants peuvent se joindre à un salon de discussion. L'invitation à rejoindre la conférence s'effectue par la fonction « Inviter des amis à une conférence » du menu Actions.
- Transfert de fichiers. Des documents de taille très importante (pouvant atteindre le gigaoctet) peuvent être échangés simplement en déplaçant les documents dans la fenêtre de messagerie d'un contact. La fonction de transfert est aussi disponible par le biais de la fonction « Envoyer un fichier » du menu Actions.
- Visiophonie. En plus d'échanger des messages textuels et audio, les conversations peuvent ajouter la vidéo si l'utilisateur dispose d'une webcam. La gestion des appels vidéo se fait soit directement dans la fenêtre de dialogue par le bouton Webcam, soit par le biais de la fonction « Inviter des contacts à voir ma webcam » du menu Actions.

La fonction « Afficher la webcam » du même menu sollicite l'affichage de la webcam d'un contact sélectionné.

### Téléphoner avec Yahoo! Messenger

Parmi les facettes remarquables du client Yahoo! Messenger, le service de téléphonie est exploitable pour appeler aussi bien des utilisateurs équipés du logiciel (téléphonie IP pure) que des téléphones standards.

Les périphériques audio sont généralement reconnus automatiquement, et aucune configuration particulière n'est nécessaire. En cas de conflit ou de problème audio, les réglages des paramètres audio peuvent s'effectuer par le biais du menu Messenger en sélectionnant Préférences. Dans la liste des catégories disponibles qui apparaît à gauche de la fenêtre, la section « Appel et Audio » s'affiche pour donner accès au choix des périphériques audio et à un assistant de configuration.

Avec le logiciel, la téléphonie IP pure est accessible simplement et gratuitement par différents moyens :

- Un clic droit sur un utilisateur ouvre le panneau représenté sur la partie gauche de la figure 10.9. En cliquant sur le bouton symbolisant un téléphone, l'utilisateur est invité à confirmer qu'il souhaite appeler l'ordinateur de son correspondant.

Figure 10.9

*Appeler l'ordinateur d'un contact*



- Un clic gauche sur un utilisateur ouvre le menu déroulant illustré sur la partie droite de la figure 10.9, sur lequel on remarque la fonction « Appeler l'ordinateur ».
- Un clic droit sur l'icône de Yahoo! Messenger dans la zone d'état système de Windows permet de sélectionner la fonction « Appeler l'ordinateur », qui présente la liste des contacts.

En sélectionnant l'un de ces choix, on ouvre automatiquement une fenêtre de dialogue avec le contact correspondant, et l'appel est initié. La barre d'outils de cette fenêtre indique la durée de l'appel en cours et permet de régler le volume sonore, de mettre l'appel en attente ou de le terminer.

## Appeler des lignes fixes et portables avec Yahoo! Voice

Fruit d'un partenariat avec la société californienne Dialpad, spécialisée dans la téléphonie sur IP, et dont elle a fait l'acquisition en juin 2005, Yahoo! propose dans sa nouvelle version de téléphoner sur des lignes fixes du réseau commuté traditionnel.

Appelé Yahoo! Voice, ce service s'utilise comme des crédits de minutes de communication à acheter par pack. Si l'utilisateur tente d'appeler un téléphone sans avoir les crédits suffisants, il est automatiquement redirigé sur une page lui permettant de créditer son compte.

Un appel téléphonique peut être initié des diverses façons suivantes :

- En sélectionnant « Appeler un numéro de téléphone à partir d'un contact ».
- En sélectionnant « Appeler un numéro de téléphone » du menu Actions d'une fenêtre de Yahoo! Messenger.
- Par le biais du raccourci clavier Ctrl + K.
- En sélectionnant « Appeler un numéro de téléphone » à partir de l'icône de Yahoo! Messenger de la zone d'état système.
- En saisissant directement un numéro de téléphone dans la barre située entre la barre d'icônes et la liste des contacts.

Dans tous ces cas, l'appel est lancé, et l'internaute voit s'afficher une fenêtre semblable à celle illustrée à la figure 10.10.



Figure 10.10

*Appel à partir du service Yahoo! Voice*

Yahoo! Voice propose des tarifs très compétitifs, qui varient selon les distances. Beaucoup d'entre eux sont plus intéressants que ceux du client dédié de Windows Live Messenger. La qualité audio est globalement bonne, même si des coupures intempestives peuvent survenir occasionnellement.

Comme WLM, Yahoo! ne propose aucun service fournissant un numéro d'appel entrant pour être joignable à partir d'un réseau téléphonique RTC. En revanche, les appels entrants provenant du réseau IP peuvent être distingués selon l'appelant. Ainsi, une sonnerie particulière retentit selon l'identité du correspondant qui appelle.

### Répondeur et historique d'appels

Une fonctionnalité particulièrement pratique du Messenger de Yahoo! réside dans la possibilité de laisser des messages audio à ses correspondants absents. Ce service est totalement gratuit.

Si un utilisateur tente de joindre un contact qui ne répond pas au bout d'une quarantaine de secondes, il est automatiquement redirigé vers la messagerie de ce dernier. Un court avertissement audio, en français, invite l'appelant à laisser un message. La barre d'outils de la fenêtre de conversation laisse place à la zone de contrôle du message audio illustrée à la figure 10.11. Ce message ne doit pas excéder deux minutes.

Figure 10.11

*Enregistrement d'un message vocal*



La personne appelée est avertie de l'arrivée d'un nouveau message sur l'interface principale de son Messenger par l'icône colorée représentant une bobine d'enregistrement qui s'active (comme pour la réception d'un nouvel e-mail). Pour consulter ses messages, il suffit de cliquer sur cette icône et de consulter les journaux de tous les appels audio qui sont sauvegardés.

L'interface comporte deux volets. Celui de gauche permet de filtrer les appels selon différents critères. Les messages correspondant au filtre sélectionné s'affichent dans le volet de droite. La section « Historique des appels » liste l'ensemble des appels, sans distinction.

Il est possible de filtrer cette liste selon des critères tels qu'une personne, un numéro de téléphone, ou les appels reçus (Appels entrants), émis (Appels sortants), sans réponse (Appels manqués) et audio (Messages vocaux).

En sélectionnant un message audio dans le volet de droite, les icônes de la barre supérieure deviennent actives et offrent différents moyens de contrôle du message : lecture, avance et recul de 7 secondes, contrôle du volume sonore, rappel de l'appelant, envoi d'un message instantané, ajout du contact, blocage de l'appelant et enfin suppression du message sélectionné.

Dans la partie inférieure gauche de la fenêtre, un lien direct donne accès à la page Web détaillant la facturation de son compte afin de connaître l'état de son crédit.

### Yahoo! Messenger sur un téléphone

À l'instar de Skype et WLM, le logiciel de messagerie de Yahoo! est implémenté sur certains téléphones afin d'être utilisable sans avoir à allumer son ordinateur.

La liste des contacts de son compte est affichée sur le poste téléphonique, et chacun de ces contacts est joignable dans les mêmes conditions tarifaires qu'avec l'ordinateur (gratuité pour les appels sur le réseau IP et tarif Yahoo! pour le réseau téléphonique commuté).

La figure 10.12 illustre un tel téléphone sans fil DECT proposé par le constructeur Siemens.

**Figure 10.12**

*Un téléphone intégrant Yahoo!  
Messenger*



Estampillé Y! Ready, ce téléphone a donné lieu à une certification Yahoo!. Celle-ci est avant tout un label commercial certifiant que le matériel est compatible avec le logiciel et validant son bon fonctionnement.

### Le partenariat Microsoft-Yahoo!

Yahoo! et Microsoft utilisent des protocoles propriétaires distincts, ce qui ne les rend pas interopérables. D'autres systèmes de messagerie, tel Jabber, que nous présentons en détail au chapitre 11, prônent un modèle libre et ouvert, dans lequel chacun est libre de choisir son système de messagerie et peut communiquer avec un utilisateur ayant choisi un autre système de messagerie. Ces concurrents reprochent à Skype, Microsoft et

Yahoo! d'user de leur suprématie pour contraindre leurs utilisateurs à rester cloisonnés dans leur réseau propriétaire respectif.

Les programmeurs n'ont pas attendu une réaction des grands éditeurs pour proposer des outils multiprotocoles capables de se connecter sur chacun de ces réseaux. Puisque les protocoles en question sont fermés, ces programmeurs ont dû user d'ingéniosité pour comprendre la logique et les messages utilisés dans ces logiciels. Pour cela, ils ont procédé à une écoute (*sniffing*) des communications réseau qu'engendre chacune des actions de ces logiciels. Ils en ont déduit les messages à envoyer pour reproduire ces actions.

Un logiciel tel que Wireshark (anciennement Ethereal), gratuit et disponible sous toutes les plates-formes courantes, permet une écoute très fine et plutôt pertinente de ces messages. Légalement, si la décompilation du programme lui-même est interdite sans l'accord des ayants droit, rien n'interdit d'analyser les échanges protocolaires à des fins de compatibilité. Ainsi, dans le cadre du droit à l'interopérabilité, il est parfaitement autorisé de procéder à la décompilation des interfaces de communication de n'importe quel logiciel, c'est-à-dire d'utiliser un sniffer sur un logiciel afin d'en déduire une forme de langage générique qui sera implémentée et adaptée dans un logiciel tiers.

Ce mode artisanal échappe au contrôle des majors de l'industrie et à leur guerre ouverte pour imposer leur standard. Pour ces majors, l'utilisation de logiciels tiers dans leur propre réseau de messagerie représente une double perte : non seulement le contraignant bandeau publicitaire finançant le service (et donc la maintenance des serveurs) disparaît dans les logiciels tiers, mais la gamme de services complémentaires (téléphonie, logos personnalisés, musique, envoi de SMS, etc.) qu'ils proposent en mode payant, n'est souvent plus accessible.

Conscients de ces difficultés naissantes, Yahoo! et Microsoft, bien que concurrents, ont décidé de travailler ensemble pour trouver une issue commune. Pour éviter qu'un utilisateur n'utilise un programme tiers pour se connecter à leur réseau de messagerie, ils ont décidé, en octobre 2005, d'offrir une couverture plus large en rendant compatibles leurs réseaux de messagerie. L'accord a pris effet en août 2006.

Un utilisateur de l'un de ces deux logiciels peut donc ajouter un utilisateur de la plate-forme concurrente de manière transparente, comme s'il s'agissait de la même plate-forme. Comme l'illustre la figure 10.13, l'ajout d'un nouveau contact dans Yahoo! Messenger permet de spécifier si ce dernier est un contact Yahoo!, LCS (la messagerie professionnelle de Microsoft) ou WLM/MSN.

Aux duplications de comptes près (certainement importantes), c'est potentiellement une base de données d'environ 390 millions de contacts qui peut ainsi circuler entre les deux réseaux. En s'alliant, les deux sociétés creusent ainsi l'écart avec leurs concurrents, notamment avec la messagerie AIM d'AOL, et se placent en position encore plus dominante sur le marché.

**Figure 10.13**

*Ajout d'un contact WLM  
à sa liste Yahoo!*



Ce partenariat technologique reste cependant limité, et les conversations audio et vidéo sont encore absentes. La compatibilité dans le cadre de ce partenariat a été portée au même niveau que celle offerte par les logiciels tiers. Nul doute que cette association offrira de nouvelles possibilités, au moins pour suivre les avancées des concurrents.

## Conclusion

Compatibilité n'est pas synonyme d'interopérabilité. En dépit d'une confusion répandue, les deux termes sont différents. La nuance concerne l'utilisation ou non de protocoles standards.

La compatibilité permet la gestion de plusieurs produits utilisant des protocoles différents. Elle fournit les outils pour permettre à ces derniers de communiquer entre eux par un mécanisme de traduction d'un produit vers un autre. La conversion d'un protocole de communication vers un autre est inhérente au mécanisme de compatibilité.

L'interopérabilité est un concept plus général, qui témoigne de bases communes dans le protocole de communication choisi par différents produits. Il n'y a pas de traduction d'un produit vers un autre, puisque tous deux utilisent les mêmes spécifications protocolaires. Pour être interopérables, les produits doivent s'appuyer sur un protocole ouvert et standardisé.

Ce n'est pas le cas des logiciels de messagerie de Yahoo! et de Microsoft, dont les protocoles sont propriétaires et fermés. Leur partenariat relève donc de la compatibilité et non de l'interopérabilité.

## Jabber et Google Talk

---

Jabber est une plate-forme libre développée pour assurer l'interopérabilité des logiciels de messagerie instantanée et fédérer les réseaux de messagerie autour de normes communes.

Comme la ToIP est devenue le pendant de la messagerie instantanée, Jabber a étendu son modèle pour traiter la gestion des flux multimédias.

Google s'en est largement inspiré et a participé à l'élaboration de protocoles de signalisation de la voix permettant de proposer un client de messagerie et de téléphonie reposant sur Jabber en rupture avec tous les autres clients.

### Jabber

L'approche de Jabber en matière de messagerie instantanée et de téléphonie se distingue de celle de ses concurrents en ce qu'elle se rapproche du courrier électronique.

Ses utilisateurs peuvent communiquer entre eux indépendamment de leur client logiciel et de leur serveur de traitement. Chacun d'eux reste libre de choisir son serveur, qui lui fournit une adresse afin de s'identifier, et d'utiliser le client qu'il souhaite parmi la gamme des logiciels compatibles disponibles. La clé de voûte de cette plate-forme est d'offrir un protocole libre et standardisé.

Jabber définit un ensemble de protocoles qui utilisent des normes communes et ouvertes laissant libres l'implémentation et l'ergonomie des logiciels clients tout en garantissant l'interopérabilité de leurs communications. Il ne s'agit donc pas à proprement parler d'un logiciel client, ni d'un logiciel serveur, mais plutôt d'une plate-forme dédiée à la messagerie instantanée et conçue pour être ouverte, rapide, facile à utiliser et à étendre pour de nouveaux services, parmi lesquels la téléphonie sur IP.

Les premières technologies Jabber ont été développées par Jérémie Miller en 1998 et ont été publiées pour la première fois en mai 2000 avec *jabberd*, première version du serveur. Le nom Jabber vient d'un vieux mot anglais évoquant une discussion rapide et presque inintelligible.

Jabber repose sur le protocole XMPP, qui a été conçu au sein de la communauté libre Jabber avant d'être soumis à l'IETF pour standardisation et évolution. À ce socle fondamental, peuvent être ajoutées un ensemble de briques destinées à favoriser l'évolution du protocole de manière modulaire. Ces améliorations sont connues sous le nom de XEP (Jabber Enhancement Proposals). Leur développement et maintenance sont gérés par la JSF (Jabber Software Foundation), créée en 2001.

### Architecture de Jabber

Jabber innove en proposant un modèle distribué dans son approche de la messagerie instantanée.

Dans les principaux protocoles de messagerie instantanée, un serveur unique (éventuellement avec redondance pour supporter la charge) gère les communications entre utilisateurs. À la différence de ce modèle centralisé, Jabber repose sur un réseau de serveurs qui assurent la connectivité entre les utilisateurs.

Chacun de ces serveurs pris individuellement n'a pas la connaissance de l'ensemble des utilisateurs présents dans le réseau, mais a la capacité d'interagir et de dialoguer avec les autres afin d'avoir une vue complète du réseau d'utilisateurs. Ce modèle distribué est assez proche de celui d'Internet, puisque les serveurs de messagerie et de résolution de nom (DNS) fonctionnent de manière analogue. En cela, Jabber se démarque radicalement de ses concurrents propriétaires.

Jabber repose sur un modèle de type client-serveur distinguant les trois types d'entités logiques suivants :

- Clients Jabber. *A priori*, les clients sont des composants extrêmement simples. Ils sont interchangeables, et un utilisateur peut décider du jour au lendemain d'utiliser un autre client plus fonctionnel, moins lourd ou tout simplement plus esthétique.
- Serveurs Jabber. C'est sur eux que repose toute la complexité logique de Jabber. Ils ont pour tâche principale la gestion d'un domaine particulier et permettent la création à la demande de comptes dans le domaine qu'ils gèrent. Les serveurs stockent l'ensemble des informations et propriétés associées aux comptes, notamment toutes leurs coordonnées et éventuellement toutes sortes d'informations textuelles complémentaires. Ils fournissent en outre des services pouvant apporter des fonctionnalités innovantes. Pour faire communiquer deux utilisateurs associés à deux serveurs différents (et donc appartenant à deux domaines différents), les serveurs Jabber doivent pouvoir communiquer entre eux.

- Passerelles. Les passerelles permettent de joindre un autre type de réseau de messagerie instantanée, utilisant un protocole non compatible avec celui utilisé par Jabber. Il s'agit le plus souvent d'une entité logique et non physique prise en charge par le serveur, le plus souvent sous forme de modules complémentaires. Le terme *transport* est indifféremment utilisé à la place de passerelle pour désigner cette fonctionnalité. Par exemple, si l'on souhaite converser avec des utilisateurs WLM ou Yahoo, des passerelles doivent être mises en place pour assurer la jonction de ces réseaux et les communications interréseau. Grâce à ces passerelles, tout se passe comme si l'utilisateur était connecté au réseau concurrent de manière transparente.

L'avantage immédiat de concentrer l'intelligence sur les serveurs, et non sur les clients, est de permettre l'évolution des protocoles de manière transparente. Autrement dit, lorsqu'une implémentation est améliorée et qu'on souhaite en disposer, il n'est pas nécessaire de mettre à jour tous les clients logiciels distribués. Il suffit de reporter les modifications au niveau du serveur. Les clients sont automatiquement adaptés, car ils sollicitent un service générique dont l'implémentation est laissée à la convenance du serveur.

Comme l'illustre la figure 11.1, le contrôle et la sécurisation des communications des utilisateurs peuvent être gérés à un niveau interne avant d'être généralisés à l'ensemble du réseau. Dans le cadre d'une entreprise, toutes les communications peuvent être prises en charge par un serveur de l'entreprise, assurant, par exemple, les fonctionnalités de messagerie instantanée, d'annuaire et de téléphonie pour ses utilisateurs, le tout en respectant les politiques de sécurité en cours dans l'entreprise.

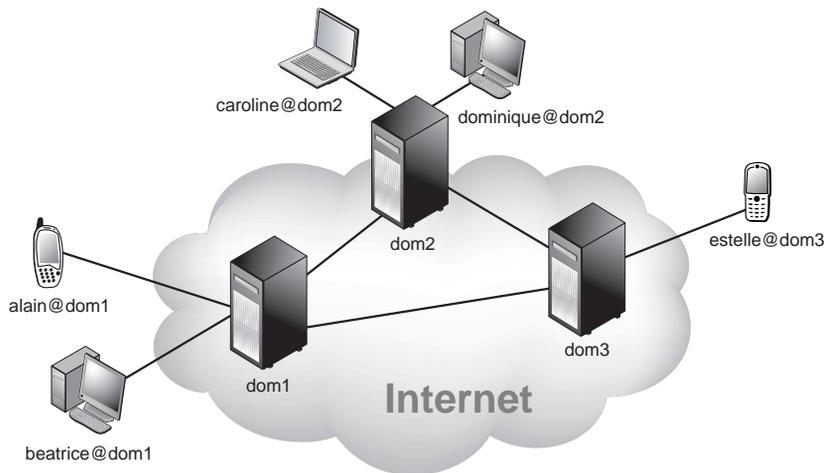


Figure 11.1

*Le modèle décentralisé de Jabber*

Jabber permet ainsi de mettre en place un système de communication local dans une entreprise avec une sécurisation parfaitement maîtrisée. En cas de montée en charge trop

importante pour être supportée par un unique serveur, il est possible d'absorber la charge sans provoquer d'interruption de service. Il suffit pour cela ajouter d'autres serveurs gérant le même domaine. On parle en ce cas d'architecture *clustérisée*, chaque serveur constituant un nœud et formant globalement un cluster, ou grappe, grâce auquel tous les nœuds peuvent communiquer entre eux.

De cette manière, la montée en charge peut être définie ultérieurement, sans remettre en cause l'architecture existante. En outre, une architecture clustérisée permet de mettre en œuvre une redondance des nœuds afin de continuer à faire fonctionner un service même en cas de panne d'un des nœuds du cluster.

En contrôlant les communications par un serveur interne, une entreprise n'est pas reclusée pour autant et peut, au prix d'une simple configuration du serveur, se connecter à un autre serveur. Cela permet d'étendre les communications à l'ensemble des utilisateurs gérés par ces serveurs.

Le parc de serveurs Jabber pouvant s'agrandir à l'infini, les communications peuvent progressivement relier de nouveaux utilisateurs. L'avantage est que chaque serveur peut décider de communiquer avec des serveurs bien déterminés, qu'il s'agisse de sites distants d'une entreprise ou de serveurs de confiance, sécurisés et reconnus.

## ***XMPP (eXtensible Messaging and Presence Protocol)***

Le standard XMPP est un protocole extensible de messagerie et de présence conçu par la communauté Jabber pour formaliser l'échange de flux de messages à contraintes temps réel.

XMPP est à Jabber ce que le protocole HTTP est au Web. Il a pour vocation d'unifier autour de son langage l'ensemble des protocoles de messagerie instantanée. Comme son nom l'indique, la première application de ce protocole concerne la messagerie instantanée et le service de présence. Mais le protocole ne se réduit pas à cela et couvre de nombreuses autres applications, comme le transfert de fichiers ou la gestion de la voix.

Le protocole XMPP s'appuie sur des communications en langage textuel XML, ce qui lui confère une grande souplesse d'utilisation sur Internet et facilite sa compréhension. Le document de référence pour le protocole XMPP est la RFC 2779 (Instant Messaging/Presence Protocol Requirements), écrite en février 2000, qui pose les bases de spécifications pour les protocoles de messagerie et de présence. Le protocole XMPP a été normalisé et est maintenu par l'IETF. Il a été décrit en octobre 2004 par quatre RFC, répertoriées de 3920 à 3923 (*voir le tableau 11.1*).

Son principal concurrent est le protocole SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions), soutenu par Microsoft et également proposé par l'IETF comme une application de SIP aux systèmes de messagerie instantanée.

Tableau 11.1 – RFC décrivant le protocole XMPP

RFC	Description
RFC 3920 : <i>eXtensible Messaging and Presence Protocol (XMPP): Core</i>	La RFC 3920 explicite les fonctionnalités élémentaires du protocole XMPP, en particulier la connexion entre les entités et l'échange de messages entre les clients et les serveurs ainsi qu'entre serveurs. Les communications entre les serveurs assurent le partage des identités et profils des utilisateurs. Deux serveurs appartenant à deux entreprises différentes peuvent ainsi communiquer entre elles et partager les informations relatives aux comptes de leurs utilisateurs. Ces derniers peuvent communiquer entre eux. De manière générale, on peut imaginer qu'au prix d'une simple configuration, tous les serveurs exploitant XMPP et ne dépendant pas forcément d'une même organisation puissent interagir et échanger leur base de données d'utilisateurs.
RFC 3921 : <i>eXtensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence</i>	La RFC 3921 s'intéresse aux applications de messagerie instantanée et de présence. Ces deux applications sont caractéristiques du protocole XMPP, mais non exclusives. La RFC explique notamment la gestion des listes de contacts et les possibilités de bloquer des contacts dans la liste.
RFC 3922 : <i>Mapping the eXtensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)</i>	La RFC 3922 explique comment faire la jonction entre les messages des protocoles XMPP et CPIM. CPIM est un protocole issu du groupe de travail IMPP (Instant Messaging and Presence Protocol) de l'IETF, qui vise à mettre en place une plate-forme d'interopérabilité entre les messageries instantanées et les systèmes de présence. C'est un modèle abstrait auquel le protocole XMPP peut s'intégrer. La RFC se propose d'établir des relations entre un service utilisant XMPP et un autre n'utilisant pas XMPP mais respectant le modèle général IMPP. Cette interaction sollicite la participation d'une entité logique chargée d'effectuer la translation d'un message d'un protocole (XMPP ou CPIM) en un message équivalent de l'autre protocole (XMPP ou CPIM).
RFC 3923 : <i>End-to-End Signing and Object Encryption for the eXtensible Messaging and Presence Protocol (XMPP)</i>	La RFC 3923 détaille les méthodes de signalisation de bout en bout et les mécanismes permettant la sécurisation des messages et des données manipulées par XMPP.

## XEP (XMPP Enhancement Proposals)

Les RFC 3920 à 3923 définissent les fondements du protocole XMPP. Leur implémentation en totalité est donc en principe indispensable pour le bon fonctionnement de l'infrastructure déployant XMPP. Comme indiqué précédemment, des améliorations peuvent toutefois être proposées grâce aux XEP (XMPP Enhancement Proposals), anciennement appelées JEP (Jabber Enhancement Proposals).

Ces propositions enrichissent le protocole de base en spécifiant formellement des ajouts fonctionnels. Elles sont à considérer comme des extensions et ne sont par conséquent pas indispensables. Chaque serveur est libre de les implémenter ou non, selon les services que l'administrateur souhaite proposer à ses utilisateurs. De même, les clients peuvent supporter certaines XEP et pas d'autres. Tous les clients et serveurs ne disposent donc pas des mêmes caractéristiques et possibilités.

Le tableau 11.2 récapitule les principales fonctionnalités complémentaires proposées dans les XEP.

**Tableau 11.2 – Principales fonctionnalités complémentaires des XEP**

XEP	Fonctionnalité
XEP-0016	Gestion des contacts privés
XEP-0030 et XEP-0128	Découverte de services dans le réseau
XEP-0045	Conférence entre plusieurs utilisateurs
XEP-0080	Géolocalisation des utilisateurs
XEP-0084	Gestion des avatars
XEP-0096	Gestion du service de transfert de fichiers
XEP-0107	Gestion des humeurs des contacts (l'indication d'une humeur est donnée directement par un utilisateur et peut se refléter, par exemple, par une image, un smiley, une couleur, etc.).
XEP-0116	Cryptage des sessions
XEP-0126	Gestion du statut invisible
XEP-0127	Alertes et notifications à l'utilisateur d'événements spécifiques
XEP-0136	Archivage des messages
XEP-0154	Gestion des profils d'utilisateurs
XEP-0159 et XEP-0161	Gestion du SPIM (Spam Over IM), ou Spam sur messagerie instantanée, incluant les mécanismes de blocage et les journaux
XEP-0166	Gestion des flux multimédias (Jingle)
XEP-0167	Description des formats audio supportés
XEP-0172	Gestion des pseudonymes
XEP-0180	Description des formats vidéo supportés

Il existe près de 200 XEP, avec des actualisations et des ajouts en permanence, mais nombreuses sont celles qui restent à l'état de XEP et ne sont pas implémentées. La liste exhaustive des XEP publiées est disponible sur le site de Jabber, à l'adresse <http://jabber.org/jeps/>.

### Jingle, la XEP de Jabber dédiée à la ToIP

Originellement, Jabber ne se préoccupait que de messagerie instantanée. La ToIP n'est traitée que comme une extension pour répondre à la demande des utilisateurs réclamant ce service. De ce fait, la gestion de la voix n'est pas décrite par le protocole XMPP, et il est parfaitement envisageable d'utiliser SIP pour offrir un service de téléphonie conjointement avec l'utilisation de Jabber. Certains clients logiciels s'y sont d'ailleurs essayés, comme OpenWengo.

C'est donc sous la forme d'une extension XEP que la ToIP peut être implémentée sur la plate-forme Jabber. Publiée en octobre 2005 sous le nom de Jingle, elle est référencée par la documentation XEP-0166, étendue par la XEP-0167 pour les formats de données.

Jingle remplace l'obsolète protocole TINS (Transport for Initiating and Negotiating Sessions), défini dans la XEP-0111, qui n'a connu qu'une seule implémentation, appelée Jabbin, peu couronnée de succès. Jingle a été conçu de concert avec les équipes de développeurs de Google.

On raconte que les travaux de Google et ceux de Jabber se sont croisés lorsque les développeurs se sont aperçus de la forte parenté de leurs avancées en matière de définition d'un protocole de session multimédia. Les documentations de Jabber ayant été rendues publiques, Google a proposé d'unir les forces pour la conception de ce protocole. C'est ainsi que, en décembre 2005, une implémentation de Jingle nommée *libjingle* a été réalisée par Google et rendue disponible au public sous la forme d'une bibliothèque libre (accessible à l'adresse <http://code.google.com/apis/talk/index.html>).

Plusieurs logiciels ont intégré cette implémentation, à commencer par Google Talk, le logiciel de messagerie de Google, que nous détaillons ultérieurement dans ce chapitre, mais également PSI et Kopete.

Notons qu'avec Jingle, Jabber ne gère pas uniquement la voix, mais n'importe quel contenu binaire, y compris multimédia. Ainsi, Jingle devrait à terme permettre l'échange de vidéos (XEP-0180).

## Utilisation

Jabber n'est ni un client, ni un protocole mais plutôt une plate-forme conçue par l'association des RFC définissant le protocole XMPP et des améliorations proposées dans les XEP. Il a été développé par une communauté libre très active et a été implémenté par plusieurs logiciels clients et serveurs, parfois libres, parfois propriétaires.

Cette section présente les serveurs et clients les plus courants et les possibilités d'extension de Jabber.

### Choix du serveur

Le serveur est l'entité en charge de la gestion des comptes des utilisateurs. Il fournit à ces derniers les services auxquels ils ont souscrit et assure notamment les fonctions de localisation et de mise en relation des contacts.

Les utilisateurs se connectent à leur serveur en utilisant l'un des clients logiciels que nous détaillons plus loin dans ce chapitre. Il leur faut préalablement choisir un serveur. Ce choix n'est pas nécessaire pour des logiciels tels que Skype, WLM et Yahoo!, car il est implicite et contraint. Jabber n'étant pas un modèle propriétaire, il ne dispose pas de serveur prédéfini. Aucun serveur n'est affecté par défaut aux utilisateurs Jabber, et il est même à leur charge d'en sélectionner librement un, parmi tous ceux que la communauté Jabber laisse à leur disposition.

Pratiquement, cette phase n'implique qu'un simple choix de serveur dans une liste proposée, qui doit être ensuite validé au niveau du client.

## JID (Jabber ID)

Un utilisateur a pour identifiant un pseudonyme auquel est associé le nom du serveur Jabber qui le prend en charge. L'identifiant d'un utilisateur obéit donc au format suivant :

*pseudonyme@serveur\_jabber\_choisi*

Cet identifiant est unique. Le choix d'un serveur Jabber se reflète ainsi dans son identifiant de connexion. Une fois le serveur Jabber spécifié dans l'identifiant de l'utilisateur, ce dernier ne peut changer de serveur sans modifier son identifiant. C'est exactement comme pour un serveur de messagerie, qui est mentionné après l'arobase d'une adresse e-mail.

Pour choisir son serveur Jabber, on peut consulter la liste des serveurs publics et gratuits accessible sur le site de Jabber (<http://www.jabber.org>) à la section *Servers*.

Voici quelques serveurs Jabber connus :

- APINC (association pour la promotion de l'Internet non commercial). Localisé en France, ce serveur constitue un choix pertinent. Il dispose des passerelles vers la plupart des réseaux des autres clients de messagerie (AIM, ICQ, MSN et Yahoo!) et permet d'utiliser un très large choix de noms de domaines (parmi lesquels les domaines *jabber.fr* et *im.apinc.org*, dont l'APINC est propriétaire). Il est possible d'utiliser son propre nom de domaine si l'on en possède un. La liste des domaines disponibles est référencée à la section « liste des domaines » du site de l'APINC, à l'adresse <http://jabber.apinc.org>.
- Jabber.org. C'est le serveur le plus célèbre, mais sa réputation en fait son principal défaut puisqu'il est particulièrement sollicité, et donc parfois ralenti. Il centralise en outre énormément de comptes, ce qui est paradoxal, compte tenu de la philosophie distribuée de Jabber, d'autant qu'il ne dispose pas de passerelles vers les autres clients de messagerie. Le serveur est géré par la JSF. Son domaine usuel est *jabber.org*.
- Amessage. Ce serveur localisé en Allemagne est plutôt bien maintenu et dispose de quelques transports. Il offre des noms de domaines (notamment *amessage.info*). Son principal attrait est de disposer d'un grand nombre de services, comme l'envoi de SMS, la gestion de blogs et la mise en place d'annuaires. Son site, en langue anglaise ou allemande uniquement, est disponible à l'adresse <http://web.amessage.info>.

Tous ces serveurs fonctionnent globalement très bien, et il en existe beaucoup d'autres. Il s'agit le plus souvent d'initiatives de bénévoles voulant gérer un serveur communautaire. De fait, leur pérennité n'est pas garantie, et leur stabilité peut parfois faire défaut, contrairement aux logiciels commerciaux grand public, qui disposent de moyens considérables. Si les pannes sont plus fréquentes, l'équipe chargée de la gestion du serveur est en contrepartie plus accessible et disposée à apporter des évolutions aux services proposés.

Le choix du serveur n'est pas anodin, à la fois pour les fonctionnalités et la qualité de service offertes, mais aussi parce que les administrateurs ont potentiellement la possibilité de lire les contenus des messages, la liste des contacts et le mot de passe associé au

compte, puisque l'ensemble de ces données transite par ces serveurs. En cas de doute, il est toujours possible de crypter ses messages afin d'éviter toute tentative frauduleuse d'interception.

### Monter son propre serveur Jabber avec ejabberd

Jabber ne limitant pas les possibilités de choix du serveur, il est possible d'exploiter un serveur privé. Pour monter son propre serveur, un bon choix logiciel est fourni par ejabberd.

Conçu en 2002, le serveur ejabberd a été écrit principalement en langage Erlang, défini par Ericsson pour les applications distribuées, de type temps réel, avec résistance aux pannes et gestion optimisée des accès concurrents, d'où son nom complet Erlang Jabber Daemon.

Le serveur ejabberd fonctionne sous presque toutes les plates-formes (Windows, MacOS, Linux, BSD, Solaris, etc.) et est disponible en licence GNU GPL 2.

L'atout principal d'ejabberd est sa robustesse, sa stabilité ayant été testée avec des montées en charge de plusieurs milliers d'utilisateurs. Il est notamment utilisé sur le serveur de l'APINC.

ejabberd est téléchargeable sur le site de l'éditeur, à l'adresse <http://ejabberd.jabber.ru>.

### Choix du client

La souplesse offerte dans le choix des serveurs permet en théorie de distribuer la charge tout en assurant l'interopérabilité entre serveurs. Mais choisir un serveur n'a que peu d'intérêt pour le client final. Ce dernier souhaite avant tout avoir le maximum de services disponibles sur une plate-forme unique à laquelle il se connecte. Ce n'est pas le cas avec Jabber, dont l'approche communautaire et non commerciale repose sur un modèle artisanal, dépendant de la bonne volonté d'individus qui œuvrent à l'émergence de normes communes.

La valeur ajoutée de Jabber s'exprime dans la très large diversité des clients disponibles. Par défaut, l'utilisateur n'a aucun choix imposé, contrairement à tous les autres logiciels de messagerie propriétaire. Le choix d'un serveur n'a pas d'incidence directe sur celui du client, sauf à vérifier que ce dernier dispose des implémentations permettant d'utiliser certaines fonctionnalités du serveur.

Parmi les nombreux clients compatibles Jabber, citons notamment les suivants, très connus et gratuits :

- Psi, Gaim, Coccinella et Jabbin, pour Windows, Linux et Macintosh ;
- Exodus, pour Windows ;
- Gabber et Gossip, pour Linux.

Une liste très large est disponible à la section Clients du site de Jabber. Une fois un client choisi, il est parfaitement envisageable d'évoluer sans contrainte vers un nouveau client, tout en conservant son identifiant de compte Jabber.

Mieux encore, comme le protocole est libre, n'importe quel programmeur peut implémenter son propre client et le redistribuer éventuellement ensuite. Les sources deviennent donc particulièrement riches.

### Configuration d'un client Jabber

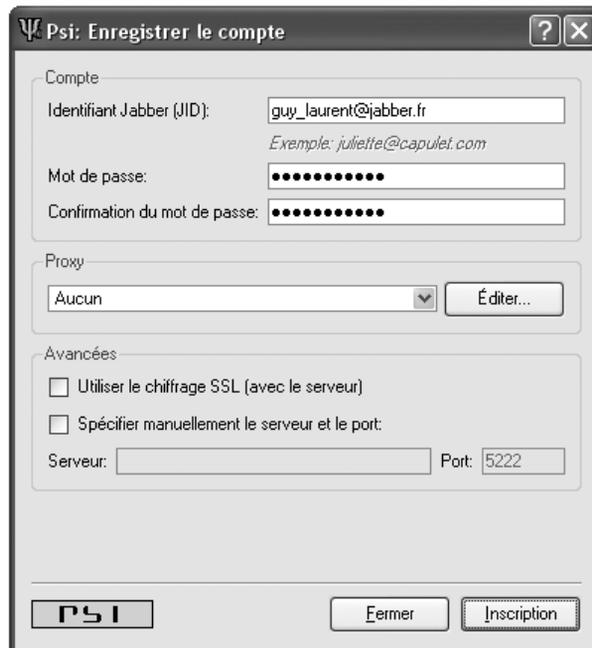
Nous allons détailler quelques éléments de configuration du client Psi, qui valent aussi pour le client Gaim, au libellé des fonctions près.

Après avoir téléchargé et installé Psi sur le site de l'éditeur (<http://psi.affinix.com>), le lancement du logiciel invite l'utilisateur à renseigner son nom, qui correspondant à l'identifiant d'un compte, existant ou non. Il s'agit du nom d'un profil, car Psi gère les profils multiples et permet à plusieurs utilisateurs de se connecter en même temps sur la même interface. Le nom spécifié est seulement utile pour l'utilisateur, qui peut ainsi discerner chacun des profils qu'il a créés.

Nous considérons que nous ne disposons pas encore d'un compte Jabber et que nous souhaitons en créer un. En sélectionnant la case « Enregistrer un nouveau compte » puis en cliquant sur le bouton Ajouter, la fenêtre d'inscription au serveur illustrée à la figure 11.2 s'ouvre.

Figure 11.2

Inscription au serveur



The image shows a dialog box titled "Psi: Enregistrer le compte" with a question mark and close button in the title bar. The dialog is divided into three sections: "Compte", "Proxy", and "Avancées".

- Compte:** Contains three input fields. The first is "Identifiant Jabber (JID):" with the value "guy\_laurent@jabber.fr" and a small example text "Exemple: juliette@capulet.com" below it. The second is "Mot de passe:" and the third is "Confirmation du mot de passe:", both filled with dots.
- Proxy:** Contains a dropdown menu with "Aucun" selected and an "Éditer..." button to its right.
- Avancées:** Contains two checkboxes: "Utiliser le chiffage SSL (avec le serveur)" and "Spécifier manuellement le serveur et le port:". Below these are two input fields: "Serveur:" (empty) and "Port:" with the value "5222".

At the bottom of the dialog, there is a "PSI" logo on the left, and two buttons: "Fermer" and "Inscription".

L'utilisateur est invité à saisir un identifiant et un mot de passe associé. C'est ici que le choix du serveur Jabber est mentionné. Ce choix peut se faire en utilisant la liste des serveurs disponible sur le site de Jabber, mais mieux vaut utiliser un serveur reconnu, comme ceux que nous avons mentionnés précédemment, comme celui de l'APINC.

Le site Web de l'APINC fournit la liste des domaines que le serveur met à disposition des utilisateurs, par exemple le domaine *jabber.fr*. L'identifiant Jabber (JID) a dans ce cas le format *nom\_d\_utilisateur@jabber.fr*.

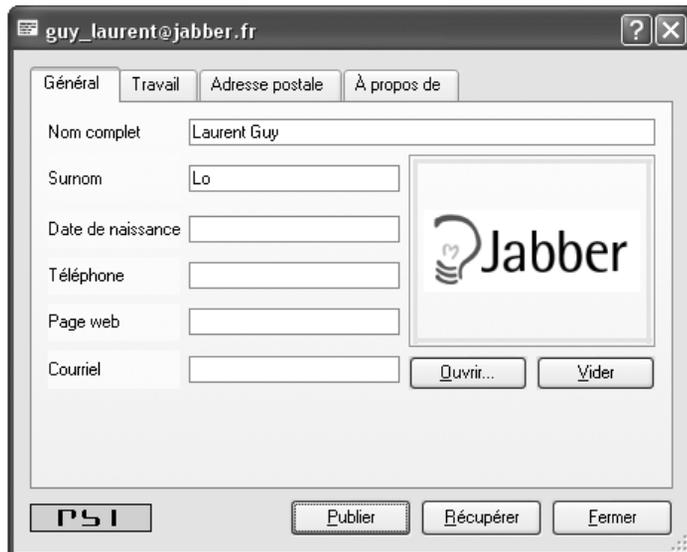
Un message d'erreur est renvoyé si le nom d'utilisateur est déjà utilisé par un autre membre. Choisissons pour identifiant *guy\_laurent*, et associons-y un mot de passe de notre choix, répété par sécurité à la ligne suivante. Ignorons les paramètres de connexions particulières, tels que les connexions par un serveur proxy ou les connexions différentes de celles s'effectuant sur le port par défaut (port 5222).

En validant la procédure par le bouton Inscription, le logiciel client tente de se connecter au serveur Jabber de l'APINC afin d'y effectuer la création du compte requis. Si le serveur est trouvé et que l'identifiant est disponible, un message annonçant que l'inscription au serveur s'est correctement déroulée s'affiche. Dans le cas contraire, un message d'erreur est retourné.

La dernière étape correspond à l'enregistrement d'une carte virtuelle, ou VCard, rassemblant l'ensemble des informations que l'on souhaite diffuser aux autres utilisateurs (voir figure 11.3). Optionnelle, cette étape peut être renseignée ultérieurement ou partiellement selon les informations que l'on souhaite partager avec les autres utilisateurs.

Figure 11.3

Renseignement d'une VCard



The screenshot shows a window titled "guy\_laurent@jabber.fr" with a standard OS title bar (help, close). The window contains a form for creating a VCard. At the top, there are four tabs: "Général" (selected), "Travail", "Adresse postale", and "À propos de". The form fields are as follows:

- Nom complet: Laurent Guy
- Surnom: Lo
- Date de naissance: (empty)
- Téléphone: (empty)
- Page web: (empty)
- Courriel: (empty)

On the right side of the form, there is a Jabber logo (a lightbulb with a heart inside) and the text "Jabber". Below the logo are two buttons: "Ouvrir..." and "Vider". At the bottom of the window, there are three buttons: "PSI", "Publier", "Récupérer", and "Fermer".

L'enregistrement de cette fiche personnelle (et, de manière générale, de toutes les options du compte, incluant la liste des contacts), se fait directement au niveau du serveur.

De cette manière, il devient possible d'accéder à son compte à partir de n'importe quelle autre station, simplement avec son identifiant et son mot de passe.

Sur la figure 11.3, le bouton Récupérer permet d'importer localement l'ensemble des paramètres enregistrés au niveau du serveur. Au contraire, en cliquant sur le bouton Publier, l'ensemble des informations est envoyé et sauvegardé sur le serveur.

L'interface principale de Psi s'affiche alors, comme illustré à la figure 11.4. À ce stade, le client est généralement hors connexion. Il lui faut modifier son statut pour se connecter au serveur.

Figure 11.4

Interface de Psi



Presque simpliste, cette interface recèle néanmoins quantité d'options et de paramètres de configuration. Les quelques boutons par défaut permettent de spécifier les catégories de contacts que l'on souhaite voir s'afficher. En bas de l'interface, les deux boutons avec une flèche descendante sont en fait des menus déroulants.

Le menu de droite permet de sélectionner son statut de connexion. C'est celui à utiliser pour se connecter au réseau Jabber. On peut sélectionner, par exemple, le mode En-Ligne ou le mode Invisible. Le menu de gauche, qui reprend le logo du logiciel Psi (du nom de la lettre grecque *psi*) est le menu principal du logiciel. Il offre de nombreux paramètres et options de configuration.

Ce menu permet notamment de spécifier toutes les informations de connexion. Pour accéder à la création d'un compte (mais aussi à sa modification ou à sa suppression), comme on l'a fait précédemment, il faut sélectionner « Configuration des comptes » puis cliquer sur le bouton Ajouter.

La personnalisation de l'interface peut se faire *via* la fonction « Configurer les barres d'outils » ou *via* l'onglet Affichage de la fonction Options. L'interface des options offre un grand nombre de possibilités de configuration, notamment des alertes ou des choix de sons sur événements, ou encore la possibilité d'afficher la traditionnelle barre de menus. Notons également la possibilité de rejoindre un groupe de discussion dont on connaît l'adresse du serveur et le nom de la salle de discussion.

## Services complémentaires

Des services complémentaires peuvent être proposés par le serveur à ses utilisateurs. Il s'agit le plus souvent de passerelles vers les autres réseaux de messagerie, mais il peut tout aussi bien s'agir de services de blog ou de calendrier partagés entre plusieurs utilisateurs. Ces fonctionnalités sont toutefois rarement prises en charge en totalité sur tous les serveurs.

Pour connaître la liste des fonctionnalités disponibles, il suffit de se rendre sur le site Web du serveur choisi et rechercher la section listant les fonctionnalités prises en charge. Si le client et le serveur le supportent, il est aussi possible d'utiliser son client pour interroger son serveur sur ses fonctions.

Après une brève inscription au transport souhaité, le service devient exploitable. La liste des services disponibles est accessible en sélectionnant « Gestion des services » dans le menu principal, en bas à gauche du logiciel. Nous verrons, plus loin, un exemple de transport pour l'ajout d'un contact appartenant à une messagerie non-Jabber.

### Ajouter des contacts WLM ou Yahoo!

L'ajout d'un compte d'une messagerie propriétaire *via* un client Jabber présuppose que le serveur Jabber dispose d'une passerelle vers le réseau de messagerie du compte utilisateur à ajouter. Autrement dit, pour ajouter un contact WLM, un utilisateur doit utiliser un serveur intégrant une passerelle WLM. C'est généralement clairement indiqué et mis en avant par le site du serveur utilisé.

Les contacts WLM comme Yahoo! ont la particularité d'utiliser les e-mails des utilisateurs comme identifiants. Ils comportent donc tous une arobase. Il se trouve que Jabber exploite lui-même l'arobase pour indiquer un nom de domaine et le serveur Jabber en charge du compte. Pour éviter toute confusion, il faut donc remplacer le symbole @ par celui du pourcentage (%) et faire précéder le tout d'un identifiant précisant la messagerie utilisée pour le compte.

Par exemple, un contact ayant pour identifiant :

*mon\_identifiant\_sur\_wlm@hotmail.com*

doit être ajouté par :

*mon\_identifiant\_sur\_wlm%hotmail.com@msn*

De plus en plus de clients effectuent automatiquement ces conversions à la place de l'utilisateur en reconnaissant le réseau de messagerie du contact qui est ajouté. C'est le cas notamment du client Psi.

La liste de contacts d'un utilisateur n'est cependant pas importable d'un réseau concurrent vers Jabber. Cela occasionne une lourde tâche manuelle pour l'ajout de ces contacts. Mais cette contrainte vaut aussi pour les autres réseaux de messagerie.

Généralement, seule la fonctionnalité de messagerie instantanée des protocoles concurrents de Jabber est disponible en passant par une passerelle, ce qui exclut la gestion des

communications vocales et vidéo, comme le transfert de fichiers ou toute autre fonctionnalité maison des autres éditeurs.

Jabber n'a pas vocation à remplacer les services concurrents, mais plutôt à proposer une solution globale résolvant les problèmes d'incompatibilité. Les méthodes permettant d'accéder aux réseaux propriétaires concurrents ne sont qu'un moyen de valoriser cette plate-forme.

Même si, théoriquement, le modèle peut être étendu selon les besoins des utilisateurs et les perspectives d'évolution des développeurs, dans la pratique, il est peu probable que cela dépasse le service de base de la messagerie instantanée. La principale raison à cela est que les protocoles concurrents sont propriétaires et qu'ils sont susceptibles d'évoluer en permanence, rendant inopérables les passerelles et imposant aux développeurs de lourds travaux pour maintenir la compatibilité. Un utilisateur dont les contacts appartiennent tous au même réseau n'a donc pas forcément à passer par Jabber.

### Coopération entre serveurs

Lorsqu'un utilisateur se connecte par le biais d'un logiciel client compatible, une connexion au serveur Jabber choisi s'effectue. Elle actualise avant tout le statut de cet abonné en spécifiant qu'il s'est connecté (sauf s'il a demandé à rester masqué). Cela permet aux autres utilisateurs d'en être informés, mais aussi de récupérer le profil de l'utilisateur et l'ensemble des paramètres qui y sont associés, notamment la liste de ses contacts.

À ce stade, la présence dans le réseau de correspondants n'est pas forcément connue du serveur. Par contre, la liste des contacts récupérés permet d'interroger chacun des serveurs en charge des profils de ces contacts puisque l'identifiant de chacun d'eux porte la mention du serveur en charge du compte. Le serveur peut donc contacter ces serveurs et, en retour, informer l'utilisateur connecté du statut de ses contacts.

La figure 11.5 illustre la manière dont les serveurs Jabber peuvent coopérer pour localiser les utilisateurs du réseau Jabber et offrir un service distribué et cohérent.

Performante, souple, puissante et évolutive, la plate-forme Jabber séduit de plus en plus les internautes, avec son modèle libre et ouvert qui s'oppose radicalement aux politiques des majors de l'industrie. Reste qu'au niveau des services proposés, Jabber souffre d'un certain retard. L'implémentation de la vidéo fait défaut, et les fonctionnalités sont encore globalement restreintes.

Il n'en reste pas moins que la définition de normes et l'appui d'importants acteurs jouent en sa faveur. Pour les communications de ses abonnés, Meetic, le numéro un des sites de rencontres, utilise ainsi les protocoles Jabber. D'autres acteurs importants, comme la marque Orange de France Télécom, les utilisent dans les Messenger destinés à leurs abonnés. Quant aux géants IBM et Sun, ils proposent également des logiciels exploitant Jabber.

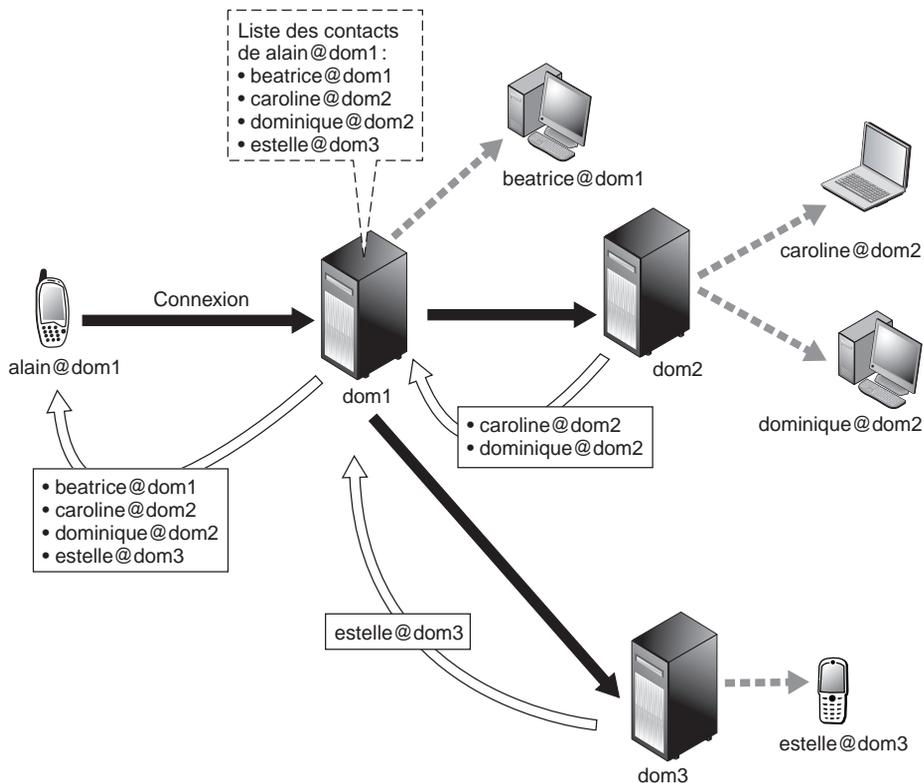


Figure 11.5

*Le modèle coopératif des serveurs Jabber*

Plus encore, le nouveau service de messagerie instantanée Google Talk du célèbre moteur de recherche Google implémente les protocoles Jabber et dispose même d'une passerelle de communication pour se connecter au réseau Jabber. Les utilisateurs de Google Talk peuvent contacter les utilisateurs Jabber et *vice versa*. Google va encore plus loin en contribuant au développement des technologies Jabber, en particulier dans le domaine du traitement de la voix sur IP.

## Google Talk

Créé en 1998 par deux chercheurs en informatique, Sergey Brin et Larry Page, Google n'en finit pas de tisser sa toile sur le réseau. La société, dont le siège est à Mountain View, en Californie, se fixe comme politique de réaliser elle-même la plupart de ses développements et d'être présente dans tous les domaines technologiques stratégiques.

Peu de rachats pour l'acquisition de sociétés, donc, mais des développements tous azimuts. Plus incroyable encore, pour un acteur dit *pure player* (c'est-à-dire dont

l'activité est visible uniquement sur Internet, et dont les bénéfices se font majoritairement sur les revenus publicitaires), Google ne passe aucune publicité sur Google Talk. Autant dire que la société est attendue au tournant et que les erreurs de parcours peuvent lui coûter cher.

L'un des derniers fers de lance de l'insatiable société Google, comparée à une pieuvre pour la diversité de ses activités, concerne la téléphonie et ses dérivés. À nouveau, la téléphonie se couple à l'inévitable messagerie instantanée, mais pas seulement.

### **Une offre à trois volets**

Pour Google, le service de téléphonie Google Talk n'est que le troisième volet de son offre de communication pour les internautes, qui inclut aussi Google Mail, le courrier électronique, et Google Chat, la messagerie instantanée.

#### **Google Mail**

Google Mail (GMail), le webmail de Google, propose une interface sobre mais hautement fonctionnelle et particulièrement rapide. Avec GMail, la société démontre, s'il en était besoin, qu'elle est un acteur avec qui compter, même dans un domaine largement occupé par les champions Microsoft et Yahoo!

Pour preuve, la version Bêta de la messagerie est restée, pendant presque trois ans, accessible uniquement à un panel d'utilisateurs triés sur le volet, qui pouvaient inviter d'autres internautes à se créer un compte. Plusieurs sites ont cependant été créés pour faire don (ou commerce) de ces précieuses invitations, qui revêtaient un parfum d'exclusivité et de privilège. La transmission virale s'est ensuite répandue à très grande vitesse.

L'innovation du service est réelle. Le webmail de Google compte parmi les plus rapides et puissants du marché. Reposant sur une interface sobre, il est programmé en Ajax, le nouveau langage à base de contrôles JavaScript qui offre une très large liberté d'action et de personnalisation.

Gratuit et doté d'un espace de stockage de plusieurs gigaoctets, Google Mail fourmille d'innovations. Ses capacités de recherche et d'organisation de courriers électroniques sont remarquables d'efficacité et de précision. Il se complète de manière naturelle par un simple lien avec l'outil de gestion d'agenda Google Calendar.

#### **Google Chat**

Toujours en version limitée, la messagerie instantanée Google Chat est discrètement passée à la vitesse supérieure en proposant un canal de communication par messagerie textuelle. Accessible à l'aide d'un navigateur Internet, le service est totalement intégré à l'interface de Google Mail.

Le carnet d'adresses de la messagerie électronique est communément partagé par la liste de contacts de la messagerie instantanée. Par un simple clic sur l'interface du webmail,

les utilisateurs peuvent connaître la disponibilité de leur contact et communiquer entre eux, en temps réel, par des messages textuels.

## Google Talk

Troisième volet de la communication initiée par Google, la ToIP a été lancée le 9 août 2005. Dans Google Talk la conversation est fluide, stable, sans coupure ni retard perceptible et globalement d'excellente qualité. Le tout est par ailleurs peu gourmand en ressource.

Comme Google Chat, Google Talk s'appuie sur le standard Jabber/XMPP et hérite de tous ses avantages, à commencer par un réseau décentralisé et une grande flexibilité du client de messagerie. Il est ainsi possible d'exploiter les serveurs de Google pour ses communications en utilisant un client différent de Google Talk.

Rappelons que le protocole Jabber laisse tout loisir au développement d'extensions annexes puisqu'il est ouvert.

## Utilisation

Disponible sous toutes les plates-formes Windows récentes, le client Google Talk se télécharge sur le site de Google et pèse à peine quelques mégaoctets.

Après une installation sommaire et l'exécution du logiciel, la conventionnelle fenêtre d'authentification s'affiche. L'utilisateur est invité à y saisir son identifiant Google Mail (avec ou sans le nom de domaine *gmail.com*) et son mot de passe associé pour accéder à l'interface principale.

Comme l'illustre la figure 11.6, celle-ci se distingue par sa sobriété. Très peu colorée, elle est réduite au strict nécessaire et ne propose que peu d'options. L'accent est mis sur l'aspect fonctionnel du logiciel. Sa simplicité sollicite faiblement les ressources système. Il n'en dispose pas moins d'une vaste gamme d'outils permettant sa personnalisation, avec notamment la possibilité de choisir son avatar ou l'apparence des fenêtres de discussion.

Figure 11.6

Fenêtre principale de Google Talk



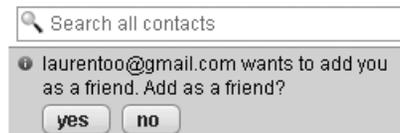
### Ajouter des contacts

Les utilisateurs du webmail de Google apprécieront de ne pas avoir à entrer dans Google Talk la liste de leurs contacts, car celle-ci est automatiquement importée de Google Mail vers Google Talk. La liste est directement disponible dans la partie centrale du logiciel.

Contrairement à d'autres logiciels, le nombre de contacts est illimité. Ces contacts ne sont pas encore joignables pour autant, et il est nécessaire d'envoyer une invitation à chacun d'eux avant de pouvoir être mis en relation. Pour cela, un double-clic sur le nom du contact suffit. Le contact reçoit alors une invitation lui demandant s'il accepte d'afficher son statut à son correspondant et s'il souhaite lui aussi ajouter ce dernier à la liste de contacts (voir figure 11.7).

Figure 11.7

*Ajouter un contact*

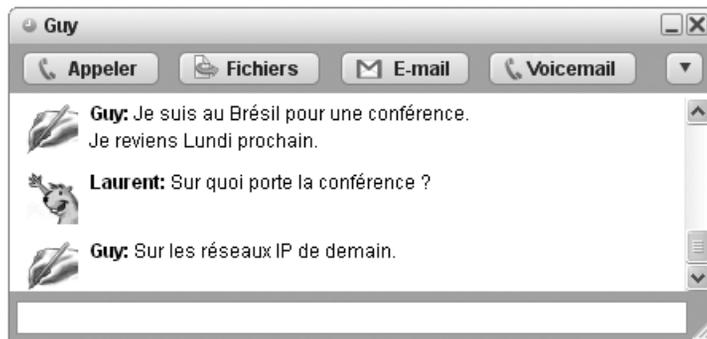


### Lancer une communication

En double-cliquant sur le nom d'un contact de sa liste, une fenêtre de conversation est lancée. Comme l'illustre la figure 11.8, celle-ci est relativement rudimentaire, en tout cas très loin des sophistications disponibles sur Skype, WLM ou Yahoo.

Figure 11.8

*Fenêtre de communication de la messagerie instantanée*



Comme on le voit, l'interface est épurée et ne comporte que les quatre boutons de contrôle suivants :

- Appeler. Permet de lancer une communication téléphonique avec le correspondant dont la fenêtre est ouverte. Seules les communications en IP sont possibles. Le service actuel utilise Jingle (XEP-0166), via l'implémentation protocolaire libjingle que Google a réalisée et rendue publique. L'intégration avec le protocole SIP est prévue dans une prochaine version.

- Fichiers. Permet d'envoyer un document, quel que soit son format (texte, audio, vidéo ou autre), à un correspondant. L'originalité de cette fonction repose sur la possibilité de visualiser un aperçu des images transmises.
- E-mail. Permet d'ouvrir son navigateur par défaut sur la page de Google Mail et d'envoyer un e-mail à son correspondant.
- Voicemail. Permet de laisser un message audio sur la messagerie de son correspondant (que ce dernier soit en ligne ou non). Les messages laissés sont ensuite accessibles sur l'interface principale grâce à une icône disposée au bas de l'interface, à côté du nombre d'e-mails, et indiquant le nombre de messages stockés sur son répondeur (*voir figure 11.6*).

Les messages vocaux sont également disponibles directement dans l'interface Web de son compte Google Mail. Il est possible de télécharger ces messages sous forme de fichiers audio au format MP3 ou de les écouter à partir du navigateur (si ce dernier dispose du plug-in Flash).

Ces fonctionnalités sont relativement peu nombreuses en comparaison de celles des concurrents. Une conversation avec la messagerie instantanée ne peut faire intervenir que deux personnes au maximum. La téléphonie se restreint au monde IP, et la vidéo n'est pas disponible, même entre utilisateurs du réseau IP. En dépit de ces limitations, le programme GMail progresse, tout en fournissant un service de qualité, que ce soit en termes de disponibilité, de fiabilité ou de performance.

### Intégration avec le service Google Mail

Le service de Google est conçu pour unifier les communications, quels que soient les canaux utilisés. Si un utilisateur n'a pas installé le client Google Talk, le simple fait qu'il se connecte sur sa messagerie webmail le rend néanmoins potentiellement disponible pour n'importe quel utilisateur de sa liste de contacts.

En fait, le service GMail agit pour l'utilisateur comme un client Jabber, au même titre que le logiciel Google Talk, à cette différence près que ses fonctionnalités sont dégradées par rapport au logiciel Google Talk, lequel s'appuie sur une installation et l'accès à des ressources système dont le navigateur ne dispose pas. Ces fonctions sont toutefois accessibles avec un simple navigateur, quel que soit le système d'exploitation utilisé.

En plus de disposer de son service de messagerie standard, l'utilisateur est automatiquement enregistré comme un client Jabber lorsqu'il accède à GMail. L'affichage de l'ensemble de ses correspondants se fait par le biais de la section Contacts. Cette dernière liste exhaustivement l'ensemble des contacts et permet de configurer, dans la section « Contacts rapides » illustrée à la figure 11.9, les contacts qui seront disponibles en permanence sur l'interface.

En regard du nom de chaque contact, la disponibilité et l'état sont indiqués. L'icône qui précède le nom de l'utilisateur peut varier selon l'état. Par exemple, si un correspondant est en train de converser par messagerie instantanée, son icône symbolise une info-bulle pour indiquer qu'une communication est en cours.

Figure 11.9

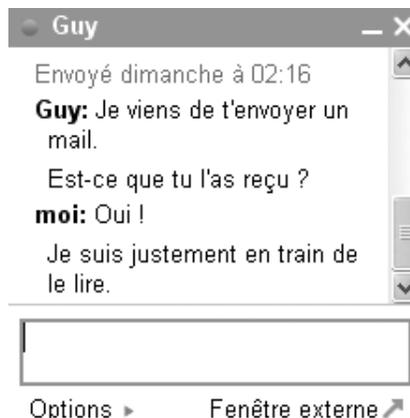
État des contacts et actions disponibles



Pour sélectionner un contact, il suffit de pointer la souris sur l'identifiant correspondant. Une fenêtre informative s'affiche alors et présente deux boutons d'action. Le premier permet d'envoyer un courrier électronique, et le second d'initier un dialogue par messagerie instantanée avec le contact. Si ce dernier n'est pas connecté, le bouton est grisé, et donc inaccessible. S'il est connecté (que ce soit au travers du logiciel Google Talk, d'un client alternatif ou de son webmail), le bouton est disponible, et un simple clic lance la fenêtre de dialogue illustrée à la figure 11.10.

Figure 11.10

Fenêtre de la messagerie instantanée avec le webmail



Cette fenêtre de dialogue est persistante. Elle reste affichée même si l'on se déplace dans les différentes sections de son webmail, afin d'y lire ou écrire ses e-mails. Cette même interface Web permet d'inviter parallèlement plusieurs contacts à discuter. Seuls les messages textuels sont possibles *via* le webmail, à l'exclusion donc des fonctions de téléphonie.

Il est possible pour un utilisateur de sélectionner un statut parmi tous ceux qu'il a personnalisés dans Google Talk (ces statuts sont sauvegardés au niveau du serveur, et non pas en local sur le poste de l'utilisateur) ou bien d'en saisir un nouveau.

Réciproquement, tous les correspondants peuvent initier une conversation avec un utilisateur de GMail. Dans ce cas, une alerte est remontée à ce dernier, lui précisant qui souhaite le contacter et lui demandant s'il souhaite accepter la conversation (*voir figure 11.11*).

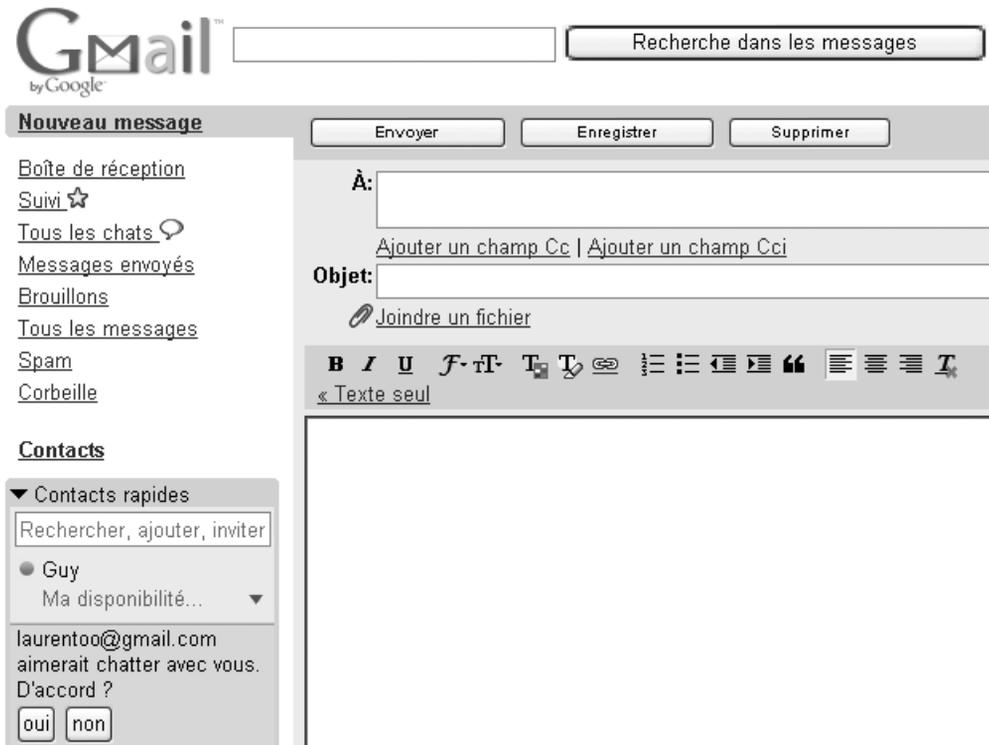


Figure 11.11

*Invitation à partir de GMail*

### Gestion des connexions multiples

Lorsqu'un utilisateur est connecté simultanément de différentes manières, il reçoit son premier message sur l'ensemble des supports auxquels il est connecté. Dès qu'il répond par le biais de l'un de ces supports, la conversation se poursuit uniquement sur ce canal.

Par exemple, si l'utilisateur Guy est connecté par le client Google Talk en même temps qu'il consulte ses e-mails sur Google Mail, un message instantané de l'utilisateur Laurent lui parvient simultanément sur Google Talk et sur son webmail. Si Guy répond ensuite à partir de son webmail, les prochains messages de Laurent lui parviennent sur son webmail uniquement.

Si, pendant la conversation, Guy décide de passer sur son client Google Talk, sa conversation bascule sur ce nouveau support, et les prochains messages de Laurent sont affichés sur le client Google Talk uniquement.

Google fournit ainsi de multiples manières de se connecter à son réseau de messagerie. Ne serait-ce qu'en consultant leur messagerie Web, les utilisateurs restent joignables, à leur travail comme à leur domicile, sans jamais avoir besoin de se déconnecter.

### Google Talk avec un logiciel alternatif

Comme le protocole Jabber l'autorise, un même utilisateur peut être connecté plusieurs fois simultanément. On en a vu une illustration avec un utilisateur connecté au réseau Jabber à la fois par Google Talk et Google Mail. Celui-ci pourrait tout aussi bien utiliser un autre client que celui proposé par défaut.

La société Google encourage du reste les programmeurs à créer de nouvelles applications clientes. La seule condition est que ce client respecte les spécifications du protocole ouvert Jabber.

Dans un tel cas, il suffit alors de configurer le logiciel avec les paramètres récapitulés au tableau 11.3.

**Tableau 11.3 – Configuration d'un logiciel alternatif à Google Talk**

Paramètre	Valeur
Serveur d'authentification	<i>talk.google.com (sur le port: 5222)</i>
Nom d'utilisateur	<i>login_d'utilisateur_gmail</i>
Mot de passe	<i>mot_de_passe</i>

Les codecs suivants sont pris en charge pour les communications audio G.711 (PCM loi A et PCM loi mu), G.723, iLBC et Speex. Cette liste est restreinte, mais satisfaisante pour la majorité des utilisateurs.

L'authentification aux serveurs de messagerie de Google doit reposer sur le protocole SASL (Simple Authentication and Security Layer). Le transport des flux s'effectue de manière chiffrée par le protocole TLS (Transport Layer Security).

L'avantage des clients alternatifs est que, en plus de proposer des services complémentaires à ceux fournis par Google Talk, ils sont utilisables dans la majorité des plates-formes, alors que Google Talk n'est disponible que sous Windows.

## Conclusion

L'unification des réseaux de messagerie et de téléphonie IP est un enjeu important, vivement souhaitée par les utilisateurs pour élargir leur cercle de contacts. Redoutée par les majors qui dominent le marché et risqueraient de perdre les utilisateurs habitués, cette unification pourra se faire selon deux axes : soit en implémentant les protocoles de chaque logiciel (bien souvent par des partenariats entre les acteurs), soit sur la base d'un protocole utilisé par tous.

Pour se faire connaître du grand public, la plate-forme Jabber n'étant pas lucrative, n'a certes pas la puissance commerciale des logiciels concurrents. De plus, elle ne centre pas non plus ses développements autour de la téléphonie, mais vise principalement la messagerie instantanée. La téléphonie n'est que l'indispensable extension de la plate-forme. Toutefois, Jabber a le mérite de proposer une approche libre, ouverte et normalisée, ce qui permet de poser de solides bases aux nouveaux acteurs et aux développements des particuliers, tout en ouvrant la voie à une unification des réseaux de messagerie et de téléphonie.

Mail, messagerie instantanée et téléphonie : l'intégration des trois facettes de l'offre Google dans une interface simple et fonctionnelle semble naturelle, même si le logiciel reste encore très limité en termes de fonctionnalités.

Si l'offre s'enrichit progressivement, elle reste loin derrière celles des leaders Microsoft et Yahoo!. Le service se cantonne à la téléphonie purement IP, et il n'est pas encore possible d'appeler ni d'être appelé avec des téléphones standards. La personnalisation du logiciel reste par ailleurs pauvre.

Le logiciel est cependant encore jeune et ne cesse d'évoluer. S'il présente des limitations certaines, il n'en reste pas moins un modèle de stabilité et de simplicité d'utilisation. Le protocole utilisé a la capacité de séduire des utilisateurs des autres réseaux de messagerie, qui peuvent conserver leur liste de contacts en passant à la technologie Jabber.

Lorsque les habitudes seront prises chez les utilisateurs, la montée en service sera inévitable tant la société dispose d'une force de frappe considérable. De fait, la véritable question n'est peut-être pas de savoir quelle société détient le plus de clients ou possède le logiciel le plus performant, mais plutôt quel acteur rassemblera tous les autres. De ce point de vue, Google est indéniablement sur les rangs.



# 12

## Asterisk

---

Les grandes entreprises sont dotées de centraux téléphoniques, appelés autocommutateurs, PABX (Private Automatic Branch eXchange) ou plus simplement PBX.

Un PBX est une entité logique, presque toujours gérée par un équipement matériel physique dont la fonction est au moins triple : router les appels au sein d'un réseau privé, interconnecter les réseaux et gérer les services de téléphonie.

Ce chapitre se penche sur un PBX d'un genre nouveau, le logiciel Asterisk, qui remplit les mêmes fonctions qu'un PBX professionnel de haut niveau. Aucun équipement spécifique n'est nécessaire, et il suffit d'installer le logiciel sur un ordinateur, librement et gratuitement. Ce type de logiciel est appelé IPBX, ou PBX-IP.

Grâce à son architecture modulaire, à sa facilité de mise en œuvre rapide et à son fonctionnement simplifié, Asterisk peut même être installé par des particuliers, qui peuvent ainsi exploiter les ressources gigantesques dont il dispose.

### Introduction aux PBX

Seul élément du réseau à connaître la localisation de chaque terminal téléphonique, le PBX a pour fonction principale le routage. Les terminaux sont des entités élémentaires, ce qui réduit leur coût unitaire et permet leur gestion centralisée.

Lorsqu'on ajoute un terminal téléphonique au sein d'une entreprise, il n'est pas nécessaire de modifier les autres terminaux pour les en informer. C'est le PBX qui centralise l'intelligence du réseau et effectue les tâches de connectivité, de mise en relation des interlocuteurs et de gestion des communications locales au réseau. Il assure en outre la liaison avec le réseau téléphonique commuté global. Autrement dit, le PBX fait office de passerelle téléphonique pour les communications locales (d'un point de vue logique et

non physique), mais aussi pour celles entre les utilisateurs du réseau local et les utilisateurs reliés au réseau téléphonique traditionnel.

De cette façon, les communications locales ne sont plus facturées par l'opérateur de téléphonie, puisqu'elles n'arrivent pas jusqu'à lui. Gérées en interne par le PBX, elles deviennent gratuites. De plus, les services mis en place sur le PBX (annuaire, messagerie, etc.) sont indépendants de l'opérateur téléphonique.

Un autre avantage des PBX est que l'entreprise n'a plus besoin d'avoir autant de lignes téléphoniques extérieures qu'elle dispose de lignes en interne. Dans la mesure où il est peu probable que tous les téléphones d'une entreprise soient utilisés en même temps, il est possible de limiter le nombre de lignes extérieures à ouvrir en majorant statistiquement l'usage qui est fait des téléphones. Ainsi, toutes les lignes internes peuvent communiquer, à condition qu'elles ne le fassent pas toutes en même temps. C'est ce qu'illustre la figure 12.1.

Dans la pratique, les utilisateurs ne perçoivent pas cette limitation, à de rares exceptions près, comme lors d'incendies, au cours desquels certains appels peuvent ne pas aboutir en raison de l'affluence des communications. C'est la raison pour laquelle on recommande de réduire au minimum les appels dans ces circonstances particulières.

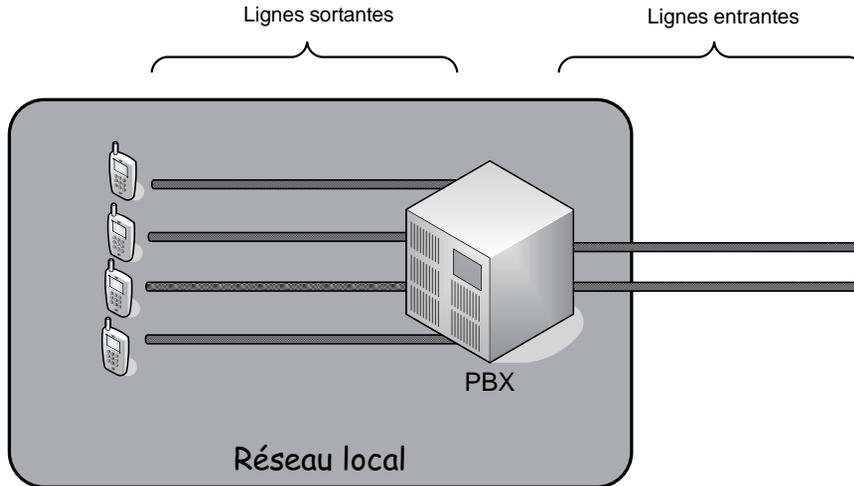


Figure 12.1

*PBX pour la gestion des appels*

Les PBX peuvent servir à héberger différents services téléphoniques, comme un serveur vocal, un répondeur personnalisé, la redirection d'appels ou encore la tenue de journaux d'appels. Nous donnons un peu plus loin une liste de quelques services parmi les plus connus.

Les PBX assument enfin la fonction d'interconnexion de réseaux de nature différente. En effet, il ne suffit pas qu'une entreprise souhaitant passer à la ToIP s'équipe de terminaux téléphoniques IP. Pour permettre aux utilisateurs de joindre le réseau téléphonique RTC, une passerelle doit être hébergée au sein du PBX. Enjeu majeur, c'est l'interconnexion entre réseaux qui permet de fédérer les réseaux et d'offrir une plate-forme de services transparente pour l'utilisateur, lui permettant d'accéder aux mêmes services, quel que soit le réseau qu'il utilise.

## Présentation d'Asterisk

Dans la plupart des langages informatiques, l'astérisque (dont le symbole est \*) est utilisé comme caractère générique (en anglais *wildcard*), pour remplacer n'importe quel autre caractère ou ensemble de caractères. Il s'agit en quelque sorte d'un joker, la carte ou valeur qui remplace toutes les autres. C'est de ce concept de généricité, évoquant à la fois la souplesse, l'adaptabilité et la puissance, que tire son nom le logiciel Asterisk.

Asterisk est un PBX-IP, ou IP PBX ou encore IPBX. Complet et performant, il offre une plate-forme personnalisable et modulable pour la mise en œuvre de services de téléphonie. Il garantit une très large interconnexion avec plusieurs serveurs PBX, mais aussi avec des réseaux de téléphonie non-IP.

Développé en 2001 par Mark Spencer, de la société américaine Digium, il continue d'être fortement soutenu par cette dernière. En France, c'est la société Eikonex qui assure le catalogue commercial de Digium et propose des formations sur le logiciel, ainsi que le développement d'applications pour des centres d'appels. Eikonex commercialise aussi des cartes servant d'interfaces entre différents réseaux et également des ordinateurs complets, préinstallés et équipés de l'ensemble des équipements optionnels dont on souhaite disposer.

Asterisk étant un logiciel libre d'utilisation, ses sources sont téléchargeables sous licence GNU GPL (General Public License). Cela permet à une importante communauté de contribuer à son développement. Des forums libres et actifs enrichissent, testent, mettent à jour et améliorent en permanence le logiciel. Bien qu'initialement conçu pour fonctionner sous Linux, il est aujourd'hui multiplate-forme et s'installe aussi bien sur OpenBSD que FreeBSD, Sun Solaris, MacOS X ou Windows.

L'enjeu d'une offre telle qu'Asterisk est colossal. Pour peu que l'on dispose des connaissances requises, il devient possible de remplacer une lourde et très onéreuse mise en œuvre d'un équipement PBX par un simple ordinateur équipé du logiciel gratuit, éventuellement muni de cartes d'interfaces pour l'interconnexion avec différents types de réseaux non-IP. Le logiciel se pose en rival viable et robuste dans un marché dominé par les géants Alcatel, Nortel, Cisco, 3Com, Avaya ou Siemens, pour ne citer que quelques-uns des équipementiers les plus connus.

Alléchées par une économie d'environ 50 % par rapport aux solutions hardware classiques, les entreprises multiplient l'adoption de ce type de solution.

## Fonctionnalités

Asterisk propose toutes les fonctionnalités d'un standard téléphonique de niveau professionnel, des plus élémentaires aux plus complexes. Non seulement, il permet de gérer le routage des appels au sein du réseau, mais en plus il supporte une large gamme de services, notamment les suivants (pour la liste exhaustive, voir le site de l'éditeur, à l'adresse <http://www.asterisk.org>) :

- Authentification des utilisateurs appelants.
- Serveur vocal, ou standard d'accueil téléphonique automatisé, aussi appelé IVR (Interactive Voice Response). Cette fonction permet de demander à l'appelant le service qu'il souhaite utiliser et d'effectuer le routage correspondant.
- Numérotation abrégée pour définir des raccourcis.
- Transfert d'appel.
- Filtrage des appels.
- Messagerie vocale (répondeur automatique).
- Notification et écoute par e-mail des messages laissés sur son répondeur (*voicemail*).
- Gestion des conférences.
- Double appel.
- Mise en attente.
- Journalisation des appels.
- Facturation détaillée.
- Enregistrement des appels.

Le logiciel peut être utilisé comme une passerelle ToIP hétérogène. Par exemple, des utilisateurs utilisant différents protocoles de signalisation, comme H.323 ou SIP, peuvent être mis en relation. C'est le logiciel qui se charge d'effectuer les conversions de signalisation. De la même manière, il peut servir de passerelle pour joindre des correspondants dans le réseau téléphonique RTC. Enfin, le logiciel est modulable et extensible au moyen de scripts et de modules implémentés en langage C ou Perl.

## Compatibilité

Les supports protocolaires d'Asterisk sont très larges. La signalisation sur IP est pleinement supportée avec les protocoles standardisés les plus courants, notamment H.323, SIP et MGCP, mais aussi avec les protocoles IAX (Inter Asterisk eXchange), conçu dans le cadre du projet Asterisk, et SCCP (Cisco Skinny), conçu par Cisco. L'interopérabilité est également assurée vers la téléphonie standard RTC (support pour E&M, E&M Wink, FXS, FXO, GR-303, Loopstart, Groundstart, Kewlstart, MF and DTMF, Robbed-bit Signaling (RBS) et MFC-R2), ainsi que vers la téléphonie RNIS (support pour 4ESS, BRI (ISDN4Linux), DMS100, EuroISDN, Lucent 5E, National ISDN2 et NFAS).

Par ailleurs, le logiciel supporte notamment les codecs audio suivants : G.711 (compatible avec la loi A et la loi ?), ADPCM, G.723.1, G.726, G.729 (soumis à une licence de la

société Digium), GSM, iLBC, Linear, LPC-10 et Speex. Au niveau vidéo, le support des codecs de référence H.263 et H.263+ est aussi fourni. Ce support des codecs les plus réputés offre une large palette de possibilités et couvre l'essentiel des besoins.

## Cible et usage

La première vocation d'Asterisk est de remplacer les PBX d'entreprise, très coûteux, et dont les configurations diffèrent d'un équipement à l'autre. L'objectif est de proposer un logiciel capable de rivaliser avec ces équipements professionnels, à commencer par le support des fonctionnalités de localisation et de mise en relation des utilisateurs.

S'il est naturel de concevoir la présence d'un central téléphonique dans un cadre professionnel, l'exploitation d'une telle puissance fonctionnelle peut-elle s'avérer judicieuse dans un cadre domestique ? Dans un cadre privé, qui souhaiterait disposer d'un mécanisme de standard automatique ? Qui souhaiterait demander à l'appelant de saisir une touche correspondant au membre de la famille qu'il cherche à joindre ?

Un PBX paraît *a priori* trop puissant pour la maison, voire incongru. En réalité, un PBX est bien plus que cela, et certaines de ses fonctionnalités étonnantes et souvent méconnues sont parfaitement adaptées aux besoins des particuliers.

## Réduire les coûts en appelant de l'extérieur au tarif domestique

De nombreux FAI proposent désormais la téléphonie gratuite illimitée vers l'étranger. L'abonné, s'il est chez lui, n'a donc pas à payer les appels longue distance et longue durée vers les destinations proposées. En revanche, il est facturé très cher pour les mêmes appels dès s'il se trouve en dehors de chez lui, par exemple s'il appelle à partir de son téléphone portable. Une solution bon marché à ce problème consiste à utiliser Asterisk comme relais.

Le principe du relais est simple. L'abonné dispose d'un serveur Asterisk correctement configuré chez lui. Lorsqu'il est en déplacement, il appelle le serveur Asterisk. Celui-ci lance automatiquement un processus d'authentification (par exemple en demandant à l'appelant de saisir un code d'accès ou en n'acceptant les appels que si le numéro de téléphone de l'appelant est visible et connu du système). Au terme de l'authentification, le serveur demande à l'appelant de saisir le numéro d'appel de la personne qu'il veut joindre et le compose automatiquement. Il agit dès lors comme un relais pour mettre en relation l'appelant et la personne que ce dernier souhaite joindre.

Dans les conditions que l'on a mentionnées, l'appel composé par le serveur Asterisk vers la destination souhaitée est gratuit. En effet, le serveur étant situé chez l'appelant et relié à la ligne du fournisseur d'accès, il bénéficie des tarifs proposés par ce dernier. Au total, seul l'appel vers le serveur Asterisk est facturé.

Les étapes que nous avons mentionnées ne sont pas si contraignantes qu'elles peuvent paraître à première vue. La même démarche est exploitée par les opérateurs de téléphonie dits alternatifs, qui vendent des cartes prépayées « à gratter » : en téléphonant au numéro indiqué, on s'identifie avec le code gratté avant d'appeler son correspondant.

La seule réelle contrainte est que le serveur Asterisk doit gérer deux appels en parallèle : celui qu'il reçoit de l'appelant et celui qu'il émet pour le compte de ce dernier vers le destinataire. Cela implique de disposer de deux lignes téléphoniques distinctes. Nous verrons que plusieurs opérateurs proposent à moindre coût des lignes téléphoniques IP associées à un même numéro de téléphone.

Pour aller plus loin dans cet exemple, on peut optimiser encore le fonctionnement décrit précédemment grâce à la procédure dite de call-back. Dans ce scénario, l'appel initial est inversé, c'est-à-dire remplacé par un appel initié par le serveur Asterisk :

1. L'appelant appelle son serveur.
2. Contrairement au cas précédent, ce dernier ne répond pas automatiquement à l'appel, mais repère le numéro de l'appelant et attend que ce dernier raccroche. Si ce numéro est reconnu comme étant habilité à utiliser le service de call-back, le serveur Asterisk lance la procédure de call-back en initiant un appel vers l'appelant.
3. Une fois en contact avec l'appelant, le serveur l'invite à saisir ses commandes et à indiquer le numéro de la personne à contacter.
4. Le serveur Asterisk appelle le correspondant.

Là encore, deux lignes téléphoniques sont nécessaires pour faire fonctionner le service. Si l'on suppose que ces lignes sont forfaitaires, un coût mensuel, parfois inclus dans l'abonnement Internet, permettant d'appeler vers plusieurs destinations, l'appel est gratuit. Le call-back permet ainsi de bénéficier de tarifs généralement avantageux.

La figure 12.2 illustre la procédure de call-back.

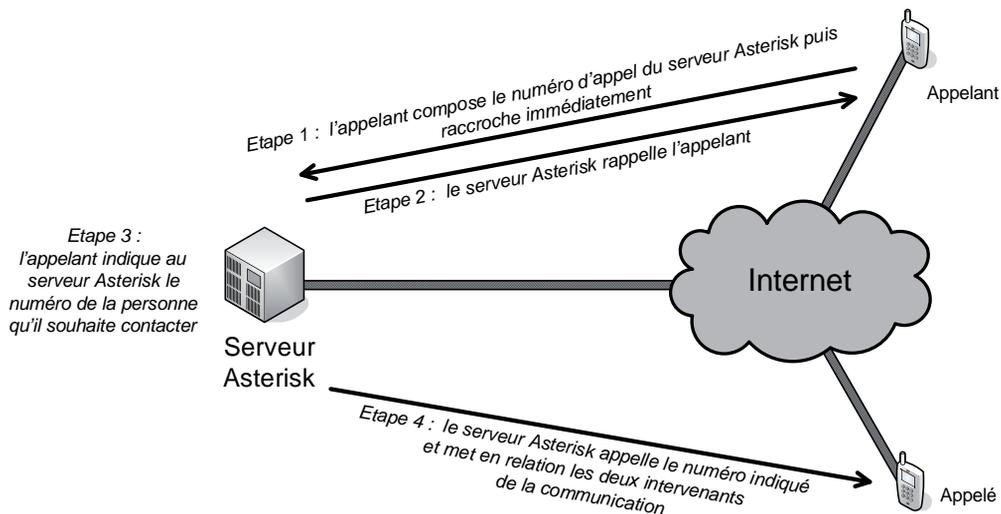


Figure 12.2

Procédure de call-back

### Assurer le nomadisme des utilisateurs

Plusieurs services permettent de disposer d'un numéro de téléphone IP. Par exemple, l'option SkypeIn de Skype fournit un numéro géographique aux utilisateurs. Étant au format international, ce numéro est utilisable depuis n'importe quelle ligne téléphonique. L'utilisateur est de la sorte joignable quel que soit l'endroit où il se trouve, pour peu qu'il dispose d'une connexion à Internet.

Avec Asterisk, si un utilisateur voyage à l'étranger, il lui suffit de configurer son serveur (relié à sa ligne téléphonique habituelle) pour demander une redirection des appels entrants vers un poste IP spécifique. Dès qu'un appelant souhaite joindre cet utilisateur, il lui suffit de composer son numéro de téléphone pour que l'appel aboutisse au serveur Asterisk, lequel n'a plus qu'à activer la règle lui permettant de relayer l'appel vers le poste IP.

Le même service de nomadisme est disponible avec un téléphone portable en activant l'option d'appels internationaux, si ce n'est que c'est l'appelé qui est généralement facturé pour la redirection d'appel de son pays d'origine vers le pays où il se trouve.

### Améliorer les services téléphoniques

Avec le serveur Asterisk, tous les services téléphoniques usuels sont disponibles. Si certains d'entre eux sont plutôt réservés à des usages professionnels, d'autres sont destinés au grand public. Il est notamment possible, et gratuit, de disposer d'une numérotation abrégée, d'interroger sa messagerie à distance, de recevoir des messages téléphoniques sur sa messagerie électronique en format audio, d'avoir le détail complet des communications passées, ou encore, sous réserve de disposer de l'accord des participants, d'enregistrer des conversations, de lancer des conférences audio, etc.

## Installation de base

En exploitant un ensemble de fichiers restreint, le PBX Asterisk offre une gamme de possibilités de configuration extrêmement large, répondant aux besoins les plus divers et permettant une personnalisation très évoluée.

Au premier abord, cette richesse d'exploitation et ce haut niveau de personnalisation et d'optimisation peuvent sembler rebutants pour les débutants. Plus encore, si la documentation d'Asterisk est globalement assez riche, elle est surtout disponible en langue anglaise. Cela vaut aussi pour les nombreux forums de discussion et d'aide dédiés.

L'ambition de cette section est de fournir une approche suffisamment simple pour initier le lecteur aux concepts fondamentaux d'Asterisk et lui permettre de disposer en un temps record d'une première version élémentaire, mais fonctionnelle, du serveur.

Sans prétendre à l'exhaustivité, puisqu'un livre entier serait nécessaire pour exposer en détail toutes les possibilités et options disponibles, nous nous contenterons d'une approche synthétique de ces concepts nourrie de nombreux exemples.

Nous commencerons par installer et exécuter le serveur avec une configuration par défaut. À ce stade, le serveur sera fonctionnel sans être exploitable. À partir de cet environnement, stable et propre, et auquel il sera possible de revenir ultérieurement en cas de problème, nous passerons à l'étape de configuration des fichiers, avant de nous pencher sur la mise en place de quelques services de base. Nous évoquerons ensuite des techniques d'optimisation de la configuration. Nous terminerons enfin par la présentation de fonctionnalités plus évoluées et fournirons des pistes pour poursuivre l'étude du logiciel de différentes manières.

### ***Mise en œuvre de la plate-forme***

Au lieu de se lancer dans la configuration du logiciel, nous allons d'abord utiliser une configuration élémentaire préinstallée dont nous détaillerons le fonctionnement avant de le personnaliser selon nos besoins. De cette manière, nous limiterons les risques de mauvaises manipulations de fichiers et les fonctionnements inattendus. Les modifications et l'enrichissement des paramètres seront effectués de manière progressive sur une base fonctionnelle.

L'architecture d'Asterisk est modulaire. Autour du serveur, qui constitue son cœur et la seule brique indispensable, viennent se greffer plusieurs composants additionnels qui enrichissent ses fonctionnalités. Il est donc possible de n'installer que les éléments dont nous avons besoin, en choisissant certains modules plutôt que d'autres et en laissant la possibilité d'en ajouter ensuite, voire d'en concevoir soi-même.

Rien n'étant imposé, nous pouvons déterminer presque à la carte les compléments à apporter. Par exemple, nous choisirons d'installer ou non un logiciel de synthèse vocale en français, des pilotes supplémentaires pour la gestion de composants hardware, un logiciel de facturation, etc.

Les sections qui suivent présentent quelques composants classiques, dont nous détaillerons les implémentations et exemples sous une plate-forme Linux.

#### **Se connecter au réseau téléphonique traditionnel**

En ToIP pure, le logiciel Asterisk est d'emblée fonctionnel, si bien qu'aucun composant supplémentaire n'est nécessaire. Cependant, le réseau téléphonique traditionnel utilisant une commutation par circuits, et non par paquets comme dans la téléphonie IP, il est nécessaire de s'équiper d'une interface permettant d'effectuer la conversion des signaux d'un flux IP en un flux RTC et réciproquement.

Ces interfaces sont disponibles facilement dans le commerce, généralement sous forme de cartes PCI, à installer sur son PC, ou de boîtiers, à connecter en USB.

Ils offrent plusieurs branchements pour disposer de plusieurs lignes téléphoniques en parallèle.

Selon les modèles, il faut en outre installer le pilote de la carte, fourni par le constructeur, permettant au système d'exploitation de la reconnaître comme nouveau périphérique matériel et de la rendre disponible et configurable avec Asterisk.

La société Digium, qui maintient le logiciel Asterisk, assure la vente de ces composants à des tarifs compétitifs. D'autres sociétés proposent des cartes analogues, mais leur compatibilité avec les composants vendus pas Digium n'est pas toujours garantie.

On distingue les deux formats d'interface suivants :

- **FXS (Foreign eXchange Subscriber)**, qui permet le branchement d'un téléphone analogique sur le serveur Asterisk. Si l'on ne souhaite pas investir dans l'achat d'équipements purement IP, il est possible d'utiliser son téléphone analogique habituel en le reliant au serveur Asterisk par le biais d'une carte FXS. La carte assure la fonctionnalité de conversion dans les deux sens d'un signal numérique en un signal analogique.
- **FXO (Foreign eXchange Office)**, qui permet le branchement du serveur Asterisk sur une ligne téléphonique classique. Pour interagir avec le monde RTC et dépasser le cadre du réseau purement IP, cette carte assure la jonction avec la téléphonie RTC. Elle joue le rôle de passerelle en faisant communiquer tout utilisateur connecté à Asterisk avec des utilisateurs connectés au réseau RTC.

Dans le cas où l'on souhaite supporter la connectivité avec le réseau téléphonique RTC, il est nécessaire d'installer les drivers Zaptel et Libpri. Tous deux concernent la gestion des cartes FXO/FXS avec Asterisk pour des composants de type Zaptel ou « Zaptel compliant », c'est-à-dire conformes aux composants de type Zaptel, sans être pour autant standards.

#### Attention !

Il est indispensable d'installer ces modules avant l'installation du serveur Asterisk proprement dit. En effet, ces pilotes étant dépendants de composants matériels, le serveur doit les prendre en compte lors de sa compilation et ne peut les intégrer par la suite. Si le serveur Asterisk a déjà été installé, il est nécessaire, après avoir compilé et installé les pilotes Zaptel et Libpri, de revenir à l'étape de compilation du serveur. Cette dernière peut alors prendre en compte le chargement des pilotes.

### Télécharger les composants utiles

Les composants d'Asterisk se présentent sous forme d'archives portant l'extension **.tar.gz** et non de fichiers binaires, qu'il faut compiler puis installer manuellement.

1. Commençons par télécharger la dernière version disponible du logiciel Asterisk, à l'adresse <http://www.asterisk.org/download> (ou <ftp://ftp.digium.com/pub/>). Le site est en anglais, mais la section de téléchargement (download) est facilement identifiable. Comme le logiciel

est libre et assez répandu, il dispose de multiples miroirs, dont un moteur de recherche fournira rapidement les liens. On y trouve les modules récapitulés au tableau 12.1.

**Tableau 12.1 – Principaux modules du logiciel Asterisk**

Module	Description
Asterisk	Cœur du logiciel, ce programme est le seul véritablement indispensable à son fonctionnement. Il est donc indispensable de le télécharger.
Asterisk-addons	Ce paquetage comporte le code source du logiciel Asterisk, ainsi que plusieurs modules complémentaires qui peuvent se révéler utiles. Il est vivement recommandé de l'installer.
Asterisk-sounds	Ces modules sont fournis sur plusieurs fichiers de paquetage. Ils fournissent une quantité de sons qui peuvent être utilisés dans des messages d'accueil ou pour signaler à l'appelant diverses informations. Ces messages audio sont disponibles en trois langues, anglais, espagnol et français, et sous plusieurs formats de codec, comme G.711, G.722, G.729 et GSM. L'utilisation de ces sons n'étant pas limitative, il est possible, en respectant les formats supportés par Asterisk, d'en ajouter de sa propre composition ou issus d'autres sources. Ces paquetages ne sont pas indispensables, mais seulement pratiques. Ils peuvent être ajoutés ultérieurement.
Libiax	Cette bibliothèque de codes source pour les communications utilisant le protocole IAX n'est pas indispensable. Elle est surtout destinée au développement de clients IAX. Nous reviendrons, plus loin dans ce chapitre, sur le protocole IAX.
Libpri	Cette bibliothèque est utilisée pour assurer l'interface avec différents types de réseaux non-IP.
Zaptel	Ce paquetage contient les pilotes permettant de prendre en charge les cartes d'interface avec les réseaux non-IP. La section qui suit présente les cas où il est indispensable d'installer ce composant.

#### Remarque

Au lieu de télécharger les sources au format archive, il est possible de télécharger un installateur du logiciel. L'installateur a l'avantage d'être totalement automatisé, et il évite de devoir décompresser puis installer manuellement le logiciel. Néanmoins, il est propre à une distribution particulière, ce qui implique de trouver l'installateur adéquat. Ce dernier se présente le plus souvent comme un fichier portant l'extension **.rpm** pour les distributions Redhat et Mandriva ou **.deb** pour Debian. Les habitués pourront préférer cette méthode à celle que nous présentons, qui reste indépendante de la distribution utilisée.

### Décompresser les sources

Une fois les téléchargements terminés, on peut ouvrir un terminal, se placer à l'emplacement où les archives ont été téléchargées puis saisir la commande qui suit pour chacune des archives que l'on a récupérées :

```
tar -xzvf nom_du_composant_à_installer
```

*nom\_du\_composant\_a\_installer* représente le nom de l'archive téléchargée. Chacune des archives a été décompressée dans un répertoire portant le nom du composant. Les sons n'ont pas besoin d'être installés ; il suffit, une fois l'installation effectuée, de les placer dans le répertoire de sons d'Asterisk (par défaut **/var/lib/asterisk/sounds**). Par contre, les autres programmes, doivent être installés.

## Installer les programmes

Les commandes suivantes permettent d'effectuer la compilation et l'installation d'un composant :

```
cd nom_du_repertoire_du_composant_à_installer
make
make install
```

La première ligne permet de se placer à l'intérieur de répertoire qui a été décompressé précédemment, la seconde lance la compilation du programme et la troisième lance son exécution. Les mêmes commandes sont à reproduire pour chacun des composants, en se plaçant chaque fois dans le répertoire adéquat.

### Important

Il faut respecter l'ordre d'installation des composants. Si l'on souhaite les utiliser, les modules Libpri et Zaptel doivent impérativement être installés avant le serveur Asterisk. Le module Asterisk-addons peut être installé à la fin.

## Lancement du serveur et exploitation

L'étape d'installation achevée, il faut la tester en lançant le serveur Asterisk, afin de vérifier son bon fonctionnement.

Asterisk n'impose pas de fonctionnement par défaut. Il ne dispose d'ailleurs pas de fichier de configuration préétabli lui conférant initialement un modèle de fonctionnement. La manière dont les appels téléphoniques sont dirigés n'est donc pas encore spécifiée. À ce stade, le serveur n'est pas exploitable.

Pour l'utilisateur débutant, il est préférable de commencer en partant d'une base de fichiers que l'on adaptera par la suite selon ses besoins. Nous allons demander à Asterisk de nous préparer une configuration initiale. Pour cela, nous utilisons un terminal, et nous nous plaçons dans le répertoire source d'Asterisk (celui dans lequel nous avons lancé l'installation), pour saisir la commande suivante :

```
make samples
```

À présent, des fichiers standards sont générés et exploitables.

### Remarque

Cette commande est utilisable à tout moment pour revenir à un état de base dont la configuration est valide, sans avoir à lancer une nouvelle installation du serveur. C'est une configuration de référence.

Dans cette section, nous n'allons pas modifier les fichiers de configuration, mais simplement lancer le serveur Asterisk. À la section suivante, nous montrerons comment personnaliser les fichiers de configuration afin de gérer des services.

Il existe deux modes différents de lancement d'Asterisk, le mode serveur et le mode client :

- **Mode serveur.** C'est le mode de fonctionnement principal, dans lequel le serveur se met en écoute des clients et prend en charge leur demande de connexion et de communication.
- **Mode client.** Le client Asterisk permet de se brancher au serveur Asterisk et de l'interroger pour lui demander des informations sur son état courant, ou bien pour lui donner de nouvelles directives qui seront prises en compte dynamiquement et modifieront son comportement.

Le mode client nous servira dans les sections suivantes à valider le fait que le serveur est correctement lancé et disponible. Il s'agit donc d'un outil d'administration interactif pratique, sans qu'il soit pour autant indispensable. Le lancement en mode client n'est possible que si une instance d'Asterisk en mode serveur a été préalablement effectuée.

Chacun de ces deux modes est obtenu par le biais de la commande *asterisk*, mais avec des arguments de commande différents dans chaque cas.

### Lancer Asterisk en mode serveur

Le mode serveur d'Asterisk peut être lancé de deux façons, selon que nous souhaitons un lancement automatique (à chaque démarrage du système d'exploitation) ou manuel (uniquement pendant la session courante).

Pour un lancement automatique du serveur, la commande est la suivante :

```
/usr/sbin/safe_asterisk
```

Pour un lancement manuel, la commande devient :

```
asterisk -vvvc
```

La succession des options *v* fournit un niveau de messages informatifs concernant le fonctionnement du serveur de plus en plus élevé. *vvv* est en fait l'acronyme de *very very verbose*. La lettre *c* demande qu'un message (indiquant que la console de contrôle d'Asterisk est bien activée) s'affiche devant chaque ligne. Le message est généralement *\*CLI>* (pour Command Line Interface).

### Se connecter à Asterisk en mode client

Une fois le serveur Asterisk démarré, nous pouvons vérifier qu'il est opérationnel grâce au client Asterisk. Ce dernier se connecte pour cela au serveur Asterisk afin de récupérer toutes sortes d'informations d'état du serveur et de connexion des utilisateurs.

Il est aussi possible de saisir des commandes d'administration afin de modifier le comportement courant du serveur. Pour exécuter le mode client d'Asterisk (on parle aussi de mode interactif), il suffit d'entrer la commande suivante dans un terminal :

```
asterisk -r
```

#### Remarque

Dès que cette commande est lancée, le client Asterisk se connecte au serveur Asterisk. Ce dernier est à l'écoute de requêtes de paramétrage et de gestion des connexions dont il a la charge. Ces requêtes sont effectuées au moyen d'une interface en ligne de commande, ou CLI (Command Line Interface). Pour connaître la liste des commandes disponibles, il suffit d'entrer la commande *help*.

Pour disposer d'informations plus complètes, la commande *set verbose* permet de récupérer les journaux.

Nous pouvons donc interroger le serveur pour en obtenir différentes informations, dont voici quelques exemples utiles :

- Savoir qui est connecté. Permet de connaître l'ensemble des utilisateurs (ou *peers*) connectés, selon le protocole qu'ils utilisent. Ce ne sont pas des utilisateurs forcément en cours de communication, mais simplement des utilisateurs connectés au serveur Asterisk, et donc joignables. Pour connaître les utilisateurs connectés qui utilisent le protocole SIP, on utilise la commande *sip show peers*, comme ci-dessous :

```
asterisk*CLI> sip show peers
Name/username  Host          Dyn Nat ACL  Port Status
101/101        192.168.1.1  D 5060      OK (20 ms)
103/103        192.168.1.3  D 5060      OK (17 ms)
105/105        192.168.1.5  D 5060      OK (22 ms)
3 sip peers [3 online , 0 offline]
```

De manière analogue, les utilisateurs qui utilisent le protocole IAX peuvent utiliser la commande *iax2 show peers*, comme ci-dessous :

```
asterisk*CLI> iax2 show peers
Name/Username  Host          Mask          Port  Status
202/202        192.168.1.2  (D) 255.255.255.255  4569 Unmonitored
204/204        192.168.1.4  (D) 255.255.255.255  4569 Unmonitored
206/206        192.168.1.6  (D) 255.255.255.255  4569 Unmonitored
207/207        192.168.1.7  (D) 255.255.255.255  4569 Unmonitored
4 iax2 peers [0 online, 0 offline, 4 unmonitored]
```

- Recharger les informations d'un fichier de configuration. Les fichiers de configuration sont en principe chargés au lancement du serveur Asterisk. Il est possible de les recharger dynamiquement pendant que le serveur Asterisk tourne, sans avoir à le redémarrer.

Pour recharger l'ensemble des fichiers de configuration, il suffit de saisir dans l'interface en ligne de commande la commande *reload*. Plus spécifiquement, il est possible de ne charger que les modifications apportées à un fichier. De manière générale, si le fichier a pour nom **fichier.conf**, il suffit de saisir la commande *fichier reload* pour recharger ce fichier. Par exemple, pour recharger le fichier **extensions.conf**, nous entrons la commande *extensions reload*. De la même manière, la commande *sip reload* permet de recharger le fichier **sip.conf**.

## Configuration

Le serveur Asterisk est à présent opérationnel. À ce stade, il n'assume encore aucune gestion des appels. L'ensemble des paramètres garantissant son mode de fonctionnement, et plus généralement son comportement pour la gestion des appels, se traduit par un ensemble de fichiers de configuration.

Plusieurs interfaces permettent de modifier ces fichiers. Quoique plus conviviales, ces interfaces n'en demeurent pas moins limitatives en comparaison des possibilités offertes en modifiant directement les fichiers de configuration. C'est pourquoi nous préférons présenter ces derniers.

### *Les quatre catégories d'éléments d'Asterisk*

Nous présentons ici les notions fondamentales du fonctionnement d'Asterisk. Pour comprendre et maîtriser le logiciel, il est indispensable de maîtriser ces notions avant de se lancer dans la configuration.

La configuration du serveur Asterisk comporte les quatre catégories d'éléments suivants :

- **Description des utilisateurs et des terminaux.** Pour être joignable, les utilisateurs (s'ils se connectent en utilisant un programme logiciel de type softphone) comme les terminaux (si les utilisateurs se connectent avec des composants matériels) doivent être identifiables. Un identifiant, qui peut être quelconque (sous forme de chaînes de caractères avec des chiffres ou des lettres), doit être affecté à chaque utilisateur (ou terminal). Une authentification peut être ajoutée pour s'assurer de l'identité des utilisateurs (ou terminaux) et préserver le système des utilisations frauduleuses. Les utilisateurs (et les terminaux) sont recensés dans des fichiers différents selon le protocole de signalisation qu'ils exploitent (SIP, H.323, IAX, etc.).

- **Plan de numérotation (ou dial plan).** Après avoir générer les comptes des utilisateurs, il faut ensuite déterminer de quelle manière ils vont communiquer, c'est-à-dire, d'abord, quel est le numéro de téléphone attribué à chaque terminal, mais aussi comment les appels et les services s'effectuent. La notion de plan de numérotation correspond précisément aux règles de routage des appels qui régissent le fonctionnement du système et permettent la mise en relation les interlocuteurs. Il convient de les programmer et de les analyser en profondeur, car elles concentrent l'intelligence de la plate-forme. En spécifiant la manière de réagir aux appels, le système fournit un premier niveau de service aux utilisateurs : le routage des communications. Un fichier unique définit le plan de numérotation (**extensions.conf**), lequel peut éventuellement appeler d'autres fichiers par inclusion.
- **Description des services supplémentaires.** Les services supplémentaires permettent d'assurer toutes sortes de fonctionnalités, telles que la messagerie téléphonique ou la journalisation des appels. Chaque service est défini dans un fichier spécifique, dont nous donnerons quelques exemples.
- **Description du matériel physique.** Si nous restons dans un réseau purement IP, aucun autre matériel qu'un ordinateur n'est nécessaire. Dès lors que nous souhaitons correspondre avec des utilisateurs du réseau téléphonique commuté ou de tout autre réseau non-IP, nous devons recourir à des équipements spécifiques. La description du matériel permet d'indiquer au serveur Asterisk la nature de ces composants, de façon qu'il en tienne compte. Les fichiers étant différents selon le matériel considéré, le serveur Asterisk doit être recompilé avec le chargement de ces fichiers.

Chacun de ces éléments est décrit par un ou plusieurs fichiers. La séparation de ces fichiers apporte une modularité à la gestion de la plate-forme. Nous décrivons à la section suivante les fichiers utilisés et la manière de les programmer. Puis, dans les sections suivantes, nous détaillerons les catégories d'éléments précédemment mentionnées pour expliquer successivement comment s'effectuent la description des utilisateurs et des terminaux, puis la programmation du plan de numérotation et enfin la mise en place de services complémentaires.

### **Organisation des fichiers (fichier asterisk.conf)**

Les fichiers d'Asterisk sont répartis dans plusieurs répertoires afin de suivre l'organisation classique des systèmes Linux. Le répertoire contenant les exécutable binaires du serveur Asterisk et ses composants principaux est situé par défaut dans le chemin **/usr/bin/**. Il comporte les commandes principales suivantes : *asterisk*, *astman*, *astgenkey*, *safe\_asterisk*.

Pour tous les autres fichiers non binaires, le fichier **asterisk.conf** offre une grande souplesse d'utilisation et laisse l'administrateur libre de modifier sa configuration par défaut, en spécifiant l'emplacement dans l'arborescence du système de fichiers utilisé.

En l'absence de ce fichier, les chemins par défaut sont considérés, comme dans l'extrait ci-dessous :

```
[directories]
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
astvarlibdir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run/asterisk
astlogdir => /var/log/asterisk
```

Le tableau 12.2 récapitule les descriptions correspondant aux différentes directives utilisables dans le fichier **asterisk.conf**.

**Tableau 12.2 – Directives indiquant l'emplacement des répertoires d'Asterisk**

Directive	Description
<i>astetcdir</i>	Spécifie le répertoire contenant l'ensemble des fichiers de configuration (portant l'extension <b>.conf</b> ).
<i>astmoddir</i>	Spécifie le répertoire contenant l'ensemble des modules (portant l'extension <b>.so</b> ).
<i>astvarlibdir</i>	Spécifie le répertoire contenant l'ensemble des bibliothèques exploitables, fournissant notamment une galerie de fichiers audio qui peuvent être utilisés dans des services avec des annonces automatisées du serveur (mise en attente, messagerie, annonces d'accueil, etc.) et seront appelés par leur nom dans la programmation de ces services.
<i>astagidir</i>	Spécifie le répertoire contenant l'ensemble des scripts AGI (Asterisk Gateway Interface), fournissant des fonctionnalités complémentaires implémentées avec un langage de programmation tel que Shell, C, PHP, Perl ou Pascal.
<i>astspooldir</i>	Spécifie le répertoire contenant l'ensemble des enregistrements audio qui sont réalisés par le serveur Asterisk, notamment pour la messagerie vocale, mais aussi pour l'enregistrement de communications, de conférences, etc.
<i>astrundir</i>	Spécifie le répertoire contenant le fichier de PID (Process ID) d'Asterisk identifiant de manière unique le processus en charge de la gestion du serveur, qu'il est possible de suivre parmi les autres processus.
<i>astlogdir</i>	Spécifie le répertoire contenant l'ensemble des fichiers journaux sauvegardant les événements survenus et le traitement qui leur a été appliqué. Il est important de vérifier que ce répertoire ne devienne pas rapidement trop volumineux pour surcharger les capacités du serveur.

À la suite de la section *[directories]*, on trouve une section *[files]* qu'il est vivement recommandé de ne pas modifier. Elle comporte des informations de sécurité, en particulier les droits restreignant l'accès au serveur Asterisk.

La section suivante détaille les principaux fichiers de configuration générale qu'il convient de connaître. De manière générale, un commentaire peut être inséré dans ces fichiers de configuration en le faisant précéder d'un point-virgule.

**Attention !**

Les fichiers ne sont pas interprétés dynamiquement. Autrement dit, pour qu'une modification effectuée dans un fichier soit prise en compte par le serveur en cours de fonctionnement, il est nécessaire de lui imposer un rechargement de ces fichiers par la commande *reload* (introduite à la section présentant la connexion en mode client à Asterisk).

### ***Première étape de configuration : Description des utilisateurs et des terminaux (fichiers *sip.conf*, *iax.conf*, *mgcp.conf*, *h323.conf*, *skinny.conf*)***

La première étape de configuration du serveur Asterisk correspond à la définition des comptes d'utilisateurs et des terminaux. Ceux-ci sont identifiés par le protocole de signalisation qu'ils utilisent. Il existe donc un fichier de configuration par protocole de signalisation supporté.

A ce stade, les numéros de téléphone ne sont pas associés aux comptes des utilisateurs et aux terminaux. Il faudra attendre la programmation du plan de numérotation pour mettre en place ces associations (voir l'exemple numéro 1, à la fin de la section suivante). Cette distinction entre un utilisateur (ou terminal) et son numéro offre une meilleure visibilité en regroupant les attributions de numéros, indépendamment des paramètres des comptes.

On notera que, de la même manière qu'on déclare un compte d'utilisateur lorsque celui-ci se connecte avec un softphone, il est possible de déclarer des téléphones physiques pour autoriser leur connexion dans le parc géré par Asterisk. On procédera pour cela de manière analogue, en déclarant un compte, et les propriétés associées à ce compte, comme on le ferait pour un utilisateur.

Nous détaillons dans les sections suivantes les fichiers ***sip.conf*** et ***iax.conf***.

#### ***Le fichier *sip.conf****

Le fichier ***sip.conf*** permet de définir tous les utilisateurs SIP. Il est segmenté en sections, dont chacune débute par une étiquette (le *label*) entre crochets.

Le label spécial *[general]* permet d'attribuer des valeurs à des paramètres génériques, tels que le port utilisé. Le label *[user\_id]* définit chaque compte d'utilisateur.

Voici un exemple de fichier **sip.conf**, dans lequel deux utilisateurs sont déclarés :

```
[general]
port=5060

[david]
username=David
secret=slp@st3r1sk!
callerid="David" <5551>
type=friend
host=dynamic
nat=yes
canreinvite=no
context=internal
disallow=all
allow=ulaw
allow=gsm
allow=h263

[laurent]
type=friend
host=dynamic
context=internal
username=Laurent
nat=yes
canreinvite
secret=h102m3E
callerid="Laurent" <5552>
```

La section *[general]* indique le numéro de port utilisé par tous les utilisateurs, soit ici *5060*. Les sections suivantes renseignent les paramètres de deux compte d'identifiant *david* et *laurent*. Les paramètres les plus utilisés pour la définition de ce compte sont récapitulés au tableau 12.6. Comme on le voit dans l'exemple, tous ne sont pas obligatoires (lorsqu'ils sont optionnels, des valeurs par défaut sont pratiqués si les paramètres ne sont pas présents), et l'ordre dans lequel ils sont donnés n'a aucune importance.

Tableau 12.6 – Paramètres décrivant un compte d'utilisateur

Paramètre	Description
<i>username</i>	Identifiant de l'utilisateur
<i>secret</i>	Mot de passe associé au compte
<i>type</i>	Indique le type de compte, et les restrictions associées. On distingue trois types de comptes : <ul style="list-style-type: none"> <li>– <i>Friend</i> : permet d'appeler et d'être appelé (autorise les appels entrants et sortants).</li> <li>– <i>User</i> : permet seulement d'être appelé (appels entrants).</li> <li>– <i>Peer</i> : permet de définir une liaison entre deux terminaux seulement.</li> </ul>
<i>host</i>	Spécifie une adresse IP à partir de laquelle l'utilisateur peut accéder à son compte. La valeur <i>dynamic</i> autorise une adresse IP fournie dynamiquement, par un serveur DHCP notamment. Cette valeur est donc moins restrictive.
<i>callerid</i>	Nom de l'utilisateur, entre guillemets, suivi de son extension téléphonique, c'est-à-dire de son numéro d'appel (au format de la RFC 822). Attention : le numéro de téléphone mentionné ici ne constitue pas une association du numéro avec l'utilisateur (cette association sera faite ultérieurement). Ce paramètre permet simplement d'identifier l'utilisateur (ou le terminal) lorsqu'il passe des appels. Autrement dit, cette information est utilisée dans les appels sortants uniquement pour indiquer le nom (si le terminal appelé permet d'afficher cette information) et le numéro de téléphone de l'utilisateur.
<i>context</i>	Spécifie le type de routage à appliquer pour l'utilisateur. Le type de routage correspond à un contexte défini dans le plan de numérotation (fichier <b>extensions.conf</b> ). Les communications avec cet utilisateur sont donc soumises au contexte du même nom dans le fichier <b>extensions.conf</b> .
<i>language</i>	Spécifie la langue utilisée pour les fichiers audio. Par exemple : <i>language=fr</i> .
<i>allow</i>	Liste les codecs autorisés par l'utilisateur de ce compte. Par exemple, pour autoriser le codec G.711, selon la loi mu, on saisira : <i>allow=ulaw</i> . Ce même paramètre est aussi valable pour spécifier les codecs vidéo (par exemple : <i>allow=h263</i> ) Il est possible d'en mentionner autant que l'on souhaite, en répétant ce paramètre autant de fois.
<i>disallow</i>	Interdit les codecs qui sont mentionnés à sa suite. Une valeur possible de ce paramètre est <i>all</i> . Dans ce cas, aucun codecs ne sera utilisable par l'utilisateur concerné, sauf ceux spécifiés explicitement dans le (ou les) paramètre(s) <i>allow</i> .
<i>nat</i>	Précise si les flux traversant un réseau utilisent la translation d'adresse (NAT). La valeur du paramètre <i>nat</i> est <i>yes</i> ou <i>no</i> . Ce paramètre est souvent indispensable car l'utilisation du nat est classique, même chez les particuliers.
<i>canreinvite</i>	Ce paramètre peut prendre les valeurs <i>yes</i> ou <i>no</i> . Si sa valeur est fixée à <i>yes</i> , Lorsqu'une communication est en train de s'établir le serveur Asterisk va récupérer des informations (notamment vers qui envoyer les flux) et ils les réémettra dans un nouveau message d'invitation une fois que la communication sera acceptée seulement. Attention, si l'utilisateur se trouve derrière un NAT, il est indispensable de mettre la valeur de ce paramètre à <i>no</i> pour laisser passer les flux multimédia correctement, car le nouveau message d'invitation du serveur Asterisk ne tiendrait pas compte du NAT.
<i>mailbox</i>	Indique la boîte vocale associée à ce compte. Nous détaillons ce paramètre et son utilisation à la section qui traite de la messagerie audio.
<i>dtmfmode</i>	Ce champ indique le type de tonalité DTMF qui sera appliqué. Les valeurs possibles sont <i>inband</i> , <i>rfc2833</i> , <i>info</i> ou <i>auto</i>

### Le fichier *iax.conf*

Les clients utilisant le protocole de signalisation IAX sont mentionnés dans le fichier **iax.conf**. Son fonctionnement et sa description sont semblables à ceux du fichier **sip.conf**.

Voici un exemple de fichier **iax.conf** définissant un utilisateur :

```
[general]
bindport=4569

[1234]
username=1234
type=friend
host=dynamic
context=internal
callerid="Guy" <1234>
```

Là aussi, comme pour le fichier **sip.conf** et contrairement à ce qu'on pourrait croire, les nombres ne sont pas des affectations de numéros de téléphones à des utilisateurs. Ce ne sont que des identifiants de comptes. Seul le plan de numérotation permet de réaliser la correspondance d'un numéro de téléphone avec un utilisateur.

### **Deuxième étape de configuration :** **le plan de numérotation (fichier *extensions.conf*)**

Une fois les comptes des utilisateurs et des terminaux définis, il faut leur attribuer des numéros de téléphone pour qu'ils soient joignables. Il faut aussi déterminer la procédure qui s'enclenchera lors de chaque appel (par exemple, si l'appel sera transmis vers deux postes en même temps, s'il sera redirigé vers une messagerie au bout d'un laps de temps, si l'appel sera journalisé dans une base de données, etc.).

Il faut enfin définir les services spéciaux que l'on souhaite activer (par exemple le service de messagerie, les serveurs vocaux disponibles, les salons de conférence mis en place, etc.). Tout cela s'effectue au moyen d'un unique composant : le plan de numérotation.

Le plan de numérotation, ou *dial plan*, est l'élément central de la configuration du serveur Asterisk. Il définit le comportement du serveur PBX. Maître de cérémonie ou chef d'orchestre, c'est lui qui régit les actions à entreprendre, dans quel ordre et dans quel cas, que ce soit pour un utilisateur donné ou pour l'ensemble des utilisateurs.

Ce plan concentre toute l'intelligence et la logique de fonctionnement du réseau téléphonique. C'est pourquoi il est indispensable d'en maîtriser à la fois la syntaxe et la sémantique. Il est constitué d'un ensemble de règles, dont chacune pose les conditions de

son application, ainsi que, lorsque ces conditions sont réunies, les traitements qui seront appliqués.

Le plan de numérotation est défini dans un fichier unique dont le nom est **extensions.conf**. Sous Linux, et avec une installation standard, ce fichier se trouve généralement dans le répertoire **/etc/asterisk/**.

Le plan de numérotation répond à la question : que doit faire le serveur PBX Asterisk lorsqu'il reçoit le flux téléphonique d'un utilisateur ? La réponse est fournie sous forme de règles qui sont structurées et dont la syntaxe est définie par les quatre éléments suivants :

- le contexte
- l'identifiant d'extension
- la priorité
- l'application

Ces éléments décrivent les critères que les flux doivent vérifier et le traitement qui leur sera appliqué le cas échéant.

Le format général d'un plan de numérotation, dans lequel se combinent ces quatre éléments, est le suivant :

```
[contexte_1]
exten => identifiant_d'extension_1, priorité_1, application_1
exten => identifiant_d'extension_2, priorité_2, application_2
exten => identifiant_d'extension_3, priorité_3, application_3

[contexte_2]
exten => identifiant_d'extension_4, priorité_4, application_4
```

On distingue dans cet exemple deux contextes différents, signalés par *[contexte\_1]* et *[contexte\_2]*. Le mot-clé *exten* est utilisé pour définir une extension. Il est suivi d'une flèche, formée par les symboles = et >.

Dans notre exemple, trois extensions sont définies dans le premier contexte, et une dans le second. Chaque extension comporte un identifiant d'extension (*identifiant\_d'extension\_i*), un numéro de priorité (*priorité\_i*) et une fonction applicative (*application\_i*). Chacun de ces critères permet de préciser qui est l'appelant, avec quel service (ou personne) il souhaite être mis en relation et comment effectuer la fourniture de ce service.

Nous pouvons lire la première règle comme suit : « Lorsque l'extension *identifiant\_d'extension\_1* se présente dans le contexte *contexte\_1*, nous exécutons l'action *application\_1* avec la priorité *priorité\_1* ». Il est nécessaire de maîtriser ces éléments

pour affecter un numéro de téléphone à un utilisateur (ou à un terminal) et plus généralement pour déterminer la manière dont les appels se déroulent. Les sections suivantes précisent le rôle de chacun de ces quatre éléments.

### D'abord le contexte

Le plan de numérotation est organisé en sections appelées *contextes*. Un contexte définit un cadre d'application. Par exemple, si un utilisateur compose le chiffre 5 sur son poste téléphonique, ce numéro est intercepté par le serveur Asterisk, lequel peut interpréter différemment ce numéro selon différentes situations, notamment les suivantes :

- Si l'utilisateur ne consulte aucun service, le chiffre 5 peut correspondre au numéro de téléphone abrégé que le serveur a enregistré pour un de ces contacts.
- Si l'utilisateur a préalablement demandé à effectuer un appel longue distance, le chiffre 5 peut être le premier d'un code d'authentification permettant la mise en relation.
- Si l'utilisateur est en relation avec un service vocal, le chiffre 5 peut correspondre au choix d'un service spécifique parmi ceux proposés.
- Si l'utilisateur est en train de consulter son répondeur hébergé par le serveur Asterisk, le chiffre 5 peut être un code de commande pour demander au serveur de réécouter un message.
- Si l'utilisateur est en cours de communication avec un contact, le chiffre 5 peut demander au serveur Asterisk de lancer l'enregistrement de la conversion en cours.

Le serveur Asterisk ne doit donc pas interpréter de la même façon toutes les saisies effectuées sur un terminal téléphonique, mais distinguer les cas de figure. Dans notre exemple, les contextes correspondent à l'état dans lequel l'appel s'inscrit : l'utilisateur peut être en ligne, non authentifié, en relation avec le service vocal, en relation avec un service de messagerie vocale ou en cours de communication. D'où nos cinq contextes.

Pour savoir quel contexte utiliser lors d'un appel, le plan de numérotation se fonde sur les éléments suivants :

- Identité de l'appelant ou de l'appelé, auquel il est possible d'attribuer des contextes particuliers.
- Actions entreprises par les utilisateurs pendant un appel : la saisie de touches par un utilisateur peut engendrer des branchements conditionnels ou inconditionnels (présentée plus loin avec les structures *goto* et *gotoIf*).

De la même manière, les contextes permettent de gérer des catégories d'utilisateurs. Nous pouvons, par exemple, définir un contexte pour les employés commerciaux autorisés à passer des appels longue distance, et un autre pour les employés ingénieurs uniquement autorisés à passer des appels locaux. Dans un cadre domestique, nous pouvons définir un contexte pour les personnes adultes, dans lequel tous les appels sont possibles, et un autre pour les enfants, dans lequel seuls des numéros d'appel prédéfinis sont possibles.

La puissance du serveur Asterisk réside dans cette capacité de personnalisation, qui peut concerner un utilisateur ou un service spécifique, comme dans les cas suivants :

- Un utilisateur peut disposer de services différents de ceux disponibles pour un autre utilisateur. Par exemple, si le poste d'un utilisateur est indisponible au bout de cinq sonneries, l'appel peut être redirigé vers le poste d'un autre utilisateur, tandis que le même cas de figure pour un autre utilisateur déclenche la messagerie.
- Tous les services étant personnalisables par l'administrateur, un administrateur peut décider que la messagerie des utilisateurs se déclenche au bout de cinq sonneries, alors qu'un autre peut décider qu'elle ne s'active qu'au bout de sept sonneries.

À chaque contexte, une politique de gestion particulière peut être définie sous la forme d'une ou plusieurs extensions. Chaque contexte regroupe un ensemble d'extensions constituées d'un identifiant d'extension, d'une priorité et d'une application.

### Les contextes particuliers

Il existe deux contextes particuliers, appelés *[general]* et *[globals]* et tous deux placés au tout début du fichier **extensions.conf**.

#### [general]

Toujours mentionnée avant les autres, cette section permet de définir des options générales appliquées par le serveur Asterisk au plan de numérotation. Par exemple, on peut indiquer à la suite de la section :

```
clearglobalvars=yes
```

Cela a pour effet d'effacer l'ensemble des variables globales enregistrées par le serveur Asterisk. Au contraire, la valeur *no* a pour effet de conserver aux variables globales des valeurs persistantes à chaque rechargement du serveur, même si elles ne sont plus spécifiées dans les fichiers de configuration.

De même, en mentionnant :

```
static=yes  
writeprotect=no
```

il est possible de sauvegarder le plan de numérotation par l'interface de contrôle (CLI) d'Asterisk présentée précédemment.

Il est également vivement recommandé de choisir :

```
autofallthrough=yes
```

Cette valeur, prise par défaut dans les versions récentes d'Asterisk, permet de terminer automatiquement un appel pour lequel tous les traitements mentionnés dans le plan de numérotation ont été effectués. À défaut, si ce paramètre a pour valeur *no*, le serveur attend, une fois tous les traitements effectués, que l'utilisateur effectue une nouvelle saisie.

[globals]

La section [globals] permet de définir toutes les variables globales susceptibles d'être utilisées dans la suite du fichier. Nous verrons plus loin comment définir de telles variables.

### Ensuite, l'identifiant d'extension

Au sein de chaque contexte, des règles permettent de définir le comportement à adopter, autrement dit la manière dont le routage et le service doivent être rendus.

L'identifiant d'extension définit la destination d'un appel. En première approximation, c'est le numéro d'un poste appelé. Nous verrons que cela peut aussi être un service déterminé ou un choix dans un menu vocal. Il correspond donc à un numéro (ou à un nom) d'appel, attribué indifféremment à une personne physique ou à un service automatisé.

À titre d'exemple, le tableau 12.3 donne quelques identifiants d'extension avec leur assignation respective.

**Tableau 12.3 – Exemples d'identifiants d'extension**

Identifiant d'extension	Description
0	Accueil
5	Répondeur téléphonique
101	Poste téléphonique d'Albert
102	Poste téléphonique de Béatrice
103	Poste téléphonique de Caroline

Le choix des identifiants d'extension est laissé à la convenance de l'administrateur. Les identifiants peuvent être formés de chiffres (comme pour un numéro de téléphone classique) ou de lettres (comme le nom d'une personne ou d'un service).

#### Remarque

Les espaces sont prohibées. En cas d'utilisation d'espaces, le comportement du système devient imprévisible, même si aucune erreur n'est retournée.

Les utilisateurs peuvent être identifiés par leur adresse e-mail. Si l'identifiant d'extension comporte des lettres, cela implique que le terminal appelant dispose du matériel adéquat, de type clavier, pour saisir l'identifiant d'appel. On peut toutefois recourir à d'autres mécanismes, comme la reconnaissance vocale.

Une règle ne peut être atteinte que lorsque l'exécution de l'action relative à la règle précédente est terminée. Les raisons à cela sont, d'une part, que deux actions peuvent entrer en concurrence (diffusion d'un message sonore avec deux actions, dont la lecture conjointe rendrait les deux messages inaudibles), d'autre part, qu'une règle peut entraîner des sorties conditionnelles ou des sauts vers une autre règle selon un déroulement particulier (par exemple, la saisie d'un choix par l'appelant peut entraîner une autre action que l'absence de choix).

### Les extensions particulières

Asterisk définit quelques extensions particulières permettant non pas de joindre un utilisateur ou un service, mais de fournir un comportement explicite pour des situations particulières. Les trois extensions prédéfinies les plus courantes sont *s*, *t* et *i*.

#### L'extension *s*

Lorsqu'un flux téléphonique parvient au serveur Asterisk, celui-ci peut lui appliquer un traitement indifférent de la personne ou du service que le flux tente de contacter. Pour cela, l'administrateur utilise l'extension *s* (pour le mot-clé *start*), qui indique que tous les flux (dans le contexte concerné) seront traités par la règle mentionnée.

En voici un exemple :

```
exten => s, 1, Playback (msg_attente)
```

Tout appel soumis à cette règle lance un message audio nommé *msg\_attente* pour indiquer à l'appelant que la communication est prise en compte et mise en attente.

#### L'extension *t*

Lorsqu'un certain délai (par défaut 10 secondes) s'écoule sans qu'une extension ait été saisie par l'utilisateur, l'extension *t* (pour le mot-clé *timeout*) est automatiquement appelée par le système.

En voici un exemple :

```
exten => t, 1, Playback (msg_delai_depasse)
```

Si un délai *t* (par défaut 10 secondes) s'écoule, un message nommé *msg\_delai\_depasse* est automatiquement lancé pour avertir l'utilisateur de l'attente d'une saisie par le système.

### L'extension *i*

Lorsque l'utilisateur saisit une extension qui n'est pas référencée dans le plan de numérotation, l'extension *i* (pour le mot-clé *invalid*) est automatiquement appelée par le système.

En voici un exemple :

```
exten => i, 1, Playback (msg_invalide)
```

Si l'utilisateur saisit une extension inconnue, un message nommé *msg\_invalide* est automatiquement lancé pour l'avertir de l'invalidité de sa saisie.

### Les filtres d'extension

Il est possible de définir des identifiants d'extension formés d'un filtre, ou *pattern*, qui représente une syntaxe générique d'identifiant. Cela permet d'offrir un service générique à des groupes d'utilisateurs ou des services spécifiques. En particulier, cela permet de distinguer les appels locaux des appels internationaux en fonction des préfixes de numérotation.

Tout filtre d'extension est précédé d'un caractère de soulignement (*underscore*). Les caractères spéciaux permettant de définir un filtre sont définis comme indiqué au tableau 12.4.

**Tableau 12.4 – Filtres d'extension**

Filtre	Description
<i>chaîne_quelconque</i>	Impose la présence de la chaîne <i>chaîne_quelconque</i> dans l'identifiant d'extension.
<i>[caractères_quelconques]</i>	Remplace un caractère dans un identifiant d'extension parmi l'un de ceux mentionnés entre les crochets.
<i>X</i>	Remplace un chiffre entre 0 et 9 dans un identifiant d'extension.
<i>Z</i>	Remplace un chiffre entre 1 et 9 dans un identifiant d'extension.
<i>N</i>	Remplace un chiffre entre 2 et 9 dans un identifiant d'extension.
<i>.</i>	Remplace n'importe quel caractère ou série de caractères. C'est donc un caractère joker qui ne devrait être indiqué qu'avec un filtre suffisamment descriptif.

La position des caractères spéciaux doit être respectée pour correspondre à l'identifiant d'extension filtré.

L'exemple suivant :

```
_0142XXXXXX
```

s'applique à n'importe quelle extension commençant par *0142* et ayant une longueur de 10 chiffres.

L'exemple suivant :

```
_0Z[12589]XXX
```

s'applique à une extension ayant pour premier caractère le chiffre *0*, pour deuxième caractère un chiffre entre *1* et *9*, pour troisième caractère un chiffre parmi les valeurs *1*, *2*, *5*, *8* ou *9*, puis, pour les trois caractères suivants (les trois symboles *X*), une valeur quelconque entre *0* et *9* et enfin pour le septième caractère (le symbole point) un ou plusieurs chiffres quelconques. Au total, le numéro fait un minimum de sept chiffres, le maximum n'étant pas mentionné.

Le filtre `_` (underscore suivi d'un point) remplace n'importe quel caractère ou série de caractères, autrement dit il s'applique à toutes les extensions. Ce filtre d'extension ne devrait donc jamais être utilisé, puisqu'il est toujours vérifié et s'applique sans restriction à tous les appels.

### Les extensions conditionnelles

Une extension définit en principe le numéro d'une personne ou d'un service à joindre, mais elle peut être conditionnée par l'identité de la personne appelante. En effet, un même numéro d'appel peut donner lieu à des traitements différents. Par exemple, dans une entreprise, un unique numéro d'appel pour un secrétariat peut être affecté à deux secrétaires ayant chacune leur spécialisation. Selon la personne appelant le secrétariat, une redirection vers le poste le plus adéquat peut s'effectuer. Il suffit pour cela de mentionner le numéro d'appel de la personne qui appelle juste après l'extension du service à joindre suivi d'un caractère slash.

Par exemple :

```
exten => 101/105, ...
```

spécifie une règle concernant l'appelant ayant pour identifiant *105* et appelant le poste *101*.

Les extensions conditionnelles peuvent utiliser des filtres, à la fois pour les identifiants de l'appelé et ceux de l'appelant.

### Puis, la priorité

La priorité définit l'ordre dans lequel la règle doit s'appliquer. Certains traitements nécessitent plusieurs actions à entreprendre pour obtenir le comportement désiré. Par exemple, si un correspondant ne répond pas, l'appelant peut être redirigé vers la messagerie de son correspondant. Dans ce cas, au moins deux étapes correspondent à un même appel : la tentative d'appel vers le correspondant, puis la redirection vers la messagerie.

Il faut ordonner chacune de ces étapes en mentionnant dans le plan de numérotation la priorité affectée à chacune des règles.

La priorité est spécifiée par une valeur numérique entière débutant par la valeur 1. Plus la valeur attribuée à une règle est faible, plus la priorité qui lui est concédée est importante. Les règles se succèdent tour à tour avec un même identifiant d'extension, selon un ordre croissant.

**Attention !**

Si une priorité est oubliée (par exemple si l'on a une priorité 1 puis une priorité 3, et donc qu'on a omis la priorité 2), le serveur ne peut poursuivre (donc passer à la priorité 3 dans notre exemple). Il importe de veiller à ce que l'ordre affecté à chaque règle n'omette pas une valeur. Lors d'une mise à jour, en particulier, il ne faut pas effacer une règle sans vérifier la cohérence et l'enchaînement avec les priorités qui suivent.

**La priorité  $n$** 

Il est possible d'optimiser la gestion des priorités en se fiant à l'emplacement d'écriture des règles pour en fixer l'ordre.

La notion de priorité numérique présente une contrainte délicate dans le maintien et la mise à jour des règles. Si nous souhaitons ajouter une nouvelle règle entre deux règles, il est nécessaire de modifier explicitement toutes les priorités des règles qui suivent la règle ajoutée.

Considérons la succession de règles suivantes :

```
exten => 98765, 1, action_1
exten => 98765, 2, action_2
exten => 98765, 3, action_3
```

Les applications spécifiées *action<sub>i</sub>* sont quelconques. Après la première règle, nous souhaitons ajouter une nouvelle règle exécutant l'action *action\_nouvelle*. Nous modifions comme suit la séquence précédente :

```
exten => 98765, 1, action_1
exten => 98765, 2, action_nouvelle
exten => 98765, 3, action_2
exten => 98765, 4, action_3
```

Cette manière de procéder est contraignante en ce qu'elle impose chaque fois de vérifier la cohérence des ordonnancements. Asterisk propose une manière plus simple de gérer les priorités grâce à la priorité  $n$ . Le  $n$  spécifie l'action suivante (en anglais *next*). Avec une priorité  $n$ , les étapes sont suivies selon leur ordre d'apparition dans la séquence du plan de numérotation.

L'exemple précédent initial devient ainsi :

```
exten => 98765, 1, action_1
exten => 98765, n, action_2
exten => 98765, n, action_3
```

L'ajout de la nouvelle règle donne simplement :

```
exten => 98765, 1, action_1
exten => 98765, n, action_nouvelle
exten => 98765, n, action_2
exten => 98765, n, action_3
```

Comme nous le constatons, il n'est pas nécessaire de modifier les règles suivant l'ajout d'une nouvelle action. L'enchaînement des règles est géré automatiquement.

### Enfin, l'application

L'application définit l'action à entreprendre pour appliquer le service sollicité par l'utilisateur appelant. Le serveur Asterisk dispose d'un grand nombre de procédures déterminant le comportement à adopter, c'est-à-dire la manière de traiter les flux audio.

Le tableau 12.5 décrit quelques applications classiques.

**Tableau 12.5 – Applications les plus courantes**

Application	Description
<i>Answer</i>	Répond à un appel téléphonique entrant.
<i>Background</i>	Lit un message audio de manière non bloquante. Autrement dit, une saisie d'une ou plusieurs touches par l'appelant est possible en parallèle. Dans ce cas, la lecture du message audio en cours est interrompue. L'extension correspondant aux touches saisies par l'appelant est automatiquement appelée. Cela permet notamment de présenter une information facultative, comme une publicité ou un menu, que l'appelant peut interrompre dès qu'il a pris connaissance de l'option qui l'intéresse.
<i>Dial</i>	Met en relation l'appelant et l'utilisateur ou le service spécifié en argument de l'application <i>Dial</i> . Cette commande est donc utilisée pour affecter un numéro de téléphone à un utilisateur, à un terminal ou à un service. En argument de l'application, il faut indiquer le nom du compte à contacter, préfixé par le protocole de signalisation que le compte utilise (IAX2, SIP, etc.). Cela permet au serveur Asterisk de déterminer dans quel fichier de configuration se trouvent les propriétés du compte appelé ( <b>iax.conf</b> , <b>sip.conf</b> , etc.). Nous verrons cela un peu plus loin, dans l'exemple 1. Il est possible d'appeler plusieurs correspondants en même temps en mettant leur extension respective en arguments de l'application <i>Dial</i> , séparés par le caractère & (esperluette). Par exemple : <i>Dial (SIP/1001 &amp; SIP/1005)</i> fait sonner les postes affectés aux comptes 1001 et 1002 en même temps. Il est aussi possible d'ajouter un argument à l'application mentionnant la durée pendant laquelle la tentative d'appel doit s'effectuer. Par exemple : <i>dial (SIP/identifiant, 30)</i> permet de faire sonner le téléphone de l'identifiant pendant une durée de 30 secondes.

Tableau 12.5 – Applications les plus courantes

Application	Description
<i>Echo</i>	Retourne l'écho de ce que l'appelant prononce. Cette application est utile notamment pour tester que les composants audio de réception et d'émission du son sont fonctionnels.
<i>Goto</i>	Branchement inconditionnel vers un contexte, une extension et une priorité particuliers.
<i>Gotolf</i>	Branchement conditionnel vers un contexte, une extension et une priorité particuliers lorsqu'une condition est vérifiée.
<i>GotolfTime</i>	Branchement conditionnel vers un contexte, une extension et une priorité particuliers lorsqu'une condition temporelle est vérifiée.
<i>Hangup</i>	Termine une communication.
<i>Meetme</i>	Invite un utilisateur à participer à une conférence identifiée en argument de l'application <i>Meetme</i> .
<i>Playback</i>	Lit un message audio de manière bloquante : la lecture du message doit se faire intégralement, et l'appelant ne peut interrompre cette diffusion par une saisie de touche sur le clavier téléphonique.
<i>Queue</i>	Met en attente une communication.
<i>Read</i>	Lit une variable. L'appelant est invité à entrer une valeur qui est sauvegardée sous forme de variable par le système. Cela permet notamment de demander un mot de passe à l'utilisateur avant d'accéder à un service spécifique.
<i>Record</i>	Enregistre une communication dans un fichier son.
<i>ResponseTimeout</i>	Spécifie en argument un délai au bout duquel l'attente du serveur expire.
<i>SayAlpha</i>	Annonce vocale de caractères textuels spécifiés en argument de l'application
<i>SayDigits</i>	Annonce vocale de chiffres spécifiés en argument de l'application
<i>SayNumber</i>	Annonce vocale de nombres spécifiés en argument de l'application
<i>SayPhonetic</i>	Annonce vocale d'un message phonétique spécifié en argument de l'application
<i>SayUnixTime</i>	Annonce vocale de l'heure, selon différents formats
<i>SendText</i>	Envoie un message textuel à l'utilisateur.
<i>SMS</i>	Envoi et réception de messages instantanés SMS
<i>System</i>	Exécute une commande système du système d'exploitation.
<i>Transfer</i>	Transfère l'appel vers un autre poste ou service.
<i>VoiceMail</i>	Laisse un message vocal.
<i>VoiceMailMain</i>	Accède au système de messagerie vocale.
<i>Wait</i>	Attend pendant un certain délai spécifié en argument.
<i>WaitExten</i>	Attend une saisie d'une extension par l'utilisateur.

Ces applications peuvent prendre aucun ou plusieurs arguments donnés à la suite du nom de l'application. Lorsque plus d'un argument doit être fourni à une application, les arguments se succèdent avec un caractère de séparation qui peut être indifféremment une virgule ou un pipe (|). Les deux formes suivantes sont donc équivalentes :

```
Mon_application (argument1 , argument2)
```

```
Mon_application (argument1 | argument2)
```

Les extraits de plan de numérotation suivants illustrent quelques exemples simples, en les commentant. On suppose que les règles suivantes s'appliquent dans le contexte courant ; nous verrons plus loin comment basculer sur un contexte particulier.

### Exemple 1

Voici un exemple important dans lequel nous allons affecter un numéro de téléphone à des utilisateurs (ou à des terminaux).

La partie précédente nous avait permis de déclarer des comptes d'utilisateurs (ou des terminaux). Nous avons ainsi été amenés à définir deux comptes SIP (dont les identifiants étaient *david* et *laurent*) et un compte IAX (dont l'identifiant était *1231*). Voici comment effectuer les affectations des numéros attribués aux utilisateurs avec leur compte respectifs.

```
exten => 5551, 1, Dial (SIP/david)
exten => 5552, 1, Dial (SIP/laurent)
exten => 1234, 1, Dial (IAX2/1234)
```

Si un appelant compose le numéro 5551, il sera mis en relation avec l'utilisateur SIP (dont la définition doit impérativement avoir été faite dans le fichier **sip.conf**), qui a pour identifiant *david*. Il en va de même pour l'identifiant *laurent*, avec le numéro 5552, et pour le numéro 1234, qui est lié au compte 1234, lequel utilise le protocole de signalisation *IAX2* (le compte doit être défini dans le fichier **iax.conf**).

### Exemple 2

En composant le numéro *54321*, les règles suivantes sont enclenchées.

```
exten => 54321, 1, Answer()
exten => 54321, 2, Echo()
exten => 54321, 3, Playback(vm-goodbye)
exten => 54321, 4, Hangup()
```

Lorsque l'appelant compose l'extension *54321*, les étapes suivantes sont lancées :

1. Accepte l'appel et y répond.
2. Renvoie en écho tout ce que dit l'appelant. L'écho se termine lorsque l'appelant saisit la touche dièse.
3. Diffuse un message audio intitulé *vm-goodbye* fourni par défaut pour dire au revoir à l'appelant.
4. Met fin à l'appel.

### Exemple 3

En appelant un numéro de téléphone prédéfini, nous souhaitons que le serveur Asterisk nous retourne l'heure en cours. Il s'agit en quelque sorte d'une horloge parlante. Les commandes suivantes permettent de fournir ce service :

```
exten => 777, 1, Answer
exten => 777, 2, SayUnixTime(,CET,km)
exten => 777, 3, Hangup
```

La première ligne accepte l'appel sur le numéro 777, et la dernière le termine. L'application *SayUnixTime* donne l'heure en cours par un message vocal.

## Tester la configuration d'un client

Les équipements des utilisateurs peuvent être divers : téléphone IP, PDA ou ordinateur. Nous allons utiliser dans cette section une solution générique de téléphonie SIP, avec le logiciel grand public gratuit X-Lite.

Simple d'utilisation, ce freeware est disponible en téléchargement (en anglais uniquement pour le moment), sur le site de l'éditeur CounterPath (anciennement XTEN), à l'adresse <http://www.counterpath.com/index.php?menu=download>.

La figure 12.3 illustre l'interface du logiciel. Parmi les autres clients SIP performants, signalons les logiciels SJPhone et LinPhone.

Figure 12.3

*Le logiciel client X-Lite*



Pour configurer le client X-Lite, un simple clic droit n'importe où dans l'interface ouvre un menu contextuel permettant d'accéder au menu « SIP Account Settings ».

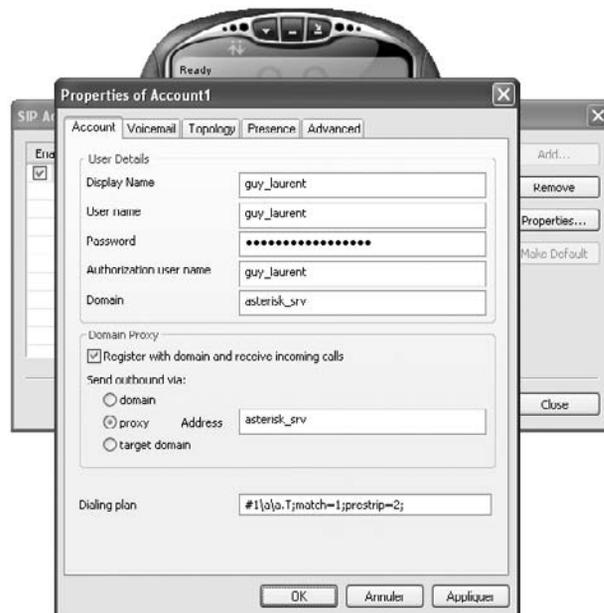
Dans la fenêtre qui s'ouvre, il suffit de renseigner les champs illustrés à la figure 12.4 :

- identifiant affiché pour l'utilisateur (Display Name), pouvant être formé de caractères alphanumériques ;
- identifiant servant à loguer l'utilisateur (User Name) ;
- mot de passe associé (Password) ;
- nom sous lequel l'autorisation d'accès est possible (Authorization user name) ;
- nom de domaine (Domain) ;
- adresse du serveur proxy (Proxy Address), c'est-à-dire le serveur Asterisk lui-même (dans notre cas asterisk\_srv).

Il est possible d'indiquer son nom si celui-ci est résolu par un serveur DNS en local ou, à défaut, d'utiliser l'adresse IP du serveur Asterisk.

Pour que l'authentification soit possible, ces valeurs doivent être conformes à celles saisies dans le fichier **sip.conf** du serveur Asterisk.

**Figure 12.4**  
*Configuration de X-Lite*



Une fois la configuration achevée, il faut valider ces paramètres en cliquant sur le bouton OK, non sans avoir pris soin de vérifier que la case « Enabled » est cochée pour le compte SIP que l'on vient de créer. Le bouton Close permet de fermer l'interface de gestion des comptes SIP. Le logiciel s'authentifie alors automatiquement auprès du serveur Asterisk mentionné.

Si cette étape s'effectue avec succès, un message « ready » s'affiche, indiquant que les communications sont désormais possibles. À défaut, un message d'erreur explique le motif qui a fait échouer le processus. Dans ce cas, il convient de vérifier la validité des paramètres du compte SIP (nom et mot de passe conformes au fichier **sip.conf**) et l'exactitude de l'adresse IP du serveur Asterisk. Vérifier au besoin qu'un pare-feu ne bloque pas les communications SIP entre le client et le serveur.

## Optimiser les traitements

La section `[globals]` du fichier **extensions.conf** permet de définir des variables, comme dans un langage de programmation.

La syntaxe pour définir une variable est la suivante :

```
Nom_de_variable => Valeur_de_variable
```

L'exemple suivant définit les variables *Numero\_de\_Guy* et *MusiqueAttente* respectivement initialisées par les valeurs *4321* et */fichier\_sons/son\_opera.gsm* :

```
[globals]
Numero_de_Guy => 4321
MusiqueAttente => /fichier_sons/son_opera.gsm
```

Comme nous le constatons, les variables peuvent contenir des valeurs de nature différente, numériques ou textuelles, ou encore des chemins d'arborescence.

Pour utiliser ces variables, il suffit de reprendre leur nom, encadré d'accolades et préfixé par le caractère \$ (dollar). En indiquant `${Numero_de_Guy}` dans le plan de numérotation, c'est la valeur *4321* qui est placée pour cette variable. Si, ultérieurement, le numéro du poste de l'utilisateur *Guy* change, il suffit de remplacer l'unique ligne initialisant la valeur de la variable *Numero\_de\_Guy* à la nouvelle valeur. Elle est aussitôt prise en compte dans le plan de numérotation sans autre changement.

Il existe des variables spéciales, qui sont préconfigurées par le serveur Asterisk. Il est indispensable de respecter les majuscules et les minuscules dans l'écriture de ces variables. Par exemple :

- EXTEN représente l'identifiant d'extension courante.
- CALLERID(all) représente le nom et le numéro de l'appelant.
- CALLERID(name) représente seulement le nom de l'appelant.
- CALLERID(num) représente seulement le numéro de l'appelant.

- DIALEDTIME représente la durée de l'appel courant.
- DATETIME représente la date courante (son usage est déprécié).
- DIALSTATUS représente l'état de l'appel en cours.

### La directive d'inclusion

Si un fichier est trop volumineux et perd en lisibilité, il est possible de le scinder en plusieurs fichiers. Il suffit pour cela d'utiliser la directive *include*, comme dans les exemples suivants.

Cas 1 : sans inclusion de fichier (un seul fichier) :

```
fichier_1 :           ligne_1
                    ligne_2
                    ligne_3
                    ligne_4
                    ligne_5
```

Cas 2 : avec inclusion de fichier (deux fichiers) :

```
fichier_1 :           #include fichier_2
                    ligne_4
                    ligne_5

fichier_2 :           ligne_1
                    ligne_2
                    ligne_3
```

Dans ces exemples, les deux cas sont parfaitement identiques, si ce n'est que le code a été segmenté en deux parties dans le second. La directive optimise la qualité du code en évitant la redondance de code ou en allégeant les fichiers longs.

### Logique de programmation

Parmi les applications disponibles dans la définition du plan de numérotation, on retrouve les classiques procédures de programmation de branchements (ou sauts) conditionnels et inconditionnels. Les applications *Goto()* et *GotoIf()* implémentent ainsi respectivement un branchement inconditionnel et un saut conditionnel.

L'application *Goto* prend comme argument un label spécifiant où doit s'effectuer le branchement. Un label peut être défini exhaustivement par la fourniture de trois éléments : un contexte, une extension et une priorité. Il n'est toutefois pas indispensable de spécifier le

contexte et l'extension. Par défaut, si aucun contexte n'est donné, c'est le contexte courant qui est considéré. De même, si aucune extension n'est fournie, c'est l'extension par défaut qui est prise en compte. La mention de la priorité est le seul élément obligatoire.

Voici un exemple d'application *Goto* :

```
[mon_annonce]
exten => s, 1, Playback(msg_indication)
exten => s, 2, ResponseTimeout(5)
exten => s, 3, WaitExten

exten => #, 1, Goto(mon_service, s, 1)
exten => *, 1, Goto(mon_menu_principal, s, 1)
exten => i, 1, Goto(s, 1)
exten => t, 1, Playback(msg_au_revoir)
exten => t, 2, Hangup
...

[mon_service]
exten => s, 1, ...
...

[mon_menu_principal]
...
```

Nous considérons que la communication est gérée par le contexte *[mon\_annonce]*. Le serveur réagit en fonction de la saisie de l'utilisateur.

La première extension, *s* (pour *start*), est toujours activée par défaut. La première étape consiste en la diffusion d'un message audio nommé *msg\_indication* (qui n'existe pas par défaut), indiquant à l'appelant que ce dernier doit saisir une touche au clavier. La ligne suivante limite l'attente de cette saisie à 5 secondes. La troisième étape de l'extension *s* enclenche l'attente de la saisie.

Les quatre cas suivants peuvent se produire :

- Si l'utilisateur saisit la touche dièse, il est orienté (par l'application *Goto*) vers le contexte *[mon\_service]* (premier argument de l'application *Goto*) à l'extension *s* (deuxième argument de l'application *Goto*) et sur l'étape numéro 1 (troisième argument de l'application *Goto*). La ligne sur laquelle le branchement doit s'effectuer est partiellement mentionnée plus bas, sans être implémentée : elle succède à la ligne *[mon\_service]*. C'est là que figure la prochaine règle examinée pour le traitement correspondant à la saisie de la touche dièse.

- Si l'utilisateur saisit la touche étoile, le traitement se poursuit sur la première étape de l'extension *s* du contexte *[mon\_menu\_principale]* (également mentionné plus bas).
- Si l'utilisateur saisit une autre touche que dièse ou étoile, la procédure reprend depuis l'étape précédente, c'est-à-dire à la première étape de l'extension *s* du contexte *[mon\_annonce]* (correspondant au contexte courant, qu'il n'est donc pas nécessaire de préciser). L'annonce l'invitant à saisir une touche est à nouveau diffusée, et le système se place en attente de cette saisie.
- Si le délai imparti est écoulé, un message *msg\_au\_revoir* (n'existant pas par défaut) est diffusé à l'appelant pour lui signifier qu'aucun signal n'a été reçu. La communication s'arrête.

L'application *GotoIf* permet d'effectuer un branchement conditionnel. En argument de la fonction, il suffit de mentionner la condition à réaliser, suivie d'un point d'interrogation puis du label désignant le branchement effectué si la condition est vérifiée, et de terminer par le caractère deux-points suivi du label désignant le branchement qui sera effectué si la condition n'est pas vérifiée.

Il est possible d'omettre l'un des deux labels et de désigner simplement le branchement dans le cas où la condition est (ou n'est pas) réalisée. Il n'est pas possible d'omettre les deux labels simultanément.

Voici un exemple d'application *GotoIf* :

```
exten => 7979, 1, Read (var_secret, rc-code, 4)
exten => 7979, 2, GotoIf ($[${var_secret} = 1234] ? code_valide,s,1 : code_invalide
s, 1)

[code_valide]
...
[code_invalide]
...
```

Ce code spécifie pour l'extension *7979* la lecture d'une variable *var\_secret*, qui est sollicitée par un message vocal nommé *rc-code* de 4 chiffres (l'absence de la valeur 4 en argument entraînerait l'acceptation d'autant de chiffres que l'utilisateur le souhaite, la séquence étant terminée et validée par l'appui sur la touche dièse).

Dans la deuxième étape de cette extension, la valeur de la variable saisie par l'utilisateur est comparée à la valeur *1234*. Si l'utilisateur a effectivement saisi cette valeur, la prochaine étape examinée est l'étape 1 de l'extension *s* du contexte *[code\_valide]*. Si ce n'est pas le cas, on passe à la priorité numérotée 1 de l'extension *s* du contexte *[code\_invalide]*.

## Optimisation du routage avec les contextes

Les contextes permettent de définir des cadres pour des procédures plus évoluées, ciblées, et contrôlées. Lorsque le code d'un plan de numérotation est segmenté en contexte, il gagne en lisibilité, ce qui simplifie son suivi et sa mise à jour, au contraire d'un code compactant toutes les possibilités avant un seul contexte.

Les contextes sont au plan de numérotation ce que les fonctions sont à la programmation. Ils permettent de factoriser les actions à entreprendre, facilitant d'autant la réutilisation du code. Surtout, le code devenant modulaire, les traitements afférents à une extension particulière, à des catégories d'utilisateurs, à des services particuliers ou à des conditions spécifiques peuvent être regroupés dans une même portion de code, qu'il est plus simple de maintenir ultérieurement.

L'exemple de code suivant :

```
[internal]
exten => 801, 1, Answer()
exten => 801, 2, Echo()
exten => 801, 3, Playback(vm-goodbye)
exten => 801, 4, Hangup()
```

peut être remplacé par le suivant :

```
[internal]
exten => 801, 1, Goto (mon_echo, s, 1)

[mon_echo]
exten => s, 1, Answer ()
exten => s, n, Echo ()
exten => s, n, Playback (vm-goodbye)
exten => s. n, Hangup ()
```

Le second est certes plus long, mais il gagne en clarté, en particulier dans un plan de numérotation de plusieurs centaines de lignes. Il est en outre réutilisable pour d'autres appels.

## Et la vidéo ?

Avec Asterisk, les sessions peuvent gérer des communications à la fois audio et vidéo.

Pour cela, et en supposant que les intervenants effectuent une communication en utilisant le protocole de signalisation SIP, il n'y a qu'une manipulation mineure à faire dans la configuration d'Asterisk.

La modification est à réaliser dans le fichier de configuration **sip.conf** (qui se trouve, par défaut, dans le répertoire **/etc/asterisk**), en décommentant la ligne suivante (c'est-à-dire en retirant les points-virgules qui précèdent cette ligne) :

```
videosupport=yes
```

Dès lors, les intervenants peuvent utiliser le logiciel SIP de leur choix et initier des communications qui couplent la vidéo à la voix.

## Ajouter des sons

Lors de la mise en place d'un serveur vocal, il peut être nécessaire de diffuser à l'appelant un message audio, par exemple, pour indiquer à un utilisateur le nombre de messages déposés sur son répondeur, lui présenter une liste de choix, lui demander de saisir un code d'authentification, ou lui transmettre une annonce informative.

Asterisk offre divers fichiers audio en anglais, espagnol et français synthétisant différents messages génériques. Il est possible d'étendre ces fichiers en ajoutant des sons de son choix.

### Synthèse vocale

Le paquetage **asterisk-core-sounds-fr** (téléchargeable à l'adresse <ftp://ftp.digium.com/pub/telephony/sounds/>) propose plusieurs centaines de fichiers audio en français utilisables avec Asterisk.

Ces messages génériques de quelques secondes sont de bonne qualité. Différents formats de codage sont disponibles (G.711, G.722, G.729, gsm, wav). Le nom complet du fichier à télécharger comporte cette information à la suite du nom du paquetage. Par exemple, en téléchargeant le fichier **asterisk-core-sounds-fr-gsm-1.4.3.tar**, nous obtenons un paquetage au format GSM, dans sa version 1.4.3. Une fois décompressés à l'aide de la commande *tar*, les fichiers doivent être placés dans **/var/lib/asterisk/sounds**.

Pour jouer ces fichiers audio à un appelant, il suffit de lancer la commande *Playback* en mentionnant en argument le nom du fichier à jouer, sans l'extension.

En voici un exemple :

```
exten => 123, 1, Answer()  
exten => 123, 2, Playback (hello-word)  
exten => 123, 3, Hangup()
```

Si l'appelant compose le numéro d'appel *123*, le fichier audio **hello-word.gsm** est lu et le message audio « salut tout le monde » est diffusé. La communication est ensuite immédiatement coupée.

Un autre cas de figure concerne les nombres. Si le serveur vocal doit renvoyer le nombre de messages reçus ou un temps d'attente ou encore un numéro de téléphone, il n'est pas envisageable de faire figurer tous les nombres préenregistrés possibles. Une synthèse vocale est disponible par le biais des fonctions *SayDigits()* et *SayNumber()*. La première permet de prononcer et distinguer chaque chiffre entier composant un nombre. La seconde prononce le nombre directement (ce nombre doit être compris entre 0 et 99 999 999). Par exemple, la commande *SayDigits(123)* synthétise le message audio « un deux trois », tandis que la commande *SayNumber(123)* synthétise le message audio « cent vingt-trois ».

### Enregistrer ses propres messages

Asterisk offre la possibilité d'enrichir sa base de fichiers audio en l'utilisant comme un magnétophone. Cette fonction est très pratique pour réaliser des annonces spécifiques. Il suffit pour cela d'utiliser la commande *Record* prenant en paramètres un chemin (emplacement du fichier dans le système) associé à un format de fichier.

La syntaxe de cette commande est la suivante :

```
Record (chemin_d'enregistrement:format_de_sauvegarde|temps_son_blanc)
```

Le paramètre *chemin\_d'enregistrement* désigne l'emplacement dans l'arborescence du système où le son doit être enregistré. Pour être exploitable par la suite par son nom et pas par son emplacement complet, le fichier doit être placé dans le répertoire **sounds** d'Asterisk.

Le paramètre *format\_de\_sauvegarde* précise le codec utilisé. Le paramètre *temps\_son\_blanc* permet d'ajouter un délai additionnel de bruit blanc à la suite du message. L'ajout d'un bruit blanc à un message audio permet à l'appelant d'effectuer ses choix et saisies. Par exemple, si nous souhaitons inviter l'utilisateur à saisir une succession de touches au clavier, il faut lui laisser un temps ni trop court (pour qu'il ait le temps de saisir toutes les touches nécessaires), ni trop long (pour ne pas l'impatienter et lui redonner des directives s'il n'est pas suffisamment réactif).

Ce temps prédéfini est donné en secondes.

En voici un exemple :

```
exten => 456, 1, Answer ()
exten => 456, 2, Playback (conf-unmuted)
exten => 456, 3, Record (/tmp/rec:gsm|7)
exten => 456, 4, Playback (/tmp/rec)
exten => 456, 5, Playback (echo-done)
exten => 456, 6, Hangup ()
```

Lorsque l'utilisateur compose le numéro 456, un message audio lui joue le message « Vous pouvez parler maintenant ». L'enregistrement du son débute alors dans le fichier **/tmp/rec**, en utilisant le codec GSM. Sept secondes de son blanc y sont ajoutées. Ensuite, le message audio enregistré est rejoué à l'appelant en écho, suivi d'un message *echo-done*, qui indique que l'appel d'écho est terminé, avant de raccrocher.

Cet exemple constitue un bon test pour valider à n'importe quel moment le fonctionnement de son équipement audio. C'est du reste une fonction disponible dans des logiciels tels que Skype (en appelant l'identifiant *echo123*) ou Wengo (numéro d'appel 333).

### Utiliser d'autres sons

Il est possible d'utiliser n'importe quel son dans Asterisk, à la condition qu'il respecte les formats de codage supportés. Si ce n'est pas le cas, plusieurs programmes permettent de convertir différents formats non pris en charge par Asterisk en des formats supportés.

C'est le cas notamment du programme SOX, en licence GPL, téléchargeable à l'adresse <http://sox.sourceforge.net>. Après l'avoir installé, il suffit, par exemple, d'effectuer la conversion d'un fichier au format MP3, nommé **test.mp3**, en un fichier au format GSM nommé **test.gsm**, avec la commande suivante :

```
sox test.mp3 -r 8000 -c1 test.gsm resample -q1
```

## Problèmes éventuels avec les modules

Situés dans le dossier **/usr/lib/asterisk/modules**, les modules peuvent être paramétrés au moyen du fichier **/etc/asterisk/modules.conf**.

Par défaut, le paramètre *autoload=yes*, défini dans le fichier **modules.conf**, implique le chargement de tous les modules figurant dans le dossier **modules** d'Asterisk. Certains modules pouvant poser problème, il est possible de changer ce mode et de désactiver les modules au démarrage, en spécifiant la valeur *no* au paramètre *autoload*. Dans ce cas, seuls les modules explicitement mentionnés par la directive *load* sont chargés.

Inversement, il est possible d'activer tous les modules et d'interdire le lancement d'autres modules explicitement mentionnés. Le plus simple en ce cas est de les désactiver. Pour désactiver des modules en particulier, il suffit de conserver la valeur *yes* au paramètre *autoload* et de saisir à la section *[modules]* du fichier **modules.conf** la directive *noload* suivie du module à désactiver.

En voici un exemple :

```
[modules]
noload => cdr_pgsq1.so
noload => codec_lpc10.so
```

## Ajouter de nouveaux services

Notre plate-forme Asterisk étant désormais pleinement opérationnelle, nous pouvons mettre en relation tous les utilisateurs entre eux.

Cependant, aucun service évolué n'ayant encore été configuré, nous allons voir comment installer sur le serveur quelques services classiques.

### *Standard vocal automatique (IVR)*

Il est possible d'installer un IVR (Integrated Voice Responder), ou standard vocal automatique, permettant à l'appelant de sélectionner lui-même la personne ou le service avec lequel il souhaite entrer en communication.

Le standard automatique propose des menus à choix multiples conduisant à diverses actions spécifiques, telles que des annonces informatives, le relais vers un service approprié, l'accès à des services de type répondeur ou la mise en relation avec un poste téléphonique particulier.

L'exemple suivant de standard vocal automatique est un cas d'école, puisque incomplet. Si l'utilisateur compose la touche étoile, il est invité à saisir le numéro de poste téléphonique qu'il souhaite contacter. Ce numéro lui est répété avant la mise en communication. Les erreurs et attentes sont également traitées.

```
exten => *, 1, Goto (menu_choix, s, 1)

[menu_choix]
exten => s, 1, Background (enter-ext-of-person)

exten => 1, 1, Playback(you-entered)
exten => 1, 2, Playback(digits/1)
exten => 1, 3, Dial(SIP/guy)
exten => 1, 4, Hangup()

exten => 2, 1, Playback(you-entered)
exten => 2, 2, Playback(digits/2)
exten => 2, 3, Dial(SIP/laurent)
exten => 2, 4, Hangup()

exten => i, 1, Playback (pbx-invalid)
exten => i, 2, Goto (menu_choix, s, 1)

exten => t, 1, Hangup ()
```

La première ligne permet d'accéder au contexte `[menu_choix]` lors de la saisie de la touche étoile. Un message est alors diffusé pour demander à l'appelant de choisir le numéro de poste téléphonique qu'il souhaite joindre. Nous traitons dans cet exemple deux postes, numérotés 1 et 2.

Les possibilités sont les suivantes :

- L'appelant saisit la touche 1 (extension 1) : un message audio lui répète la touche qu'il vient de saisir, puis le système lance l'appel vers le poste numéro 1, qui est attribué à l'utilisateur *guy* (en signalisation SIP).
- L'appelant saisit la touche 2 (extension 2) : comme précédemment, après la confirmation de la saisie, la communication vers le poste 2 attribué à l'utilisateur *laurent* est initiée (en signalisation SIP).
- L'appelant saisit une autre touche que 1 ou 2 (extension *i*) : un message audio l'informe de l'invalidité de la saisie, puis l'étape précédente est immédiatement réactivée, l'invitant à saisir une nouvelle touche.
- L'appelant ne saisit aucune touche pendant un délai déterminé (extension *t*) : l'appel se termine aussitôt.

## Conférence

Deux étapes suffisent pour mettre en place une conférence avec Asterisk : créer les salons de conférence virtuelle et y inviter des participants.

### Créer des salons virtuels de conférences (fichier `meetme.conf`)

Pour créer des salons de conférences, il suffit de configurer le fichier `meetme.conf` en ajoutant à la section `[rooms]` le code suivant :

```
conf => numero_de_conference
                [ , code_accès_simple ]
                [ , code_accès_administrateur ]
```

Le mot-clé `Conf` correspond à une nouvelle salle de conférence virtuelle, définie au minimum par l'indication d'un numéro de salle (`numero_de_conference`). Il peut être complété optionnellement par un code d'accès que l'utilisateur devra fournir pour accéder à la salle virtuelle et éventuellement d'un code d'accès permettant de reconnaître l'administrateur, auquel des droits de gestion du salon virtuel sont attribués.

Par exemple :

```
conf => 770
```

permet de créer un salon ayant pour identifiant le numéro 770. Nous pouvons le compléter en remplaçant la ligne précédente par :

```
conf => 770, 12345, 150379
```

Cela crée un salon d'identifiant 770, auquel les utilisateurs peuvent accéder en indiquant le code 12345 et dont l'administrateur s'identifie par le code 150379.

### Inviter des participants à la conférence (application *Meetme*)

Pour inviter des participants à entrer dans la salle de conférence, il faut les aiguiller en utilisant le plan de numérotation et l'application *Meetme*.

Pour rediriger une communication vers la conférence précédente, Il suffit d'utiliser l'appel *Meetme* (770) dans le fichier **extensions.conf**.

Par exemple, si l'appelant compose le numéro 770, l'extension suivante l'invite à rejoindre la conférence 770 :

```
exten => 770, 1, Meetme (770)
```

Il est possible d'ajouter en second argument de l'application *Meetme* une ou plusieurs des options récapitulées au tableau 12.7 (s'il y en a plusieurs, les options sont indiquées en se succédant sans caractère de séparation).

**Tableau 12.7 – Options de l'application *Meetme***

Option	Description
m	Active le mode <i>monitor</i> : les participants peuvent écouter, mais pas parler.
p	Un participant peut quitter la conférence en pressant la touche dièse.
t	Active le mode <i>talk</i> : les participants peuvent parler, mais pas écouter.
v	Active le mode <i>video</i> .
q	Mode silencieux ( <i>quiet</i> ) : aucun son n'est émis lorsque des utilisateurs entrent dans la conférence ou en sortent.
d	Ajoute une conférence dynamiquement.
M	Active une musique d'attente lorsqu'il n'y a qu'un seul participant à la conférence.
b	Lance le script AGI spécifié dans la variable <i>MEETME_AGI_BACKGROUND</i> (celle-ci doit avoir été initialisée auparavant).

## Le service de messagerie audio (fichier *voicemail.conf*)

Le service de messagerie audio se met en place très simplement *via* la configuration de seulement trois fichiers : le premier permet de définir les paramètres du compte de messagerie vocale, le deuxième d'accéder à la boîte vocale créée, et le troisième de signaler à un utilisateur tout nouveau message vocal reçu.

Le principe du service de messagerie audio consiste à définir un numéro de boîte vocale associé à un utilisateur. Cela permet notamment à un même utilisateur d'avoir plusieurs numéros de téléphone (professionnel, personnel, autre) et une seule messagerie unifiée, archivant les messages qui lui sont destinés, que ces derniers soient personnels, professionnels ou autres.

Dans un cadre professionnel, il est aussi possible de permettre à plusieurs utilisateurs d'avoir un numéro de téléphone spécifique, tout en leur assignant une messagerie unifiée. Un message laissé sur le répondeur unique peut de la sorte être traité par la première personne disponible dans le service.

Les sections qui suivent montrent comment créer des boîtes vocales, les affecter à des utilisateurs et signaler la présence de nouveaux messages audio.

### Créer des comptes de messagerie audio (fichier *VoiceMail.conf*)

La première étape consiste à créer les boîtes vocales. Le fichier **VoiceMail.conf** comporte à cet effet deux sections distinctes : *[general]*, qui permet de paramétrer les options des comptes, et *[default]*, qui décrit les comptes à créer.

En voici un exemple :

```
[general]
format=gsm
attach=yes
fromstring=Ma Messagerie Vocale

[default]
3535 => 15155, guy_laurent, guy_laurent@fai_home.fr
```

La section *[général]* permet de définir différents paramètres de configuration, notamment les suivants :

- *format* : définit le format d'encodage des messages audio enregistrés (par défaut *wav49*, *gsm* et *wav*).
- *maxmessage* : spécifie la durée maximale (en secondes) d'un message. La valeur par défaut est 0, signifiant qu'il n'y a pas de durée maximale.
- *minmessage* : spécifie la durée minimale (en secondes) d'un message. La valeur par défaut est 0, signifiant qu'il n'y a pas de durée minimale.

- *maxmsg* : spécifie le nombre maximal de messages qui peuvent être laissés sur la boîte vocale. La valeur par défaut est *100*.
- *review* : autorise l'appelant à écouter et modifier le message laissé. La valeur par défaut est *no*.
- *attach* : indique si les messages vocaux laissés sur une messagerie sont également envoyés par e-mail aux utilisateurs concernés. La valeur par défaut est *no*.
- *maxlogins* : indique le nombre maximal de tentatives d'accès infructueuses autorisé.
- *emailsubject* : mentionne le sujet de l'e-mail notifiant le message vocal.
- *mailcmd* : indique chemin de la commande utilisée pour l'envoi des e-mails. La valeur par défaut est *usr/sbin/sendmail -t*.
- *fromstring* : spécifie un nom avec lequel l'e-mail notifiant le message vocal est envoyé. La valeur par défaut est *Asterisk PBX*.
- *emailbody* : indique le contenu de l'e-mail signalant la réception de nouveaux messages vocaux. Il est possible d'utiliser des variables dans la définition de ce paramètre.

Dans l'exemple suivant, le nom de l'utilisateur et le nombre de messages sont automatiquement remplacés par les variables correspondantes :

```
emailbody=Bonjour ${VM_NAME}, vous avez ${VM_MSGNUM} message(s) dans votre boîte vocale.
```

Dans notre exemple, après la section *[general]*, la section *[default]* spécifie les comptes à créer et leurs propriétés. Nous avons créé une seule boîte vocale identifiée par le numéro *3535*. Son mot de passe, *15155*, permet à son possesseur d'accéder à sa messagerie. Cette dernière est affectée à l'utilisateur *guy\_laurent* ayant pour adresse de messagerie *guy\_laurent@mon\_fai.fr* (c'est à cette adresse que seront envoyés les messages vocaux laissés sur la messagerie de l'utilisateur si le paramètre *attach* défini précédemment est activé à la valeur *yes*).

### Laisser un message et écouter ses messages (applications *VoiceMail* et *VoiceMailMain*)

Nous considérons ici deux cas de figure, selon qu'un utilisateur souhaite laisser un message sur la boîte vocale d'un autre utilisateur ou consulter ses propres messages sur sa propre boîte vocale.

#### Accéder à la boîte vocale d'un correspondant

Dans le premier cas, sans réponse de l'appelé au bout d'un certain temps, il faut enclencher une procédure effectuant le basculement de l'appel vers la messagerie du correspondant.

La fonction à utiliser pour cela est *VoiceMail*, qui prend comme argument le numéro de la boîte vocale, ainsi que, optionnellement, le contexte associé. Cet argument est donné sous la forme *numéro\_de\_boite\_vocale@contexte\_associé*, par exemple :

```
Voicemail(1000@default)
```

Dans l'exemple suivant inséré dans le plan de numérotation (fichier **extensions.conf**), l'appel est initié vers l'utilisateur *guy\_laurent* en composant le numéro *1235*. Au bout de 50 secondes sans réponse, la messagerie affectée à *guy\_laurent* et identifiée par le numéro de boîte vocale *3535* est enclenchée :

```
exten => 1235, 1, Dial (SIP/guy_laurent, 50, tm)
exten => 1235, 2, Voicemail (3535@default)
exten => 1235, 3, Hangup
```

La fonction *VoiceMail* prend en charge le service de répondeur en invitant l'appelant à parler et en enregistrant le message. Ce dernier est accessible par l'appelé en consultant sa boîte vocale.

### Consulter sa boîte vocale

Pour gérer la consultation des messages, il suffit de mettre en place un numéro d'appel spécialement dédié à la consultation de la messagerie. La fonction *VoiceMailMain* permet de lancer la messagerie souhaitée. Comme la fonction *VoiceMail*, elle prend pour argument le numéro de boîte vocale et optionnellement le contexte associé.

Dans l'exemple suivant, en composant l'extension *800*, l'appel est directement redirigé vers la consultation de la messagerie correspondant à la boîte vocale identifiée par le numéro *3535* (correspondant au compte de messagerie vocale de l'utilisateur *guy\_laurent*) :

```
exten => 800, 1, Answer
exten => 800, 2, VoiceMailMain(3535@default)
exten => 800, 3, Hangup
```

La fonction *VoiceMailMain* prend en charge le service de consultation en demandant notamment le mot de passe de l'utilisateur, puis en lançant la lecture, la suppression et plus généralement la gestion des messages audio.

Cette implémentation peut être optimisée. Par exemple, au lieu d'affecter le numéro d'appel *800* exclusivement à la boîte vocale de l'utilisateur *guy\_laurent*, il serait préférable, comme cela se fait couramment en téléphonie classique, de disposer d'un numéro d'appel unique pour gérer la messagerie de tous les abonnés et d'utiliser une identification et authentification pour accéder à la boîte vocale désirée.

### Signaler la présence d'un nouveau message audio

Pour signaler aux utilisateurs la présence d'un nouveau message audio dans leur boîte vocale, il suffit d'associer cette dernière au compte d'un utilisateur. Par exemple, si un utilisateur utilise le protocole de signalisation SIP, son compte est créé dans le fichier **sip.conf**, et il suffit d'ajouter à la section `[guy_laurent]` de ce compte la ligne suivante :

```
mailbox=3535@default
```

## Aller plus loin avec Asterisk

Les possibilités d'exploitation du serveur Asterisk sont telles que le logiciel n'a pas à rougir de la comparaison avec ses équivalents PBX traditionnels, souvent hors de prix et de portée des particuliers.

Dans ce chapitre, nous n'avons évoqué qu'un fonctionnement standard, illustrant les usages les plus courants, mais quantité d'autres possibilités sont envisageables. On pourrait presque dire que tout ce qu'un PBX physique traditionnel sait faire, Asterisk peut le faire aussi.

Nous mentionnons dans cette section quelques autres fonctionnalités parmi les plus remarquables offertes par Asterisk.

### Connecter Asterisk à un fournisseur SIP

Il existe de nombreux acteurs qui proposent à leurs clients de communiquer en utilisant le protocole de signalisation SIP. C'est le cas, par exemple des fournisseurs d'accès français Free et Neuf Télécom, mais aussi de la société Wengo avec son logiciel WengoPhone dont nous avons déjà parlé dans un précédent chapitre, ou encore de VoIPDiscount, également évoqué précédemment. Bien souvent, ces comptes sont associés à des conditions tarifaires très avantageuses, notamment la gratuité des appels dans plusieurs dizaines de pays. Pourquoi ne pas faire profiter Asterisk de ce compte ?

L'idée ici serait alors de disposer d'un compte SIP que nous procure l'une de ses sociétés, et de configurer Asterisk avec celui-ci. De cette manière tous les téléphones reliés au serveur Asterisk pourront bénéficier des mêmes conditions tarifaire de leur compte SIP. Les avantages sont multiples :

- Tous les téléphones connectés à Asterisk peuvent tirer profit du compte SIP, même s'ils ne sont pas compatibles SIP, puisque Asterisk sert de passerelle.

- Les services activés sur Asterisk restent disponibles dans le cadre des communications effectuées *via* le compte SIP (journalisation des appels, enregistrement du carnet d'adresses, etc.).
- Les utilisateurs connectés n'ont pas à configurer leur logiciel avec le compte SIP (ils n'ont même pas besoin de le connaître).
- Si le fournisseur SIP propose un numéro d'appel entrant, il n'est pas nécessaire d'avoir un logiciel SIP spécifique qui soit actif, ni même un téléphone de VoIP compatible SIP pour recevoir les communications : Asterisk pourra être configuré pour recevoir tous les appels entrants vers ce numéro d'appel et les rediriger vers n'importe quel téléphone.

On considère dans la suite qu'on dispose d'un serveur Asterisk fonctionnel et d'un compte SIP (identifiant, mot de passe, adresse du serveur SIP à contacter et éventuellement numéro de téléphone si le fournisseur SIP le propose). On ne doit s'assurer que l'on dispose des paramètres SIP correctes (souvent, l'utilisateur dispose d'un compte auprès du fournisseur, qui n'a rien à voir avec son compte SIP, mais sert, par exemple, à la facturation de son compte sur Internet ou à d'autres fonctionnalités : c'est notamment le cas avec le logiciel WengoPhone ou les utilisateurs se connectent au logiciel avec des paramètres qui ne sont pas des paramètres SIP).

Nous allons mettre en œuvre cette fonctionnalité, en réalisant plusieurs étapes : d'abord on configurera le serveur Asterisk comme étant un client SIP, puis on déclarera le compte SIP dont on dispose et enfin on redirigera les appels sortants vers le fournisseur SIP, et les appels entrants vers le téléphone de son choix.

Pour réaliser la première étape, on ajoute une ligne dans le fichier `sip.conf`, à la section `[general]`. Cette ligne va permettre de s'enregistrer auprès du fournisseur SIP comme étant l'utilisateur, et comme étant actif dans le réseau (donc pouvant appeler et recevoir des communications). Elle doit avoir le format suivant :

```
register => identifiant[:mot_de_passe]@adresse_fournisseur_sip[:port][/param]
```

L'extension à saisir pour faire d'Asterisk un client est *register*. On lui associe trois informations : le nom de l'utilisateur (*identifiant*), son mot de passe presque toujours obligatoire (*mot\_de\_passe*), et on mentionne le nom (ou l'adresse IP) du serveur SIP que le fournisseur SIP doit fournir (*adresse\_fournisseur\_sip*). Éventuellement on peut préciser un port spécifique si le fournisseur SIP l'exige (*port*), et un paramètre que le fournisseur SIP pourra interpréter.

Un exemple d'utilisation est donné dans ce qui suit :

```
register => david:d@v1d@sip.voipdiscount.com
```

Dans un second temps, et toujours dans le fichier `sip.conf`, après la section `[global]`, on déclare une section qui définira les propriétés du compte SIP :

```
[nom_fournisseur_sip]
type=peer
username=identifiant
secret= mot_de_passe
fromuser=identifiant
callerid=identifiant <numero_tel>
fromdomain=adresse_fournisseur_sip
host=adresse_fournisseur_sip
nat=yes
canreinvite=no
context=appels_entrants
```

On retrouve essentiellement des mêmes éléments déjà présentés, et qu'il faut remplacer par les paramètres du compte (ces paramètres sont en italiques), avec, optionnellement, un numéro de téléphone (*numero\_tel*) si le fournisseur SIP le propose. Les paramètres *nat* et *canreinvite* sont également à adapter (selon qu'on se trouve effectivement derrière un nat ou non). L'intitulé de la section est quelconque (ici *nom\_fournisseur\_sip*). Le contexte nommé *appels\_entrants* sera défini plus loin.

Puis, dans la section `[internal]` du fichier `extensions.conf`, on ajoute une ligne pour rediriger les appels sortants (donc que les utilisateurs d'Asterisk émettent) vers le serveur SIP du fournisseur :

```
exten => _0., 1, Dial(SIP/${EXTEN}@adresse_fournisseur_sip)
```

L'identifiant d'extension « *\_0.* » permet de filtrer tous les appels dont le numéro commence par le chiffre 0. On remarque l'utilisation de la variable spéciale `EXTEN`, qui correspond au numéro de téléphone saisi par l'appelant.

Enfin, et optionnellement, si le fournisseur SIP propose à ses clients un numéro de téléphone, alors toujours dans le fichier `extensions.conf`, on ajoute une section pour rediriger les appels entrants sur ce numéro (qu'on notera comme précédemment *numero\_tel*) vers le compte de son choix (ici le compte *nom\_compte*, à remplacer par *nom\_compte*) :

```
[appels_entrants]
exten => numero_tel, 1, Answer ()
exten => numero_tel, 2, Dial(SIP/nom_compte)
exten => numero_tel, 3, Hangup ()
```

Dans ce cas, lorsque des utilisateurs cherchent à joindre le numéro *numero\_tel*, l'appel est automatiquement redirigé vers le compte *nom\_compte* en utilisant le protocole de signalisation SIP (qui doit être défini dans le fichier *sip.conf*).

Bien sûr, cette possibilité est à restreindre au seul cercle privé. Elle entrerait en violation avec les conditions d'utilisation du contrat si elle était exploitée à des fins mercantiles, ou tout simplement en dehors du cercle privé.

## AGI (Asterisk Gateway Interface)

Asterisk est un logiciel totalement ouvert, à la fois par ses sources, qui sont disponibles en téléchargement et que les programmeurs peuvent enrichir et personnaliser au sein de la communauté, et par le développement de logiciels tiers et l'interaction avec eux.

Les développeurs d'Asterisk facilitent le travail des autres programmeurs en leur proposant une interface générique de contrôle et de gestion du serveur Asterisk, appelée AGI (Asterisk Gateway Interface). N'importe qui peut donc développer une application, dans le langage de programmation de son choix, et la personnaliser à son grès afin d'interagir avec le serveur Asterisk.

Peu de constructeurs offrent ce degré d'ouverture et de compatibilité. Le plus souvent, les utilisateurs doivent se contenter de ce que leur proposent leurs équipementiers, car l'implémentation logicielle, en plus d'être fermée, est protégée dans son code source et n'offre généralement aucun système d'interface comparable à l'AGI.

## Trixbox

Trixbox (anciennement Asterisk@home) est une distribution complète et libre du serveur Asterisk qui a la particularité d'être accessible à partir d'un CD bootable. Elle peut être téléchargée sur la page de l'éditeur, à l'adresse <http://www.trixbox.org>.

Actuellement disponible uniquement en version anglaise, ce logiciel permet de se faire une idée d'Asterisk avant de l'adopter et sans se lancer dans des procédures d'installation et de configuration.

Par défaut, tout est prévu pour fonctionner au lancement. Il est toutefois possible de modifier les configurations de base en suivant les indications fournies dans ce chapitre. Pour cela, il suffit d'arrêter le serveur Asterisk puis de remplacer le contenu des fichiers du répertoire **/etc/asterisk** par ceux indiqués aux sections précédentes de ce chapitre.

Certains modules peuvent être désactivés s'ils sont inutiles ou qu'ils posent problème (option *noload* dans le fichier **modules.conf**).

## Communiquer avec le protocole IAX

Le projet Asterisk a donné naissance à un second projet, appelé IAX (Inter Asterisk eXchange). Celui-ci définit un protocole permettant l'interconnexion entre serveurs Asterisk, mais également la communication entre un client et un serveur Asterisk.

Initialement, le protocole IAX a été développé par le concepteur d'Asterisk, Mark Spencer, de la société Digium. Il est aujourd'hui maintenu par la société Digium et est disponible dans sa deuxième version IAX2, laquelle fait l'objet d'une proposition de normalisation à l'IETF.

Pour être convaincante dans un contexte où la concurrence entre les protocoles H.323 et SIP est déjà importante, la philosophie proposée par IAX diffère sur deux points importants :

- Traversée transparente des passerelles NAT et des pare-feu. Contrairement aux protocoles SIP et H.323, qui n'assurent que la fonction de signalisation et se combinent généralement à RTP pour la fonctionnalité de transport des flux, le protocole IAX est à la fois un protocole de transport et un protocole de signalisation. Cela lui permet plus facilement de traverser les pare-feu et de supporter les translations d'adresses IP (NAT) dans un réseau. Ses flux n'utilisent qu'un port fixe et unique (le port 4569) et peuvent de la sorte être aisément identifiés.
- Utilisation réduite de la bande passante. Si H.323 et SIP sont prévus pour le multimédia en général, IAX a été conçu spécifiquement pour le problème du transport et de la signalisation de la voix, en écartant les considérations plus générales des applications multimédias. Le protocole IAX répond ainsi à des objectifs simples et bien délimités. Bien qu'il n'exclue pas *a priori* le traitement de flux vidéo, il s'intéresse avant tout aux flux audio et optimise les paramétrages des flux en tenant compte des contraintes et des spécificités de ces flux audio.

IAX est un protocole puissant, qui propose des solutions efficaces aux deux problèmes importants rencontrés par H.323 et SIP et permet les communications entre serveurs Asterisk. Il souffre toutefois de l'inconvénient de ne pas être normalisé. De plus, il n'optimise que le traitement des flux téléphoniques, alors que H.323 comme SIP sont plus généralistes et peuvent s'appliquer au transfert de la vidéo.

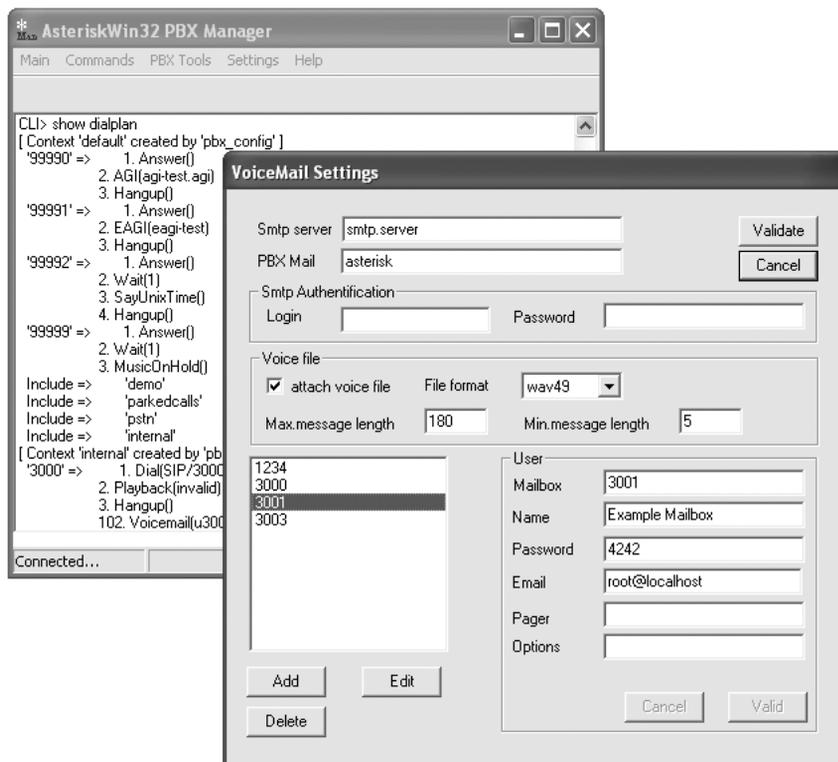
## Asterisk sous Windows

Une version du serveur Asterisk a été développée sous plate-forme Windows. Nommée *asteriskwin32*, elle est téléchargeable à l'adresse [www.asteriskwin32.com](http://www.asteriskwin32.com). Bien qu'elle soit beaucoup moins performante que le logiciel original sous Linux, cette solution constitue un excellent moyen de se familiariser avec les concepts généraux d'Asterisk.

L'utilisateur dispose d'une interface conviviale lui permettant d'accéder à toutes les fonctionnalités du logiciel. On y trouve notamment une préconfiguration initiale du plan de numérotation, qui peut être visualisée en sélectionnant la fonction Dialplan du menu Commands. Comme sous Linux, la modification s'effectue en éditant le fichier **extension.conf** placé dans le répertoire **/etc/** du répertoire spécifié lors de l'installation du logiciel.

De même, des comptes d'utilisateurs sont déjà créés (SIP et IAX uniquement), et les principaux services sont préconfigurés. Pour personnaliser ces services, nul besoin d'aller modifier des fichiers. Il suffit d'utiliser l'interface, comme l'illustre la figure 12.5 représentant la configuration du service de messagerie téléphonique.

**Figure 12.5**  
*Configuration du service de messagerie audio sous Windows*



## La concurrence

Si Asterisk est le plus connu des PBX logiciel, il n'est pas le seul. Deux autres logiciels, également libres, Vocal et SIP-X, ont une vocation semblable à celle d'Asterisk.

### Vocal

Développé en 1999 par la société Vovida, Vocal (Vovida Open Communication Application Library) est soutenu par d'importants industriels tels que Cisco, qui en a fait l'acquisition en novembre 2000.

Solution de téléphonie complète et multiplate-forme (BSD, Linux, Windows, Solaris), Vocal inclut des interfaces d'administration du système, un serveur de politiques exploitant le protocole COPS (Common Open Policy Service) et un large choix de fonctionnalités téléphoniques.

La communauté Vocal demeure cependant nettement moins active que celle d'Asterisk. Le produit lui-même est moins abouti en termes d'intégration avec les codecs et protocoles existants.

Le site de l'éditeur (<http://www.vovida.org>) permet d'obtenir davantage d'information sur ce PBX-IP.

### SIP-X

Parrainé par la société Pingtel, SIP-X est le dernier-né des autocommutateurs libres. À la manière de Digium pour Asterisk, Pingtel assure les services de maintenance et de mise à jour du logiciel. Plus généralement, les développeurs sont regroupés au sein du groupe SIPFoundry.

Pour fonctionner, SIP-X requiert l'utilisation de logiciels clients ou de terminaux compatibles SIP exclusivement. Il n'est donc pas aussi large d'utilisation qu'Asterisk.

Plus d'informations peuvent être trouvées sur le site de l'éditeur, à l'adresse <http://www.sipfoundry.org/sipX/index.htm>.

## Conclusion

Ouvert à tous, gratuit, simple à utiliser, puissant et performant, Asterisk a vraiment de quoi séduire, et même rivaliser avec les équipements professionnels. En fait, les vrais concurrents d'Asterisk ne sont pas les autres PBX logiciels, mais les PBX hardware eux-mêmes.

En effet, si les PBX hardware sont chers, ils demeurent performants et fiables. Surtout qu'ils disposent généralement d'un support technique appréciable. Au moindre problème, un technicien peut intervenir dans des délais très courts, ce qui rassure évidemment les entreprises, lesquelles ne sont pas toujours prêtes à faire des économies s'il faut, pour cela, tenter le pari d'une solution ouverte. Ces solutions libres, en effet, peuvent fournir des outils performants et très bien documentés, mais généralement sans procurer la garantie d'un service après-vente, pourtant rassurante lorsqu'on exploite une infrastructure destinée à la téléphonie sur IP.

Il faudra donc certainement encore du temps avant qu'Asterisk gagne le même statut que ces concurrents et dispose d'un capital de confiance aussi fort chez les professionnels. De nombreuses sociétés œuvrent cependant dans ce sens et proposent d'ores et déjà un service de bout en bout assurant la fourniture matérielle et logicielle, ainsi que l'installation et la maintenance du logiciel.

Dans un secteur en pleine mutation, où le monde RTC s'efface au fur et à mesure que le monde IP prend sa place, Asterisk influence d'ores et déjà les stratégies des équipementiers en montrant la voie.

# 13

## La téléphonie chez les fournisseurs d'accès

---

Le haut débit à l'accès est devenu une nécessité dans un monde où la quantité et la qualité des informations à transporter augmentent sans discontinuer. Un débit de l'ordre du mégabit par seconde semble être une valeur minimale pour réaliser des accès dits à haut débit. Avec l'arrivée de la télévision et de sa version haute définition associée à plusieurs chaînes de télévision simultanées, il faut pouvoir compter aujourd'hui sur une cinquantaine de mégabits par seconde.

Ce chapitre s'intéresse aux accès haut débit terrestres pour les particuliers et les petites et moyennes entreprises, et plus précisément à l'intégration de la parole téléphonique dans ces environnements. Ces accès comprennent quatre types : la ligne téléphonique par le biais d'un modem xDSL, le câble CATV associé à un modem câble, la fibre optique et l'accès Wi-Fi en Quadruple-Play.

### Les accès xDSL

Les modems xDSL permettent d'utiliser les paires métalliques du réseau d'accès pour réaliser une boucle locale à haut débit. Le débit dépend fortement de la qualité du câble utilisé et de la distance à parcourir. Plusieurs catégories de modems xDSL sont commercialisées, la lettre *x* permettant de les différencier.

Les modems ADSL (Asymmetric Digital Subscriber Line) sont les plus répandus. Leurs vitesses sont dissymétriques, plus lentes entre le terminal et le réseau que dans l'autre sens. En règle générale, le sens montant est quatre fois moins rapide que le sens descendant. Les vitesses sur le sens descendant peuvent atteindre 2 Mbit/s pour une distance de

l'ordre de 5 km et dépasser la vingtaine de mégabits par seconde lorsqu'on est à moins d'un kilomètre de l'équipement de l'opérateur.

Le modem ADSL utilise une modulation d'amplitude quadratique, c'est-à-dire que 16 bits sont transportés à chaque signal. Avec une rapidité de modulation de 340 kilobauds et une atténuation de l'ordre d'une trentaine de décibels, on atteint plus de 5 Mbit/s.

Devant le succès rencontré par la technique ADSL, des dérivés en ont été proposés, notamment la technique consistant à faire varier le débit sur le câble, qui a donné naissance au RADSL (Rate Adaptive DSL). Pour les hauts débits, les solutions HDSL (High bit rate DSL) et VDSL (Very high bit rate DSL) peuvent être exploitées avec succès si le câblage, souvent en fibre optique, le permet.

Les mesures effectuées chez les opérateurs montrent que les débits deviennent de plus en plus symétriques depuis l'apparition des applications peer-to-peer (P2P), les stations des utilisateurs client devenant des serveurs. Les techniques SDSL (Symmetric DSL) vont donc devenir de plus en plus courantes chez les particuliers. Elles sont aujourd'hui réservées aux entreprises jusqu'à des valeurs de 8 Mbit/s.

## Le modem xDSL

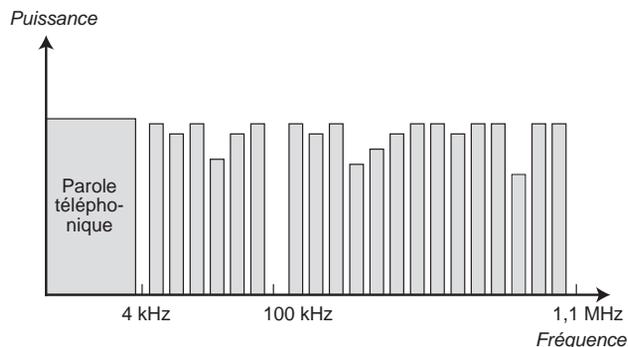
Deux techniques sont utilisées pour augmenter le débit sur une communication xDSL : le full-duplex, qui est assuré sur une même paire grâce à l'annulation d'écho, et l'utilisation d'un code spécifique (2B1Q).

Les modems ADSL possèdent une bande montante de 4 à 100 kHz, qui est utilisée pour des débits de 0,64 Mbit/s. La bande descendante est comprise entre 100 kHz et 1,1 MHz, qui permet d'atteindre le débit de 8,2 Mbit/s. La parole analogique, entre 0 et 4 kHz, passe en parallèle des données utilisant le modem.

Les codes en ligne des modems ADSL reposent soit sur la modulation CAP (Carrierless Amplitude and Phase), soit sur la norme DMT (Discrete MultiTone), de l'ANSI (American National Standards Institute) et de l'ETSI. La méthode DMT consiste en l'utilisation de 256 canaux de 4 kHz, chaque canal permettant l'émission de 15 bits par hertz au maximum.

La figure 13.1 illustre la partie du spectre utilisée par les modems ADSL.

**Figure 13.1**  
*Partie du spectre utilisée  
par l'ADSL*



Le spectre est découpé en trois parties, une entre 0 et 4 kHz pour faire passer la parole téléphonique, qui continue à être acheminée en parallèle des données, une entre 4 et 100 MHz pour la voie montante allant du terminal vers le réseau et une entre 100 kHz et 1,1 MHz pour la voie descendante allant du réseau au terminal.

Des variantes peuvent exister en découpant la bande de façon différente, par exemple entre 4 et 250 MHz pour la voie montante et entre 250 MHz et 1,1 MHz pour la voie descendante. La génération de modems ADSL2+ utilise une bande passante encore plus importante, permettant d'atteindre la fréquence de 2,2 MHz et des débits associés de 25 Mbit/s dans le sens descendant et 1,2 Mbit/s dans le sens montant.

La partie montante du spectre de l'ADSL de base est divisée en 20 sous-bandes de 4,3 kHz. Chaque sous-bande est capable de transporter de 4 à 15 bits en parallèle. En choisissant 8 bits par intervalle d'horloge, avec 4 000 intervalles de temps par seconde, le modem ADSL permet de transporter :

$$4\,000 \times 8 \text{ bits} = 32 \text{ Kbit/s par sous-bande}$$

Comme il y a 20 sous-bandes, on arrive au total de  $32 \times 20 = 640$  Kbit/s.

La partie montante de la communication est découpée en 256 tranches de 4,3 kHz. Toujours pour un transport de 8 bits par intervalle de temps, on arrive au débit de :

$$4\,000 \times 8 \text{ bits} \times 256 = 8,2 \text{ Mbit/s}$$

Il est possible d'améliorer le débit en augmentant le nombre de bits par intervalle de temps.

Des versions simplifiées de modems ADSL sont parfois mises en œuvre dans certains pays, telles que l'ADSL Lite, ou G-Lite, et U-ADSL (Universal ADSL). L'objectif de cette simplification est d'offrir un accès à Internet à très bas prix. Les capacités de transmission sont respectivement de 1,5 Mbit/s et 512 Kbit/s. Des cartes ADSL Lite sont commercialisées pour les PC.

Les modems G-Lite ressemblent aux modems ADSL, mais ils sont capables de s'adapter aux possibilités de la ligne. Le modem G-Lite ne se place pas à côté de la communication téléphonique, comme dans l'ADSL, mais prend toute la capacité de la ligne. En particulier, le modem s'interrompt si une communication téléphonique doit passer par la ligne. Les modems G-Lite s'adaptent bien aux accès haut débit, en particulier pour l'ATM. Dans ce cas, le protocole PPP (Point-to-Point Protocol) peut être utilisé. Il a été standardisé dans cette configuration par l'ADSL Forum et l'ANSI. Nous revenons sur les protocoles utilisés par les modems ultérieurement dans ce chapitre.

L'ADSL Forum a défini l'interface à respecter. Cette dernière a commencé par suivre l'architecture ATM, déployée par les opérateurs et les équipementiers du secteur des télécommunications vers le début des années 90. À cette époque, l'ATM représentait une potentialité forte pour l'unification des réseaux des années 2000. Depuis quelques années, la technologie Ethernet prend le dessus, et la plupart des modems ADSL sont aujourd'hui des modems Ethernet.

Les octets provenant des différentes sous-bandes sont encapsulés dans des trames ATM ou Ethernet. Les trames ATM ou Ethernet sont elles-mêmes encapsulées dans une trame de niveau physique, ou supertrame, qui correspond à la vitesse de l'accès ADSL. Par exemple, pour une connexion d'une capacité utile de 1,5 Mbit/s, les trames ATM sont transmises dans une supertrame de 68 cellules ATM, plus une cellule de synchronisation. Chaque supertrame demande un temps de 17 ms pour être émise, ce qui correspond à un peu moins de 250  $\mu$ s par trame. La vitesse de transmission utile pour le client atteint dans ce cas 1,5 Mbit/s, une fois enlevées les synchronisations et les bits de redondance ou de correction d'erreur.

### *Ethernet dans le premier mile*

L'IEEE a promulgué un standard pour l'utilisation d'Ethernet sur la boucle locale. Appelée EFM (Ethernet-in-the-First-Mile), cette solution au débit symétrique de 10 Mbit/s devrait permettre aux entreprises de se connecter à Internet en mode symétrique à haut débit. L'avantage est d'utiliser les paires torsadées téléphoniques, deux paires en l'occurrence, et donc de permettre à un coût faible une connexion à un débit acceptable. Cette solution correspond à la norme IEEE 802.ah. Elle est limitée à une distance de 750 m à 10 Mbit/s et de 2 700 m à 2 Mbit/s.

Une extension à 50 Mbit/s est en cours de normalisation avec 8 paires de fils métalliques torsadés sur une distance de l'ordre du kilomètre. Pour dépasser cette distance, il faut faire appel à de la fibre optique, qui permet en EFMF (EFM Fiber), avec une fibre en monomode, d'atteindre des vitesses de 100 ou 1 000 Mbit/s jusqu'à des distances de l'ordre de 10 km. Comme nous le verrons, les techniques PON (Passive Optical Network) permettent d'allonger encore la distance à 20 km à des vitesses de 1 et bientôt 2,5 Gbit/s.

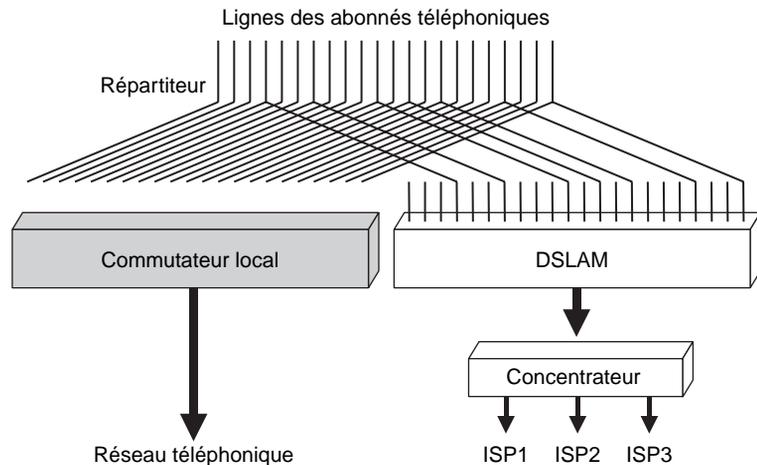
Les DSLAM forment l'autre extrémité de la liaison, chez l'opérateur. Le rôle de ces équipements est de récupérer les données émises par l'utilisateur depuis son équipement terminal au travers de son modem ADSL. Ces équipements intègrent des modems situés à la frontière de la boucle locale et du réseau de l'opérateur.

La figure 13.2 illustre le positionnement d'un DSLAM.

Les lignes des abonnés à l'opérateur local arrivent sur un répartiteur, qui permet de connecter l'utilisateur au commutateur téléphonique et au DSLAM s'il a un abonnement DSL. Le DSLAM est lui-même connecté à un concentrateur, que nous présentons un peu plus loin du point de vue protocolaire. Ce cas de figure est celui de l'opérateur historique.

Le dégroupage désigne l'arrivée d'opérateurs alternatifs pour offrir des services téléphoniques de données à haut débit et de vidéo, comme la télévision. Parmi les diverses possibilités de réalisation pratique du dégroupage, la pose de câbles a été envisagée pour réaliser une boucle locale différente de celle de l'opérateur historique, lequel en possédait, jusqu'à la fin des années 90, le contrôle total.

Figure 13.2

*Positionnement d'un DSLAM*

En raison du prix très élevé de la pose d'un réseau d'accès et de l'aberration que représenterait l'arrivée de plusieurs boucles locales jusque chez l'utilisateur, une par opérateur, d'autres solutions ont été adoptées. Certains opérateurs ont choisi de se positionner au niveau du répartiteur. À partir de ce répartiteur, ils ont installé leurs propres connexions et leur propre DSLAM.

L'inconvénient de cette solution provient de la situation géographique du DSLAM de l'opérateur alternatif, qui doit se trouver dans une salle proche de celle de l'opérateur historique. De plus, l'opérateur alternatif doit tirer une liaison vers son propre réseau sans connaître avec précision le nombre d'utilisateurs qui le choisiront.

Une autre possibilité consiste à se positionner derrière le DSLAM en ayant son propre concentrateur ou bien, comme sur la figure, en se connectant à la sortie du concentrateur. L'opérateur qui prend en charge la connexion entre le modem de l'utilisateur et la sortie du concentrateur s'appelle le fournisseur d'accès, ou NAP (Network Access Provider).

Une dernière solution consiste à utiliser le réseau de France Télécom pour atteindre un POP (Point of Presence) de l'opérateur alternatif.

Jusqu'en 2004, les utilisateurs étaient obligés d'avoir un abonnement à France Télécom pour transmettre sur la boucle locale entre l'équipement terminal et le DSLAM. Désormais, le dégroupage est total, et la facture de la communication sur la boucle locale est gérée par l'opérateur alternatif, qui doit malgré tout louer la ligne de la boucle locale à France Télécom à un coût décidé par le régulateur.

## Les protocoles de l'ADSL

L'utilisateur générant des paquets IP, il faut pouvoir transporter ces derniers vers le modem ADSL. Pour cela, on utilise soit une trame Ethernet, soit une trame PPP, soit une trame USB, soit une superposition de ces trames, comme une trame PPP encapsulée dans une trame Ethernet ou une trame PPP encapsulée dans une trame USB.

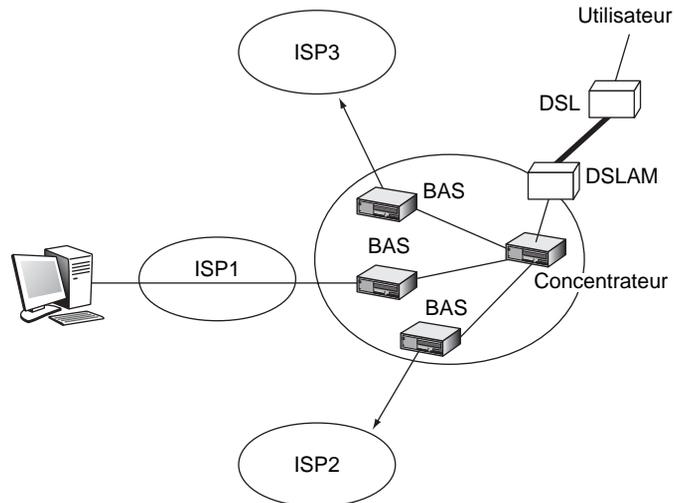
Prenons l'exemple de paquets IP encapsulés dans une trame Ethernet. Cette trame est envoyée soit sur un réseau Ethernet reliant le PC du client au modem, soit dans une trame PPP sur une interface de type USB. Dans le modem ADSL, il faut décapsuler la trame pour récupérer le paquet IP puis l'encapsuler de nouveau, mais cette fois dans une trame ATM. Cette fragmentation en morceaux de 48 octets est réalisée par le biais d'une couche AAL5 (ATM Adaptation Layer de type 5).

Une fois la trame ATM arrivée dans le DSLAM, plusieurs cas de figure peuvent se présenter suivant l'architecture du réseau du FAI auquel le client est connecté. Une solution consiste à décapsuler les cellules ATM et à récupérer le paquet IP qui est transmis vers le concentrateur dans une trame Ethernet. Le concentrateur l'envoie vers le FAI également dans une trame Ethernet. Une autre solution consiste à laisser les trames sous forme ATM. C'est le cas lorsque l'opérateur de la boucle locale et le FAI utilisent la même technologie. Dans ce cas, la cellule ATM est directement envoyée vers le concentrateur, qui joue le rôle de commutateur ATM. Celui-ci envoie les trames ATM par des circuits virtuels vers des BAS (Broadband Access Server), qui sont les équipements intermédiaires permettant l'accès vers les réseaux des FAI alternatifs.

Ces topologies sont illustrées à la figure 13.3.

Figure 13.3

*Équipements de concentration entre l'utilisateur et le serveur*



Une autre solution, qui est aussi très utilisée, consiste à placer le paquet IP de départ dans une trame PPP et à garder cette trame tout le long du chemin, quitte à l'encapsuler dans d'autres trames. Cela a donné naissance au protocole PPPoE (PPP over Ethernet) dans le cas où la trame PPP est émise sur Ethernet. La trame PPP peut être encapsulée dans plusieurs trames ATM après avoir été découpée en morceaux de 48 octets par le biais du protocole AAL5.

L'avantage de conserver la trame PPP tout le long du chemin est de pouvoir l'encapsuler dans un tunnel L2TP.

## Le protocole L2TP

Pour réaliser les communications entre les BAS et les serveurs, un protocole de tunneling doit être mis en place puisque ce chemin peut être considéré comme celui à prendre par tous les paquets ou trames provenant des différents DSLAM et allant vers le même serveur.

Le tunneling est une technique courante, qui ressemble à un circuit virtuel. Les trois protocoles utilisés pour cela sont PPTP (Point-to-Point Tunneling Protocol), L2F (Layer 2 Forwarding) et L2TP (Layer 2 Tunneling Protocol). Ces protocoles permettent l'authentification de l'utilisateur, l'affectation dynamique d'adresse, le chiffrement des données et éventuellement leur compression.

Le protocole le plus récent, L2TP, supporte difficilement le passage à l'échelle, ou scalabilité, et n'arrive pas à traiter correctement et suffisamment vite un nombre de flots dépassant les valeurs moyennes. Dans ce cas, on ajoute des concentrateurs d'accès L2TP, ou LAC (L2TP Access Concentrator), qui récupèrent tous les clients provenant d'un même DSLAM et allant vers un même BAS et les multiplexent sur un même circuit virtuel.

La figure 13.4 illustre l'architecture protocolaire d'une communication d'un PC vers un serveur situé dans un réseau de FAI différent de celui de l'opérateur d'entrée. Le PC travaille sous TCP/IP et est connecté à un modem ADSL par le biais d'un réseau Ethernet.

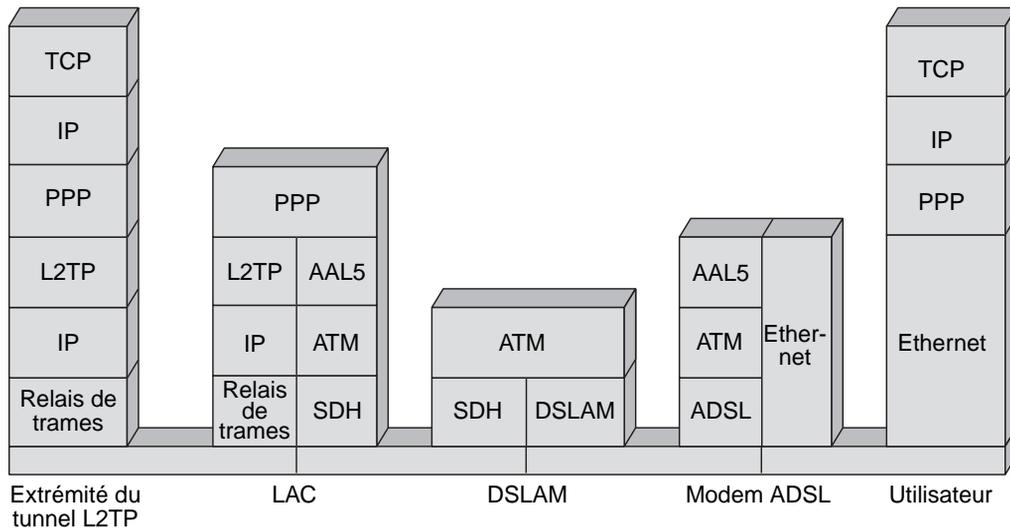


Figure 13.4

Architecture protocolaire d'une communication ADSL

## Les modems VDSL

Les modems VDSL (Very high bit rate DSL) permettent d'atteindre des vitesses beaucoup plus élevées que les modems ADSL, mais sur quelques centaines de mètres seulement. Leur capacité est de plusieurs dizaines de mégabits par seconde. Les modems VDSL peuvent se mettre à la sortie d'un PON (Passive Optical Network), que nous décrivons un peu plus loin, pour prolonger leur liaison vers l'utilisateur. Les PON pouvant être en technologie ATM. Les modems VDSL doivent en ce cas accepter la trame ATM. Aujourd'hui, on commence toutefois à développer des E-PON (Ethernet PON).

Les débits peuvent être asymétriques ou symétriques, au choix de l'utilisateur. Selon les propositions de l'ANSI, les débits en asymétrie atteignent 6,4 Mbit/s dans le sens montant et 52 Mbit/s dans le sens descendant sur une distance de 300 m. Pour une distance de 1 000 m, il devrait être possible d'obtenir la moitié des débits précédents. La bande de fréquences située entre 300 et 700 kHz est dévolue à la bande montante. La partie du spectre située entre 700 kHz et 30 MHz sert à la bande descendante. La partie basse du spectre est réservée à la parole téléphonique classique de l'opérateur France Télécom.

Comme dans le cas de l'ADSL, un filtre permet de séparer la partie téléphonique classique, qui va vers un répartiteur téléphonique, et la partie données, qui va vers l'équivalent d'un DSLAM, lequel peut utiliser la fibre optique du PON pour atteindre le local technique de l'opérateur.

Le réseau d'entreprise, à la sortie du modem VDSL, peut être de deux types : ATM ou Ethernet. Dans le premier cas, les trames ATM sont directement acheminées sur ce réseau par un commutateur ATM connecté au modem. Dans le second cas, un hub Ethernet est mis en place, complété par un routeur.

## La parole et la vidéo sur xDSL

Nous avons vu qu'en xDSL la parole téléphonique classique était transportée parallèlement aux données sur la partie basse du spectre. Cette technologie convient très bien aux opérateurs historiques, aussi appelés ILEC (Incumbent Local Exchange Carrier). Les nouveaux venus, ou CLERC (Competitive Local Exchange Carrier), peuvent aujourd'hui espérer concurrencer les opérateurs historiques grâce à la déréglementation de la boucle locale.

Pour prendre en charge des clients sur la boucle locale de l'opérateur historique, ces opérateurs entrants font passer la parole téléphonique sur la partie DSL. On appelle cette solution ToDSL (Telephony over DSL). Le passage de la parole sur la partie donnée rentre bien dans la catégorie de la ToIP.

Les paquets de parole devant arriver au récepteur avant 150 ms, il faut qu'une priorité leur soit appliquée. Dans ce cas, la dizaine de kilobits par seconde de la parole compressée passe assez facilement. Il faut toutefois que la priorité puisse s'exercer non seulement

sur la partie modem mais aussi sur les parties réseau précédant et suivant les deux modems. Cela suppose, pour la partie réseau d'entreprise, l'application d'une technique de priorité et, pour le réseau du FAI, la possibilité de négocier un SLA (Service Level Agreement) compatible avec le temps maximal de traversée de bout en bout.

Une autre solution, moins intégrée mais plus simple à mettre en œuvre, est commercialisée par de nombreux FAI pour offrir un service téléphonique de type ToDSL. Elle consiste à utiliser une bande spécifique du modem de 4,3 MHz, offrant un débit de 32 Kbit/s. L'inconvénient de cette solution est que si la parole téléphonique n'est pas utilisée, la bande passante correspondante est perdue. Cependant, comme la bande passante utilisée est très petite, cela ne pose pas vraiment problème.

La ligne DSL doit aussi convoyer la signalisation téléphonique, ce qui constitue la deuxième difficulté après la contrainte temporelle. Sur le modem, plutôt que d'utiliser une priorité sur les données, il est possible d'utiliser l'AAL1, qui offre des fonctionnalités de synchronisation et de priorité. Cette solution, appelée VoATM (Voice over ATM), est complémentaire de la technologie ToDSL.

La vidéo est un deuxième service qui peut être offert par les modems xDSL. S'il est encore difficilement imaginable de voir ce système supplanter la vidéo diffusée à grande échelle, la vidéo sur DSL, ou VoDSL (Video over DSL), commence à être déployée par de nombreux FAI pour des diffusions limitées et des services de VoD (Video on Demand).

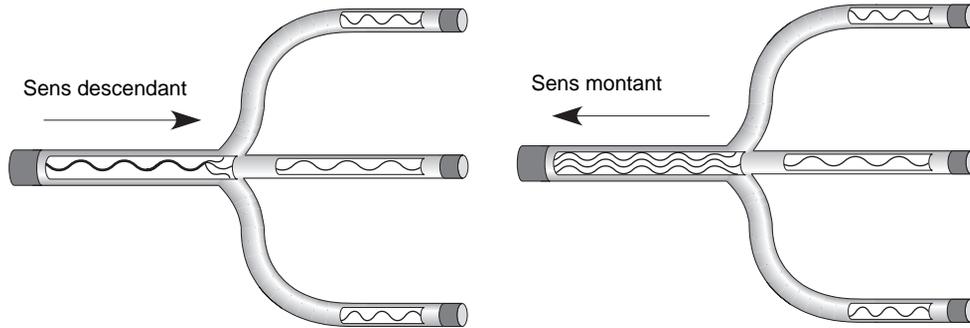
Ici encore, les deux solutions que nous avons examinées pour la téléphonie sont possibles pour la vidéo : soit on intègre les paquets de vidéo dans le flot en leur donnant si possible une priorité forte, soit on leur affecte un canal spécifique. Dans ce dernier cas, la largeur de la bande passante affectée à la vidéo diffère suivant les opérateurs pour aller de 800 Kbit/s à quelques mégabits par seconde. Pour une télévision à 800 Kbit/s, il suffit de récupérer 25 des 256 sous-bandes, chacune transportant 32 Kbit/s.

Dans le cas du multipoint, c'est-à-dire de la diffusion limitée à un petit nombre d'utilisateurs, la vidéo est compressée en MPEG-4 ou éventuellement en MPEG-2 et émise en utilisant un protocole multipoint. Le plus performant de ces protocoles est IP Multicast, puisque les paquets sont à l'origine IP. Cependant, comme il faut compresser au maximum les données vidéo, le choix du codec vidéo est capital pour que le flot arrive dans les temps.

## La téléphonie sur CATV

Les câblo-opérateurs disposent d'un environnement leur permettant de relier l'utilisateur à un ou plusieurs opérateurs. Ce câblage est réalisé à partir du CATV (Community Antenna Television) reliant la tête de réseau aux utilisateurs, comme l'illustre la figure 13.5. Les canaux de télévision dans le sens descendant sont diffusés sur toutes les

branches du câblage. Dans le sens montant, les canaux doivent se superposer sur le tronc de l'arbre.



**Figure 13.5**

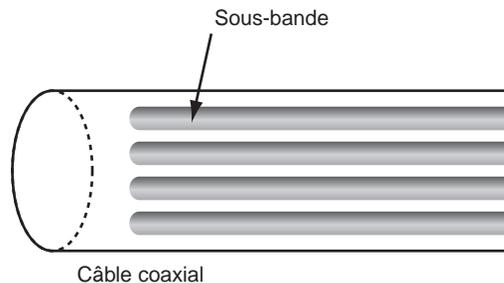
*Distribution de programmes TV par un câblo-opérateur*

Le câblage part d'une tête de réseau pour atteindre l'utilisateur après une diffusion sur l'ensemble des branches du câblage. Dans le cadre de la diffusion de la télévision, les différents programmes sont tous poussés vers les utilisateurs. Chaque abonné reçoit l'ensemble des chaînes et en sélectionne une à visualiser. Cette technique est à l'opposé de l'ADSL, où seule la chaîne sélectionnée par l'utilisateur est acheminée.

Dans le CATV, une division en fréquence est utilisée pour le transport des différents canaux de télévision (voir figure 13.6). La division en fréquence donne naissance à des sous-bandes, chaque sous-bande portant un canal de télévision.

**Figure 13.6**

*Multiplexage en fréquence dans le CATV*



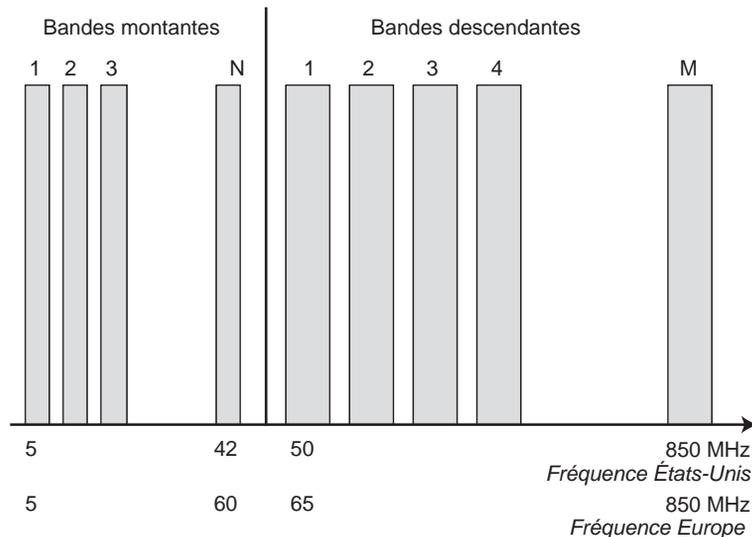
La ToIP est introduite de deux manières différentes. La première consiste à affecter une bande étroite de 32 Kbit/s par utilisateur pour réaliser de la ToIP entre le combiné de l'utilisateur et la tête de réseau qui est reliée à un opérateur télécoms. La seconde revient à multiplexer le flux de ToIP avec les données en allouant une priorité forte à la téléphonie. C'est cette seconde solution qui prend l'ascendant dans les technologies de CATV.

La ToIP passe donc par une sous-bande pour la connexion à un opérateur de type FAI. Cette sous-bande devrait être suffisante pour supporter la somme des débits crête des utilisateurs. Par exemple, 1 000 utilisateurs connectés à 1 Mbit/s, exigent un débit total potentiel de 1 Gbit/s. Cette valeur de 1 Gbit/s pourrait être obtenue dans un avenir proche. Cependant, si le nombre d'utilisateurs sur une tête de réseau est de 10 000, voire beaucoup plus, la solution offrant une sous-bande spécifique à chaque utilisateur n'est plus possible.

La solution à ce problème consiste à choisir sur le CATV une bande assez large, actuellement de 38 à 300 Mbit/s, et à utiliser une technique de multiplexage pour faire passer un maximum d'utilisateurs simultanément. La figure 13.7 illustre un partage de la bande passante d'un CATV en Amérique du Nord et en Europe. Les bandes montantes en Europe se situent entre 5 et 42 MHz et ont une largeur de 200 kHz à 3,2 MHz. Les bandes descendantes se situent entre 50 et 850 MHz, et la largeur des bandes de télévision est de 8 MHz.

Le nombre de bandes montantes et descendantes est laissé libre aux opérateurs. Les valeurs pour l'Amérique du Nord et l'Europe sont indiquées à la figure 13.7. Les bandes de télévision sont de 6 MHz.

**Figure 13.7**  
*Plages de fréquences dans un CATV*



Pour réaliser le multiplexage des utilisateurs sur la bande commune, deux grandes normes ont été proposées :

- IEEE 802.14, qui utilise une technologie ATM et qui est de moins en moins utilisée.
- MCNS-DOCSIS (Multimedia Cable Network System-Data Over Cable Service Interoperability Specification), qui a surtout été utilisée en Amérique du Nord au départ, mais que les câblo-opérateurs européens ont adopté depuis quelques années. Les versions 1

et 2 ont été normalisées par l'UIT-T, et une version spécifique pour l'Europe a été réalisée avec EuroDOCSIS. Les canaux de télévision PAL demandent une bande passante de 8 MHz contre seulement 6 MHz avec le système américain NTSC. Une version 3.0 de DOCSIS est en cours de normalisation, permettant de gérer 300 Mbit/s de flux descendant et 120 de flux montant.

Pour réaliser le multiplexage des utilisateurs sur la bande commune, la norme DOCSIS divise le temps en slots, numérotés entre 1 et 4 096. Les deux standards disponibles, DOCSIS 1.1 et DOCSIS 2.0, utilisent des tables d'allocation de slots qui indiquent qui a le droit de transmettre dans un slot. Jusqu'à 14 slots peuvent être utilisés simultanément pour une même communication. Les slots avec accès aléatoire, que l'on appelle slots de contention, permettent aux stations d'effectuer leur réservation. L'accès n'est pas vraiment aléatoire, puisqu'il utilise l'algorithme en arbre BEB (Binary Exponential Backoff) pour résoudre les collisions.

La différence entre les deux standards réside dans la prise en charge de la qualité de service, qui n'est garantie que dans la version 2.0. Cette qualité de service est totalement compatible avec le modèle DiffServ, présenté au chapitre 6, qui consiste à marquer le champ de qualité de service de chaque paquet d'une valeur correspondant à son niveau de priorité.

En conclusion, le CATV permet de faire passer de la téléphonie sur IP, mais la version DOCSIS 2.0 doit être choisie pour permettre l'obtention de la qualité de service nécessaire. La version 3.0, qui utilisera IPv6, permettra une meilleure utilisation de la bande commune tout en gardant le principe de la différenciation de classe DiffServ.

## La téléphonie sur fibre optique

La fibre optique commence à se développer dans la boucle locale pour desservir des applications à très haut débit. La parole téléphonique est ici multiplexée avec les autres applications, mais en y ajoutant une classification permettant aux paquets de téléphonie de transiter sur la fibre en priorité.

Nous ne ferons ici qu'introduire cette technologie, qui ne pose pas de problème particulier pour le passage de la ToIP, qui est multiplexée avec la plus haute priorité DiffServ. Comme nous allons le voir, ce sont la technologie Ethernet et donc les fonctionnalités de classification DiffServ au niveau Ethernet, permises par le champ IEEE 802.1p, qui sont utilisées.

Plusieurs technologies de fibre optique sont disponibles suivant l'emplacement de la prise terminale. Cette prise peut être chez l'utilisateur, ce qui donne naissance au FTTH (Fiber To The Home), ou à l'entrée du bâtiment avec le FTTB (Fiber To The Building). Cette dernière solution est, par exemple, très utilisée au Japon, complétée par une technologie VDSL sur les derniers mètres.

La solution FTTH permet d'atteindre des débits de 1 Gbit/s et FTTB + VDSL de 80 Mbit/s. D'autres solutions sont également disponibles en arrêtant la fibre optique au trottoir, avec FTTC (Fiber To The Curb), voire au quartier pour desservir de petits DSLAM très près de l'utilisateur afin que des technologies Ethernet « first mile » puissent être utilisées efficacement.

Parmi les différents pays raccordés en fibre optique, le plus câblé est certainement le Japon, où NTT déploie un réseau optique depuis 1990. Ce déploiement a été partiellement arrêté lorsque les modems ADSL sont arrivés sur le marché, avec en particulier les offres de SoftBank à bas prix. NTT a dû concurrencer cette solution pendant plusieurs années en offrant la même technologie. Aujourd'hui, le déploiement optique a repris. Les utilisateurs sont connectés à 100 Mbit/s et les entreprises à 1 Gbit/s. La fibre optique représentait approximativement début 2007 25 % du marché japonais de la connexion haut débit Internet.

Le marché français est en attente d'un démarrage, tiré essentiellement par France Télécom et Free, mais qui attend une position officielle de l'ARCEP, l'organisation de régulation française, avant d'investir massivement dans cette nouvelle technologie. Cette attente concerne une potentielle dérégulation du premier opérateur. La position de l'ARCEP est de partager les investissements entre opérateurs, la boucle locale optique étant dégroupée de fait par cette proposition.

Deux grandes solutions peuvent être déployées, la fibre active et la fibre passive. Dans le premier cas, on retrouve la structure en étoile autour d'un répartiteur. La seconde solution, qui semble vouloir s'imposer, consiste à partager la fibre optique entre toutes les Internet Box d'arrivée. Le câblage est conçu avec des étoiles optiques passives, qui diffusent le signal dans toutes les directions. De ce fait, chaque station reçoit une copie de l'ensemble des flux qui partent de tous les points d'accès au câblage optique.

L'avantage de cette solution est de permettre un partage total de la capacité de la fibre optique. Si un utilisateur n'utilise pas son accès ou peu, le trafic qu'il n'utilise pas peut être récupéré par les autres utilisateurs. Si l'on considère un câblage supportant le 2,5 Gbit/s et que la tête de réseau supporte 48 utilisateurs, chaque client a un débit minimal moyen de 50 Mbit/s. En revanche, s'il n'y a que cinq utilisateurs, chacun a 500 Mbit/s. On peut donc raisonnablement penser que l'utilisation classique par utilisateur n'étant que de quelques mégabits par seconde, un client pourra disposer à certains moments de 2,5 Gbit/s mais plus raisonnablement de 1 Gbit/s en tenant compte des contraintes de partage.

La trame utilisée est principalement la trame Ethernet, ce qui donne naissance à la technologie E-PON (Ethernet Passive Optical Network). Le nombre de clients raccordés peut atteindre 64 et la distance 20 km avec un débit symétrique.

## La téléphonie sur Quadruple-Play

La ToIP se déploie aujourd'hui sur le Quadruple-Play, par la superposition de quatre médias, les données, la télévision, le téléphone fixe de type ToIP et la téléphonie itinérante. C'est à cette dernière fonction que nous allons nous intéresser dans cette section.

Nous avons abordé au chapitre 7 les développements autour de la téléphonie par Wi-Fi. La téléphonie itinérante consiste à changer de technologie lorsque l'utilisateur se déplace sans couper la communication.

Une première solution, développée par BT (British Telecom) au Royaume-Uni, consiste en l'utilisation de Bluetooth et du GSM. Lorsque l'utilisateur est chez lui, il est connecté par Bluetooth à son Internet Box, et lorsqu'il sort du rayon d'action de Bluetooth, il passe en GSM.

Les solutions de Quadruple-Play développées en France depuis juillet 2006 et au Japon à partir de janvier 2007 associent Wi-Fi et le GSM ou une autre technologie 2 G ou 3 G. La difficulté de cette approche est de passer d'une solution de ToIP à une solution de type GSM.

L'approche qui a été choisie, en particulier par Orange, consiste à utiliser la technologie UMA (Unlicensed Mobile Access). Cette technique consiste à encapsuler la téléphonie GSM dans des paquets IP. De ce fait, lorsque le client est connecté à Wi-Fi, un logiciel se trouvant dans le poste client encapsule les données de supervision et de téléphonie GSM dans des paquets IP qui sont transportés par l'intermédiaire du point d'accès Wi-Fi vers un contrôleur UMA ou UNC (UMA Network Controller). De ce fait, on émule la fonction GSM au travers du réseau Wi-Fi. L'authentification du client est réalisée par la carte SIM du GSM.

Cette solution UMA permet de sécuriser la communication téléphonique comme celle d'un GSM, mais ne permet pas d'exploiter les logiciels de signalisation de la ToIP.

Dans la solution Unik proposée par Orange, le téléphone est bi-mode GSM/Wi-Fi. Lorsqu'il se trouve à portée d'un point d'accès, il se connecte en Wi-Fi, mais avec une émulation GSM. Lorsqu'il se trouve hors de portée de la cellule Wi-Fi de l'utilisateur, il fonctionne en mode GSM sur l'opérateur cellulaire Orange. Le passage de l'un à l'autre s'effectue sans coupure de la communication dans le sens Wi-Fi vers GSM, mais pas dans l'autre sens.

La sécurisation de la communication Wi-Fi est réalisée grâce à la clé secrète utilisée par le propriétaire de la LiveBox. L'inconvénient est de devoir modifier cette clé secrète lorsque le client veut se connecter sur une autre LiveBox que la sienne. Pour pouvoir se connecter à l'ensemble des LiveBox, il faudra attendre que la clé secrète de l'utilisateur puisse être récupérée en temps réel ou que les bornes Wi-Fi des LiveBox disposent d'un deuxième SSID (nom du point d'accès) permettant de construire un VLAN spécifique pour les accès externes.

Les solutions proposées par Free et Neuf Telecom sont assez différentes. Free propose deux catégories de téléphones. Le téléphone Wi-Fi se connecte aux différentes FreeBox ayant un accès Wi-Fi ouvert grâce au chargement sur le téléphone Free d'un logiciel

client mis en place par connexion sur la FreeBox de l'abonné. Cette solution permet à l'abonné de téléphoner sur n'importe quelle FreeBox HD, au nombre de 300 000 début 2007. Le téléphone GSM est un complément qui utilise les mêmes fonctionnalités sur la partie Wi-Fi et qui permet d'utiliser la puce GSM de n'importe quel opérateur mobile. Le passage de Wi-Fi à GSM ne peut pas se faire sans coupure, la communication étant interrompue lorsqu'on sort du champ de couverture Wi-Fi. Dans ce cas, il faut rappeler en GSM.

La solution Neuf Telecom est certainement la plus intéressante, à l'exception du passage sans coupure Wi-Fi vers GSM de la solution Orange. Cette solution consiste à utiliser la puissance de la signalisation SIP. L'offre TWIN de l'opérateur permet de se connecter sur tous les accès Wi-Fi ouverts, qu'ils soient de Neuf Telecom ou d'opérateurs concurrents. Par une signalisation SIP, l'utilisateur accède à son compte et téléphone au prix de son abonnement Neuf Telecom.

L'inconvénient majeur de cette solution est de ne pouvoir se connecter que sur les accès ouverts, puisque le téléphone TWIN ne peut acquérir la clé secrète des utilisateurs de points d'accès à l'exception de celle de sa propre 9Box. En revanche, sur les hotspots, qui ne sont pas sécurisés par une clé secrète, une fois l'authentification effectuée, le téléphone TWIN peut fonctionner normalement. Le coût d'accès aux hotspots peut toutefois être supérieur à une communication téléphonique GSM. Le passage ne s'effectue pas sans coupure entre les technologies Wi-Fi et GSM.

## Conclusion

La téléphonie sur IP a fortement augmenté avec l'arrivée des modems ADSL, qui permettent à l'utilisateur d'avoir un débit important sur Internet. La parole téléphonique ne demandant que quelques dizaines de kilobits par seconde, une surcapacité de la boucle locale permet une ToIP de bonne qualité puisque les paquets de parole passent rapidement.

Si l'environnement DiffServ est fortement employé dans les entreprises, il ne l'est pas du tout chez les particuliers. Cela vient du fait que la différenciation de trafic sur les Internet Box devrait donner lieu à une tarification. S'il y avait introduction de classes de service sans tarification, il est évident que tout le monde opterait pour la classe de plus haute priorité et que le problème serait inchangé. Comme les opérateurs ne peuvent ni ne veulent s'entendre sur une tarification de la différenciation de service, il n'est pas possible de l'introduire dans les Internet Box actuelles. La surcapacité reste la meilleure solution sans classification, et c'est ce que permettent les modems ADSL.

Nous avons introduit dans ce chapitre l'arrivée de solutions radio à partir des Internet Box pour compléter les services Internet. Cela a engendré l'arrivée de nouveaux terminaux associant la voix sur le GSM et la voix sur Wi-Fi.

La partie terminale devenant de plus en plus hertzienne, de nouveaux réseaux radio sont en cours de normalisation, comme la technologie WiRAN (Wireless Regional Area Network), qui devrait permettre de faire passer par une seule antenne un million d'utilisateurs de ToIP, mais évidemment nettement moins si l'on ajoute des données dans les communications des utilisateurs, sur une surface géographique à la taille d'une région.



## Filtrage des flux de ToIP

---

L'un des problèmes essentiels inhérents aux protocoles de signalisation réside dans le filtrage des flux. Très souvent, les entreprises utilisent des mécanismes de translation d'adresse (NAT), lesquels sont incompatibles avec les traitements appliqués sur les flux multimédias.

Les boîtiers NAT et les pare-feu imposent les trois verrous suivants à la traversée des flux de ToIP :

- Les protocoles de signalisation pour la ToIP standard, tels que H.323, SIP et MGCP, annoncent l'adresse IP des correspondants à l'intérieur des messages qu'ils envoient. Or si ces adresses IP sont privées, elles ne sont pas exploitables en dehors du réseau local déployant le NAT.
- Un boîtier NAT n'effectue la correspondance d'une adresse IP locale privée avec une adresse IP publique correspondante que lorsqu'un terminal local effectue une connexion réseau. En conséquence, si un terminal distant tente de joindre le terminal derrière le NAT, aucune règle n'est ajoutée dans le boîtier NAT, et le correspondant derrière le NAT est injoignable. Il peut émettre un appel, mais pas en recevoir.
- Les ports utilisés dans le canal de données sont négociés dynamiquement dans le canal de contrôle par les protocoles de signalisation. Un pare-feu ne peut donc statiquement ouvrir les ports du canal de données puisque ces choix sont imprévisibles.

Sauf à trouver des parades, ces verrous compromettent la traversée des réseaux d'entreprise. Ce chapitre expose en détail cette problématique, en précisant comment fonctionne le NAT, à l'origine de ces verrous, pourquoi il reste d'un usage courant, quelles sont les différentes formes de NAT disponibles, quels sont les problèmes rencontrés et quelles solutions peuvent leur être apportées.

## Le mécanisme de NAT (Network Address Translation)

Le protocole IP version 4, que nous utilisons massivement actuellement, offre un champ d'adressage limité et insuffisant pour permettre à tout terminal informatique de disposer d'une adresse IP. Une adresse IP est en effet codée sur un champ de 32 bits, ce qui offre un maximum de  $2^{32}$  adresses possibles, soit en théorie 4 294 967 296 terminaux raccordables au même réseau. Pour faire face à cette pénurie d'adresses, et en attendant la version 6 du protocole IP, qui offrira un nombre d'adresses beaucoup plus important sur 128 bits, il faut recourir à un partage de connexion en utilisant la translation d'adresse, ou NAT (Network Address Translation).

Ce mécanisme se rencontre fréquemment à la fois en entreprise et chez les particuliers. Il distingue deux catégories d'adresses : les adresses dites publiques, c'est-à-dire visibles et accessibles de n'importe où (on dit aussi routables sur Internet), et les adresses dites privées, c'est-à-dire non routables sur Internet et adressables uniquement dans un réseau local, à l'exclusion du réseau Internet.

Le NAT consiste à établir des relations entre l'adressage privé dans un réseau et l'adressage public pour se connecter à Internet.

### Adresses privées et adresses publiques

Dans le cas d'un réseau purement privé, et jamais amené à se connecter au réseau Internet, n'importe quelle adresse IP peut être utilisée. Dès qu'un réseau privé peut être amené à se connecter sur le réseau Internet, il faut distinguer les adresses privées des adresses publiques. Pour cela, chaque classe d'adresses IP dispose d'une plage d'adresses réservées, définies comme des adresses IP privées et donc non routables sur Internet. La RFC 1918 récapitule ces plages d'adresses IP, comme l'indique le tableau 14.1.

Tableau 14.1 – Plages d'adresses privées

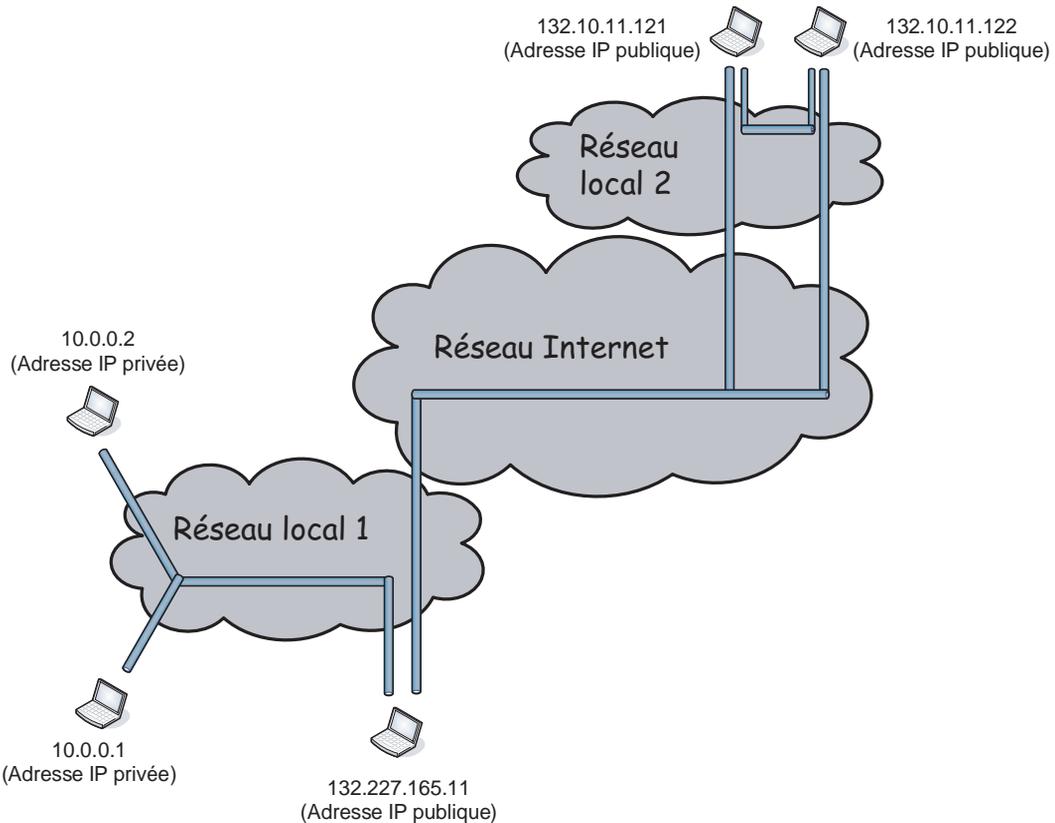
Classe d'adresses	Plages d'adresses privées
A	10.0.0.0 à 10.255.255.255
B	172.16.0.0 à 172.31.255.255
C	192.168.0.0 à 192.168.255.255

Dans ce cadre, et avant d'introduire la notion de NAT, les utilisateurs qui possèdent une adresse IP privée ne peuvent communiquer que sur leur réseau local, et non sur Internet, tandis qu'avec une adresse IP publique, ils peuvent communiquer sur n'importe quel réseau IP.

L'adressage privé peut être utilisé librement par n'importe quel administrateur ou utilisateur au sein de son réseau local. Au contraire, l'adressage public est soumis à des restrictions de déclaration et d'enregistrement de l'adresse IP auprès d'un organisme spécialisé,

l'IANA (Internet Assigned Numbers Authority), ce que les FAI effectuent globalement en acquérant une plage d'adresses IP pour leurs abonnés.

La figure 14.1 illustre un exemple d'adressage mixte dans lequel on distingue les différentes communications possibles, selon un adressage de type privé ou public.



**Figure 14.1**

*Adresses privées et publiques*

### **Partager une adresse IP privée**

Moyennant la souscription d'un accès Internet auprès d'un FAI, ce dernier fournit à ses utilisateurs une adresse IP privée. Dans un même foyer ou une même entreprise, deux utilisateurs ne peuvent communiquer en même temps sur Internet avec cette seule adresse IP fournie. Les adresses IP privées conviennent généralement pour couvrir un réseau privé, de particulier ou d'entreprise, mais elles ne permettent pas de communiquer directement avec les réseaux publics.

Pour résoudre ce problème et permettre à un terminal disposant d'une adresse IP privée de communiquer avec le réseau public, le processus de NAT fait intervenir une entité tierce entre un terminal, ayant une adresse IP privée, et tout autre terminal, ayant une adresse IP publique. Ce mécanisme consiste à insérer un boîtier entre le réseau Internet et le réseau local afin d'effectuer la translation de l'adresse IP privée en une adresse IP publique. Aujourd'hui, la plupart des boîtiers, ou Internet Box, des FAI proposent à leurs abonnés cette fonctionnalité. Toutes les machines qui s'y connectent reçoivent par le biais du service DHCP une adresse IP privée, que le boîtier se charge de traduire en une adresse IP publique.

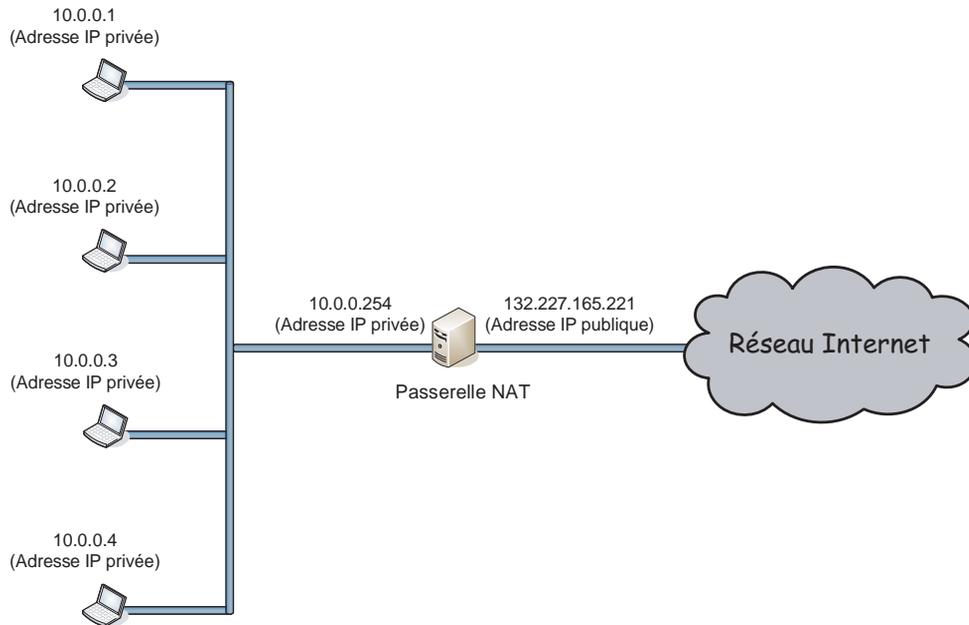


Figure 14.2

*Translation d'adresses*

La figure 14.2 illustre un exemple dans lequel une passerelle NAT réalise une translation d'adresses pour quatre terminaux. Cette passerelle possède deux interfaces réseau. La première est caractérisée par une adresse IP publique (132.227.165.221). Connectée au réseau Internet, elle est reconnue et adressable normalement dans le réseau. La seconde interface est caractérisée par une adresse IP non publique (10.0.0.254). Connectée au réseau local, elle ne peut communiquer qu'avec les terminaux qui possèdent une adresse IP non publique de la même classe.

Lorsqu'un terminal ayant une adresse IP privée tente de se connecter au réseau Internet, il envoie ses paquets vers la passerelle NAT. Celle-ci remplace l'adresse IP privée d'origine par sa propre adresse IP publique (132.227.165.221). On appelle cette opération

une translation d'adresse. De cette manière, les terminaux avec une adresse IP privée sont reconnus et adressables dans le réseau Internet par une adresse IP publique.

La translation d'adresses est bien sûr réalisée dans les deux sens d'une communication, afin de permettre l'émission de requêtes aussi bien que la réception des réponses correspondantes. Pour cela, le boîtier NAT maintient une table de correspondance des paquets de manière à savoir à qui distribuer les paquets reçus. Par exemple, si un émetteur dont l'adresse IP est 10.0.0.3 envoie vers la passerelle NAT un paquet à partir de son port 12345, la passerelle NAT modifie le paquet en remplaçant l'adresse IP source par la sienne et le port source par un port quelconque qu'elle n'utilise pas, disons le port 23456. Elle note cette correspondance dans sa table de NAT. De cette manière, lorsqu'elle recevra un paquet à destination du port 23456, elle cherchera cette affectation de port dans sa table et retrouvera la source initiale.

Ce cas de figure est illustré à la figure 14.3.

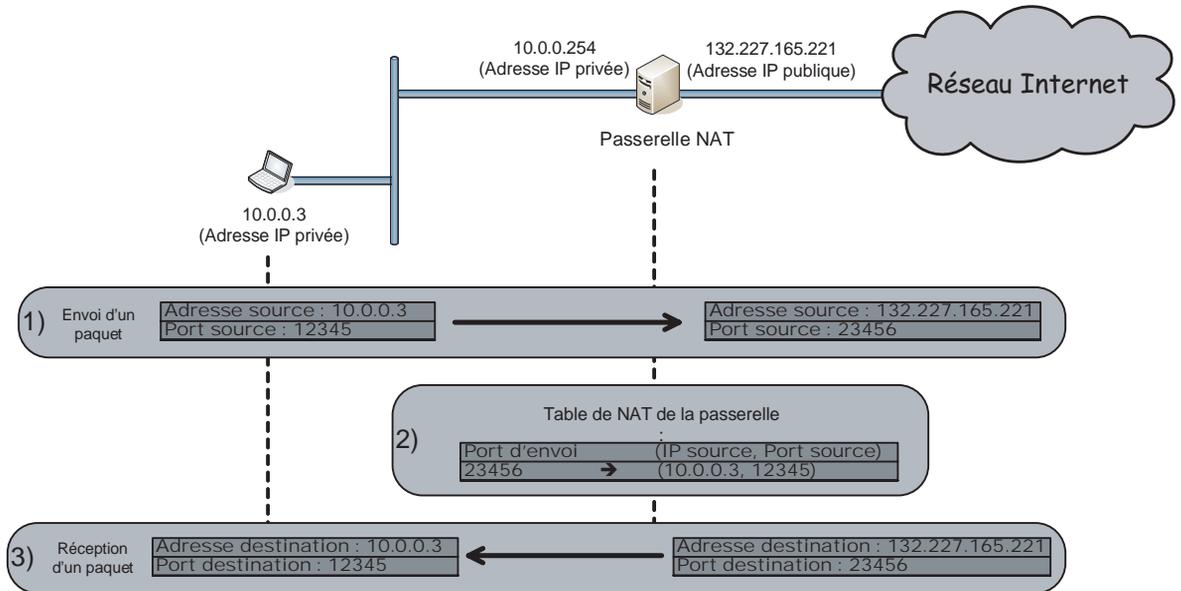


Figure 14.3

Modification de paquets lors du NAT

### Avantages du NAT

Le premier atout du NAT est de simplifier la gestion du réseau en laissant l'administrateur libre d'adopter le plan d'adressage interne qu'il souhaite. Étant privé, le plan d'adressage interne ne dépend pas de contraintes externes, que les administrateurs ne maîtrisent pas toujours. Par exemple, si une entreprise utilise un plan d'adressage public

et qu'elle change de FAI, elle doit modifier l'adresse de tous les terminaux qui composent son réseau. Au contraire, avec le NAT et un plan d'adressage privé, le choix d'un nouveau FAI n'a pas d'impact sur les terminaux. Dans ce cas, l'administrateur n'a pas besoin de reconfigurer les adresses IP de tous les terminaux de son réseau. Il lui suffit de modifier, au niveau de la passerelle NAT, le pool d'adresses IP publiques, qui est affecté dynamiquement aux IP privées des terminaux du réseau local.

Le deuxième atout du NAT est d'économiser le nombre d'adresses IP publiques. Le protocole réseau IP, qui est utilisé dans l'Internet actuel dans sa version 4, présente une limitation importante, car le nombre d'adresses IP disponible est faible comparé au nombre de terminaux susceptibles d'être raccordés au réseau Internet. Comme cette ressource est rare, sa mise à disposition a un coût pour les administrateurs qui souhaitent en bénéficier.

Le NAT comble cette pénurie d'adresses propre à la version 4 d'IP en offrant la possibilité d'économiser les adresses IP à deux niveaux distincts. Tous les terminaux d'un réseau local n'ont pas forcément besoin d'être joignables de l'extérieur, mais peuvent se limiter à une connexion interne au réseau. Par exemple, des serveurs d'intranet, des annuaires d'entreprise, des serveurs dédiés aux ressources humaines avec des informations confidentielles de suivi du personnel ou bien encore des serveurs de tests n'ont pas à être joignables à partir du réseau Internet, mais seulement en interne au sein de l'entreprise. En conséquence, ces serveurs peuvent se contenter d'une adresse IP privée, qui ne sera jamais « nattée » par le boîtier NAT puisque ces serveurs reçoivent des requêtes mais n'en émettent jamais.

Un deuxième niveau d'économie d'adresses IP publique est opéré avec le mécanisme que nous avons mentionné à la section précédente, qui permet de masquer plusieurs terminaux disposant chacun d'une adresse IP privée avec une seule adresse IP publique, en jouant sur les ports utilisés. Cette méthode est très couramment employée, car elle n'impose aucune condition quant au nombre de terminaux susceptibles d'accéder à Internet dans le réseau local. Elle n'en reste pas moins qu'un cas particulier du NAT. Nous verrons qu'elle est aussi la méthode la plus contraignante pour recevoir des appels téléphoniques.

Un autre avantage important du NAT concerne la sécurité. Les terminaux disposent en effet d'une protection supplémentaire, puisqu'ils ne sont pas directement adressables de l'extérieur. En outre, le boîtier NAT offre la garantie que tous les flux transitant entre le réseau interne et l'extérieur passent toujours par lui. Si un terminal est mal protégé et ne dispose pas d'un pare-feu efficace, le réseau dans lequel il se connecte peut ajouter des mécanismes de protection supplémentaires au sein de la passerelle NAT, puisqu'elle représente un passage obligé pour tous les flux. Globalement, l'administrateur concentre les mécanismes de sécurisation à un point de contrôle unique et centralisé. Cela explique que, bien souvent, les boîtiers NAT sont couplés avec des pare-feu filtrant les flux.

## Les trois catégories de NAT

Le mécanisme de NAT que nous avons pris comme exemple précédemment, consistant à jouer sur les ports pour masquer plusieurs terminaux avec une adresse IP unique, est un cas particulier. Il repose sur une translation de port appelée NPT (Network Port Translation). Lorsqu'elle se combine avec le NAT, on parle de NAPT (Network Address Port Translation).

Bien que les concepts soient différents, le processus de NAT inclut fréquemment par abus de langage le processus de NPT. En réalité, il faut distinguer trois formes de NAT, le NAT statique, le NAT dynamique et NATP. Ces formes peuvent se combiner selon les besoins de chaque utilisateur et les politiques d'administration établies dans un réseau. D'autres formes de classification du NAT sont possibles. La RFC 3489 en recense quatre types, par exemple. Nous nous contenterons de détailler dans les sections suivantes les formes les plus courantes.

### *Le NAT statique*

Dans le NAT statique, à toute adresse IP privée qui communique avec l'extérieur, une adresse IP publique fixe lui est affectée. Avec ce type de NAT, les utilisateurs du réseau local sont joignables de l'extérieur, car la passerelle réalise la correspondance d'une adresse IP locale en une adresse IP publique dans les deux sens. C'est un avantage indéniable, en particulier pour la téléphonie, car un utilisateur à l'extérieur du réseau privé peut appeler un abonné à l'intérieur du réseau privé puisqu'il connaît son adresse IP fixe.

Ce cas de figure est illustré à la figure 14.4. Le terminal ayant l'adresse IP privée 10.0.0.4 n'a pas de correspondance d'adresse IP publique, car c'est un serveur interne. Les administrateurs font l'économie d'une adresse IP pour ce serveur et s'assurent en outre que ce dernier n'est pas joignable directement de l'extérieur. De plus, un changement de fournisseur d'accès Internet ne remet pas en cause le plan d'adressage en local.

### *Le NAT dynamique*

Avec le NAT dynamique, une plage d'adresses IP publiques est disponible et partagée par tous les utilisateurs du réseau local. Chaque fois qu'une demande d'un utilisateur local (avec une adresse privée) parvient à la passerelle NAT, celle-ci lui concède dynamiquement une adresse IP publique. Elle maintient cette correspondance pour une période fixe, mais renouvelable selon l'activité de l'utilisateur, qui assure le suivi des communications.

Avec ce type de NAT, les utilisateurs locaux ne sont joignables de l'extérieur que s'ils ont une entrée dans la table de la passerelle NAT, autrement dit que s'ils entretiennent une activité avec le réseau Internet. En effet, les correspondants externes ne peuvent s'adresser qu'à la passerelle NAT pour envoyer leur flux. Or tant que le correspondant interne n'a pas d'activité réseau, aucune entrée ne lui est attribuée dans la table de NAT.

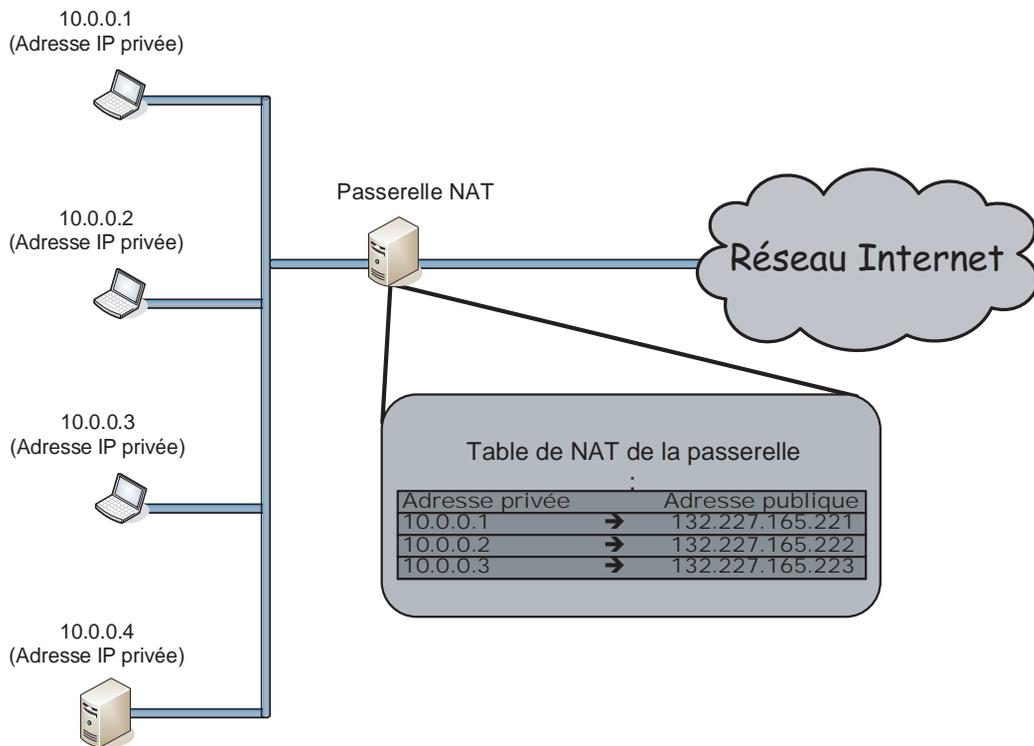


Figure 14.4

Le NAT statique

De plus, l'adresse IP qui leur est affectée est temporaire et peut être différente à la prochaine connexion, ce qui restreint les possibilités d'être joignable de l'extérieur.

Il existe même une forme de NAT particulière, appelée NAT symétrique ou « full cone » dans la RFC 3489, qui consiste à établir une correspondance entre l'adresse IP privée et publique selon la destination d'une communication. Autrement dit, un utilisateur du réseau local a une certaine adresse IP publique lorsqu'il communique avec un correspondant extérieur et une autre adresse IP publique lorsqu'il communique avec une autre destination.

Le modèle dynamique offre une plus grande souplesse d'utilisation que le modèle statique puisque les associations d'adresses IP privées et publiques n'ont pas besoin d'être mentionnées statiquement par l'administrateur, mais sont attribuées automatiquement. En outre, il présente l'avantage d'optimiser au maximum les ressources. Si un utilisateur n'exploite pas sa connexion Internet et se contente de sa connexion locale, la passerelle NAT n'a pas besoin de lui attribuer une adresse IP. Le NAT dynamique est cependant plus complexe puisqu'il impose à la passerelle NAT de maintenir les états des connexions

pour déterminer si les utilisateurs exploitent leur adresse IP publique ou s'il est possible, passé un certain délai, de les réutiliser.

Ce modèle ressemble à celui déployé avec la téléphonie RTC. Le nombre de lignes sortantes d'un commutateur téléphonique d'entreprise et même d'immeubles de particuliers est généralement inférieur au nombre de lignes entrantes. Autrement dit, tous les abonnés disposent d'un téléphone, mais tous ne peuvent appeler en même temps. Dans la pratique, il est assez exceptionnel que tous les abonnés appellent en même temps, si bien que ces derniers ne perçoivent pas cette restriction, qui permet aux opérateurs de limiter le nombre de lignes. Avec le NAT dynamique, les notions sont différentes, mais le principe est le même : l'attribution des adresses IP se fait à la demande, avec les limitations du nombre d'adresses IP publiques disponibles que cela suppose.

## Le NAPT

Variante du NAT dynamique, le NAPT (Network Address Port Translation) est en fait celui que nous avons présenté précédemment sans le nommer. Il consiste à attribuer une même adresse IP à plusieurs utilisateurs d'un même réseau local.

Comme nous l'avons expliqué, pour associer une même adresse IP publique à deux terminaux ayant une adresse privée distincte, la passerelle NAT joue sur les ports des applications : une requête envoyée à partir du port A d'une source est retransmise avec le port B de la passerelle, tandis qu'une requête émise à partir du port C d'une autre source est retransmise avec le port D de la passerelle. De cette manière, la passerelle peut contrôler et distinguer chacune des demandes qui lui parviennent.

L'inconvénient de cette méthode est que seuls les utilisateurs du réseau local peuvent amorcer une communication vers l'extérieur. Autrement dit, ils ne peuvent répondre à une communication qu'ils n'ont pas préalablement initiée. Les correspondants externes à la passerelle NAT ne possèdent en effet des entrées que pour une adresse IP et un port source privés. Or si le port source est mentionné, c'est qu'une application a déjà été ouverte par le terminal du réseau local. Le correspondant externe n'a aucun moyen d'établir une telle association en lieu et place du terminal dont il ignore la véritable adresse IP.

Pour la téléphonie, les utilisateurs qui ont ce type de NAT subissent la forte contrainte de pouvoir appeler un correspondant et communiquer avec lui mais sans pouvoir répondre à un appel. Certaines méthodes, que nous détaillons ultérieurement dans ce chapitre, permettent cependant de contourner ces limitations.

Le NAPT est sans conteste la méthode la plus économe puisqu'elle permet de masquer tout un réseau local avec une seule adresse IP. Elle est la plus couramment employée chez les particuliers et les petites et moyennes entreprises.

## Les problèmes engendrés par le NAT

Pour être pratiques et courantes, les fonctionnalités du NAT n'en posent pas moins des problèmes de nature différente, comme les protocoles dits « sensibles » au NAT, la difficulté de recevoir une connexion derrière un NAT ou la sécurité.

Les sections suivantes détaillent chacun de ces problèmes.

### *Les protocoles sensibles au NAT*

Le problème le plus important à considérer concerne les protocoles dits « sensibles » au NAT. C'est le cas des principaux protocoles de signalisation utilisés pour les échanges multimédias, dont H.323, SIP et MGCP, mais également de bien d'autres protocoles, comme Kerberos, SNMP, DNS, ICMP ou encore les protocoles de partage de fichiers tels que FTP et les protocoles de mobilité tels que IP Mobile.

Ces protocoles ne se contentent pas de mentionner leur adresse IP dans l'en-tête des paquets qu'ils envoient, mais ils l'indiquent également dans le corps de leurs messages. Par exemple, avec le protocole SIP, un message d'invitation INVITE comporte dans le paquet des informations sur l'adresse IP de la source. Ces informations permettent d'établir entre les correspondants la connexion dans laquelle les données véritables (la voix ou la vidéo notamment) sont transmises. Dans cette situation, même si le boîtier NAT modifie l'adresse IP source du paquet, le récepteur ne peut répondre correctement à la requête puisque cette dernière comporte une adresse IP source initiale, qui est une adresse privée. Le récepteur envoie donc sa réponse vers l'adresse IP source spécifiée qui ne lui est pas accessible, et le paquet de réponse n'arrive jamais à son destinataire.

Cette contrainte ne se pose pas pour toutes les applications. Par exemple, les flux d'application Web utilisent le protocole HTTP, dont les paquets ne contiennent pas l'adresse IP de la source à l'origine de la requête. En conséquence, le récepteur peut répondre sans connaître de problème de routage. Ce cas est en fait celui de la majorité des protocoles, mais pas des protocoles de signalisation tels que H.323 et SIP.

### *Recevoir une connexion derrière un NAT*

Ce problème est spécifique au NAT, qui translate les utilisateurs à la fois selon une adresse IP et selon un port. La question qui se pose est de savoir comment solliciter une entité masquée derrière un boîtier NAT.

Nous avons vu le cas où un terminal en adressage local effectuait une demande de connexion. La table de NAT est alors mise à jour conformément à la demande du terminal local, et la connexion avec l'extérieur peut se poursuivre. Mais comment faire si ce n'est pas le terminal local au boîtier NAT qui initie la connexion, mais un terminal distant ? Dans ce cas, le terminal distant ne sait pas vers où envoyer sa demande de connexion, puisque la seule adresse publique est celle du boîtier NAT et que la table de NAT ne contient à ce stade aucune entrée permettant de déterminer à qui est destinée

cette communication. En conséquence, un terminal téléphonique qui se trouve derrière une passerelle NAPT peut émettre un appel, mais pas en recevoir.

Une solution élémentaire à ce problème pourrait consister à connaître le port d'écoute d'une application et à configurer sur le boîtier NAPT une règle de redirection des paquets externes à destination de ce port d'écoute vers une machine locale en particulier. Par exemple, tous les paquets reçus d'Internet à destination du port 34567 sont systématiquement redirigés vers le terminal dont l'adresse IP est 10.0.0.2. Si ce dernier a configuré son application pour utiliser le port 34567 comme port d'écoute, la connexion devient possible.

Malheureusement, cette solution n'est guère satisfaisante. Deux applications qui tournent sur deux terminaux distincts ne sont pas adressables simultanément. En outre, la procédure n'est pas automatique, et il est nécessaire de configurer statiquement les règles de redirection, ce qui rend le mécanisme contraignant pour l'administrateur du réseau, en plus de ne pas être toujours une fonctionnalité disponible sur les boîtiers NAPT. Sur la majorité des équipements, les règles de redirection sont configurées au moyen d'une interface Web propriétaire.

## ***La sécurité avec le NAT***

Comme les codes de contrôle (checksums) inclus dans les en-têtes TCP d'un paquet sont calculés en fonction de l'adresse et du port du terminal source, ils deviennent invalides lorsque la passerelle NAT a modifié l'un ou l'autre de ces deux éléments. Si le destinataire reçoit le paquet avec le code de contrôle initial, il considère le paquet comme corrompu et demande sa réémission. En conséquence, la passerelle NAT doit recalculer les codes de contrôle et remplacer les originaux afin que les paquets restent valides et ne soient pas considérés par le destinataire comme corrompus.

Pour cette raison, le mécanisme de NAT est davantage une parade à la pénurie d'adresses IP qu'une véritable solution. Il ne se met en place qu'au prix de traitements sensibles et pas toujours réalisables. Par exemple, si l'émetteur crypte ses flux avec une couche IPsec, il devient impossible pour la passerelle NAT d'accéder aux en-têtes TCP des paquets relayés et donc de les modifier, si bien qu'ils sont transmis de manière erronée aux destinataires, qui les refusent.

On peut considérer le NAT comme une forme de « hack », en ce qu'il impose une rupture entre un émetteur et son récepteur et ne respecte pas les en-têtes d'origine des paquets, puisqu'il doit retravailler certains champs pour que les paquets demeurent conformes aux spécifications des protocoles.

## ***En résumé***

Conçue essentiellement pour faciliter l'administration d'un réseau et offrir une solution de rechange aux restrictions d'adressage du protocole IP dans sa version 4, la translation d'adresses est aujourd'hui largement déployée, à la fois chez les particuliers et dans les

entreprises, sous différentes formes, plus ou moins restrictives. Elle fait néanmoins intervenir, de manière obligatoire, une entité tierce intermédiaire entre l'émetteur et le récepteur. Cette technique impose donc des traitements supplémentaires sur les flux. Or ces traitements ne sont pas toujours compatibles avec d'autres protocoles. En particulier, le NAPT bloque la réception d'appel. Surtout, les protocoles de signalisation les plus courants ne prennent pas en compte la translation d'adresses qui sera appliquée aux flux et insèrent dans leur message des adresses privées, invalides pour un récepteur distant.

Il existe des parades pour lever ces verrous, que nous allons présenter et discuter dans la suite du chapitre. Au préalable, nous allons évoquer un autre verrou fort, celui concernant les pare-feu, que nous pourrions traiter avec les mêmes solutions que le NAT.

## Le passage des pare-feu

Les pare-feu constituent des remparts indispensables pour se protéger des attaques extérieures. Ils sont aujourd'hui couramment employés, à la fois par les particuliers et par les entreprises. Par le biais de règles de filtrage, ils inspectent tous les paquets qui transitent et vérifient s'ils sont conformes à la politique de sécurité implémentée. Si c'est le cas, les paquets sont autorisés à traverser le pare-feu et à poursuivre leur cheminement vers leur destinataire. Si ce n'est pas le cas, ils sont détruits.

Les pare-feu les plus classiques distinguent cinq éléments qui caractérisent les flux : l'adresse IP de la source, le port utilisé par la source, l'adresse IP du destinataire, le port utilisé par le destinataire et enfin le protocole de transport spécifié dans un paquet. Une règle de filtrage mentionne donc la valeur de chacun de ces cinq éléments et ordonne une action à entreprendre lorsque toutes ses valeurs sont validées.

L'action entreprise revient soit à autoriser, soit à interdire le paquet, c'est-à-dire respectivement à laisser passer le paquet ou à le détruire. Typiquement, un pare-feu adopte pour politique de bloquer tous les paquets pour lesquels aucune règle d'acceptation ne convient. La politique inverse, consistant à autoriser tous les paquets pour lesquels aucune règle d'interdiction ne convient, est trop permissive.

L'état d'une connexion peut être un sixième élément à prendre en compte par un pare-feu. Lorsqu'une communication est établie avec les cinq éléments précédemment mentionnés, on considère que la connexion est à l'état actif ou établi. Autrement, l'état est considéré comme inactif.

On distingue ainsi deux catégories de pare-feu :

- Les pare-feu sans état (*stateless*), qui ne maintiennent aucun état des connexions et se contentent des cinq éléments caractéristiques d'un flux précédemment cités pour autoriser ou interdire les flux qui transitent dans le réseau.
- Les pare-feu avec état (*statefull*), qui maintiennent l'état des connexions et sont capables de distinguer si une communication s'effectue sur un port déjà ouvert ou sur un port que le paquet demande d'ouvrir.

La notion d'état est utile pour les protocoles à ports dynamiques. Avec des applications exploitant ces protocoles, une communication s'établit sur un port fixe vers un destinataire (canal de contrôle). Lorsque ce dernier est contacté, il convient avec l'émetteur de poursuivre la communication sur un autre port dynamiquement et arbitrairement sélectionné (canal de données). De cette façon, il reste disponible pour servir un autre correspondant qui tenterait de le joindre ultérieurement sur le port fixe. Face à une telle situation, seul un pare-feu avec état est capable d'autoriser l'usage du port dynamique. Pour cela, il lui faut analyser les paquets et déterminer s'ils sont liés ou non à une connexion préalablement établie.

Imaginons à titre d'exemple un protocole dans lequel un destinataire demande à la source de remplacer le port statique initial par un port dynamique qu'il lui impose. Les trois étapes suivantes sont nécessaires :

1. La source émet un premier paquet vers un port fixé du destinataire.
2. Le destinataire lui répond en précisant le port sur lequel il souhaite poursuivre la communication.
3. La source reprend la communication en utilisant le port mentionné.

Pour le pare-feu sans état, seules les deux premières étapes sont possibles puisqu'elles peuvent correspondre à une règle statique simplement fondée sur le « 5-uplets » initial. L'ouverture d'un port dynamique lui est impossible, car aucune règle n'en permet la définition, sauf à être totalement permissive et à ouvrir tous les ports possibles, ce qui constituerait une piètre politique de sécurité.

Pour le pare-feu avec état, la troisième étape est possible. En effet, ce type de pare-feu est capable d'analyser les flux et de déterminer que le port dynamique sur lequel la source tente de communiquer correspond à la demande qui a été faite précédemment par la destination. La gestion des états offre une performance accrue dans le traitement des paquets, mais cela a un coût en ce qu'elle introduit une latence supplémentaire pour le pare-feu, qui doit en outre savoir analyser les protocoles correctement et, pour cela, connaître leur syntaxe.

L'état est facilement discernable avec le protocole TCP, puisque ce dernier positionne des bits indiquant si la connexion est nouvelle, se poursuit ou se termine. Au contraire, le protocole UDP ne fournit pas ces indications. Pourtant, le pare-feu ne peut attribuer éternellement le statut actif à une connexion UDP. Il alloue généralement le statut actif à une connexion UDP pendant un certain délai. Passé ce délai, la connexion est considérée comme perdue et devient par conséquent inactive.

Cette manière de procéder est cependant très approximative et ne convient pas aux applications de ToIP, qui utilisent très majoritairement le protocole UDP pour transporter leurs données. Si, lors d'une communication, les intervenants cessent de parler, le silence correspondant n'est pas transmis, et aucun paquet n'est transmis durant cet intervalle de temps. Le pare-feu risque de considérer ce silence comme une terminaison de la communication, ce qui est erroné.

Un pare-feu est utile pour centraliser la politique de sécurité au sein d'un équipement unique. De cette manière, la gestion du contrôle des applications autorisées n'est pas laissée au libre choix des utilisateurs, mais est à la charge du réseau, ce qui réduit les possibilités de contournement des règles édictées au sein de l'entreprise.

Les fonctionnalités de NAT sont souvent implémentées en parallèle avec les fonctionnalités de pare-feu. En effet, l'opération réalisée par le NAT comme par le pare-feu doit s'appliquer au niveau d'une passerelle, point de jonction entre le réseau local privé et le réseau public. En outre, dans ces deux fonctions, une notion de filtrage est requise. Lorsque les flux traversent le réseau, le boîtier NAT détecte l'adresse IP source privée et la translate avec une adresse IP publique, tandis que le pare-feu inspecte l'adresse IP source pour savoir si l'utilisateur est autorisé à émettre des flux. Dans le même temps, le pare-feu détecte les ports et protocoles utilisés par l'application pour opérer un filtrage avec une granularité plus forte. Autrement dit, l'analyse des paquets est un mécanisme partagé par les fonctions de NAT et de pare-feu, ce qui justifie leur couplage.

## Méthodes de résolution de la translation d'adresse pour les flux multimédias

Dans cette section, nous considérons une application dont les flux posent problème parce qu'ils comportent dans le corps du message l'adresse IP de la source première, qui est une adresse IP non routable sur Internet.

Différentes solutions permettent de résoudre les problèmes engendrés par la translation d'adresse. Certaines sont définies pour traiter des applications en particulier, tandis que d'autres sont plus génériques et peuvent s'appliquer à n'importe quelle application, pour peu qu'elle soit compatible avec le mécanisme mis en œuvre.

### *Filtrage applicatif des données*

Pour opérer les modifications d'adresses IP et de port requises par la translation d'adresse, le boîtier NAT doit impérativement connaître le format et la syntaxe des protocoles sous-jacents. Les protocoles utilisés dans les couches basses de la communication réseau sont généralement classiques. Pour l'adressage IP, il s'agit du protocole IP (couche de niveau réseau) ; pour le port, il s'agit du protocole TCP ou UDP (couche de niveau transport). La majorité des flux sont donc reconnus et peuvent être traités.

On peut généraliser cette idée. En connaissant les spécificités d'un protocole, on peut opérer exactement les mêmes modifications que celles effectuées avec le NAT pour l'adresse IP et le port. Même si le problème est plus complexe, puisqu'il existe de nombreux protocoles, cette solution demeure parfaitement fonctionnelle. Ainsi, le boîtier NAT ne supporte plus uniquement les fonctionnalités de NAT classiques, mais est en plus capable d'analyser les flux pour déterminer quels sont les protocoles utilisés. En connaissant la syntaxe de ces protocoles, le boîtier peut effectuer toutes les modifications nécessaires.

La réponse apportée dans ce cadre est donc une solution de filtrage de tous les protocoles utilisés par les applications qui posent des problèmes de NAT.

### Les passerelles de niveau applicatif

Une nouvelle gamme de passerelles multimédias a été mise au point pour permettre la reconnaissance des flux. Appelées ALG (Application Layer Gateway), ces passerelles sont proposées dans un grand nombre de solutions commerciales, embarquées le plus souvent au sein d'un pare-feu. Les flux sont filtrés, et, s'ils sont reconnus, les modifications nécessaires au bon fonctionnement du NAT sont opérées parallèlement à l'autorisation accordée à ces flux de traverser le pare-feu.

C'est dans cet esprit que le projet libre Netfilter sous Linux (<http://www.netfilter.org>), propose la reconnaissance d'un très grand nombre de protocoles, des couches basses aux couches les plus hautes. Les modules de reconnaissance sont également disponibles pour le protocole H.323 (modules `ip_conntrack_h323` et `ip_nat_h323`), ainsi que pour le protocole SIP (modules `ip_conntrack_sip` et `ip_nat_sip`). Deux modules sont nécessaires, le premier (`ip_conntrack`) réalisant le suivi de connexion (car les flux utilisent des ports dynamiques qui doivent être détectés durant la communication) et le second (`ip_nat`) réalisant la translation d'adresse.

La technologie Netfilter est accessible par défaut dans toutes les distributions actuelles de Linux, par le biais de la commande *iptables*. Elle est fournie avec un ensemble de filtres pour la reconnaissance des protocoles les plus standards. Selon les distributions, le module de suivi de connexion n'est pas toujours fourni, mais peut être facilement complété avec la technologie patch-o-matic, qui automatise les mises à jour de Netfilter.

Cette solution est simple à mettre en œuvre et transparente pour l'application des utilisateurs. L'application n'a pas à modifier la structure des paquets envoyés. Le boîtier se charge en émission (du réseau local vers le réseau Internet) de les rendre valides et en réception (du réseau Internet vers le réseau local) de les distribuer au terminal adéquat.

Le boîtier NAT a une tâche beaucoup plus lourde à accomplir puisqu'il doit filtrer des protocoles complexes, de niveau applicatif, ce qui réclame des ressources de traitement importantes. Cette fonctionnalité implique la reconnaissance des protocoles, mais aussi des traitements pour remonter jusqu'au niveau applicatif des paquets susceptibles de freiner les transmissions. On peut donc lui préférer d'autres solutions.

### Tunneliser les applications

Une autre solution pour résoudre les problèmes du NAT consiste à adopter un tunnel de bout en bout, dans lequel le mécanisme du NAT n'a pas d'impact. Plus précisément, en établissant un VPN (Virtual Private Network) entre l'émetteur et le récepteur, toutes les données que l'un ou l'autre transmet sont acheminées à travers ce tunnel, qui garantit le routage correct des données. L'établissement de ce tunnel peut être réalisé avec les protocoles L2TP ou IPsec, par exemple.

La figure 14.5 illustre l'établissement d'un tunnel entre deux entités appartenant à des réseaux locaux distincts et disposant toutes deux d'adresses IP privées.

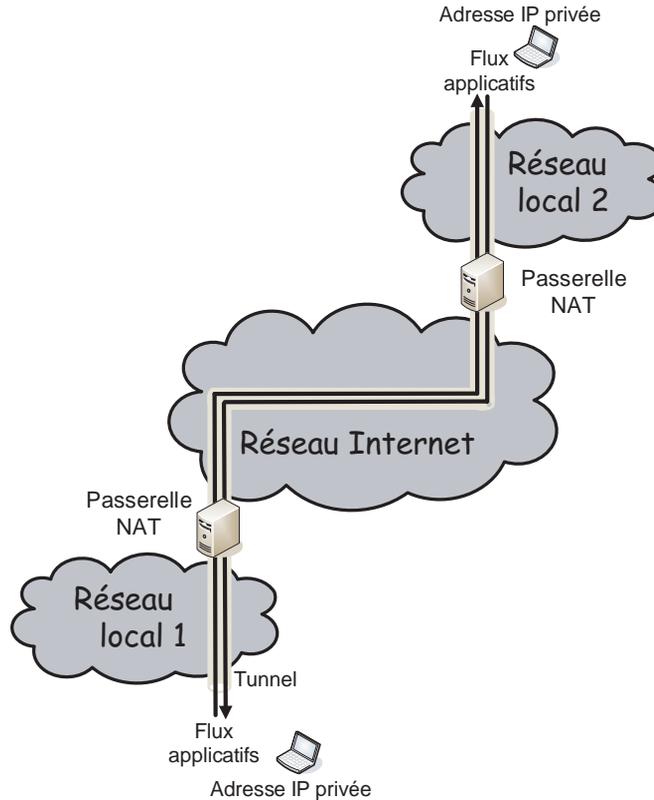


Figure 14.5

*Établissement d'un tunnel pour le transport des applications*

Cette méthode présente un double avantage : elle s'applique quelle que soit l'application utilisée et ne requiert aucune adaptation, ni de la part des applications clientes, ni de la part de la passerelle NAT. Elle comporte néanmoins de sérieux inconvénients, à commencer par le fait que l'émetteur et le récepteur sont contraints d'ouvrir un tunnel avant de procéder à leurs échanges. De plus, l'envoi des flux à travers un tunnel requiert des encapsulations protocolaires supplémentaires, ce qui alourdit notablement les transmissions.

L'établissement d'un tunnel est parfois impossible avec le NAT. Comme indiqué précédemment, le NAT doit modifier les en-têtes TCP, notamment le code de contrôle checksum. Or certains tunnels cryptent ce champ, rendant impossible sa modification.

L'opération de NAT devient alors incomplète et invalide tous les paquets transmis. Enfin, dans une entreprise, si le tunnel crypte les données, il est probable que le pare-feu bloque les flux correspondants, puisque les flux inconnus sont potentiellement dangereux et le plus souvent contraires aux politiques de sécurité mises en place.

### ***La gestion du NAT par le client***

Pour réduire l'implication et la charge du boîtier, une autre approche consiste à reporter les modifications de l'adresse IP au niveau de l'application elle-même. Si l'application a connaissance de l'adresse IP du boîtier, il lui suffit de la reporter dans le corps de son message. Après un NAT classique, le paquet émis devient complètement valide dans le réseau Internet. De cette manière, aucune modification n'est à apporter au boîtier NAT, ce qui allège la gestion de ce dernier.

Le premier problème que l'on rencontre est qu'une application qui se trouve au sein du réseau local n'a aucun moyen de détecter l'adresse IP publique avec laquelle la passerelle NAT va translater ses flux. L'opération de NAT est totalement transparente pour l'utilisateur. Il ne sait donc pas avec quelle adresse les utilisateurs externes au réseau natté vont l'identifier.

#### **La méthode UPnP**

Initialement prévue dans un cadre domestique, cette solution reste applicable dans tous les cas, comme l'explique l'architecture générale de déploiement de UPnP (Universal Plug-and-Play) fournie sur le site <http://www.upnp.org>.

Pour être mis en œuvre, le boîtier NAT et l'application cliente doivent tous deux supporter ce mécanisme. L'application cliente sollicite dynamiquement le boîtier NAT pour lui demander à la fois l'adresse IP publique et le port que le boîtier va utiliser pour translater ses flux. Le boîtier NAT qui ouvrira les ports dynamiquement selon les demandes du client est appelé IGD (Internet Gateway Device).

Son inconvénient principal est de présenter des problèmes de sécurité importants inhérents à sa manière de fonctionner. De plus, cette méthode est limitée et ne peut fonctionner si plusieurs NAT sont appliquées entre les correspondants. Pour des raisons évidentes de sécurité, cette méthode ne doit pas être généralisée.

#### **La méthode STUN**

Le protocole STUN (Simple Traversal of UDP through NAT) apporte une solution plus efficace aux problèmes du NAT. Défini dans la RFC 3489 de mars 2003, il permet aux terminaux de détecter dynamiquement le type de NAT qui leur est appliqué.

L'idée de base proposée par le protocole STUN consiste pour un client à demander l'adresse IP publique à un serveur externe, comme l'illustre la figure 14.6. En connaissant cette adresse, l'application peut reporter l'information dans ses messages.

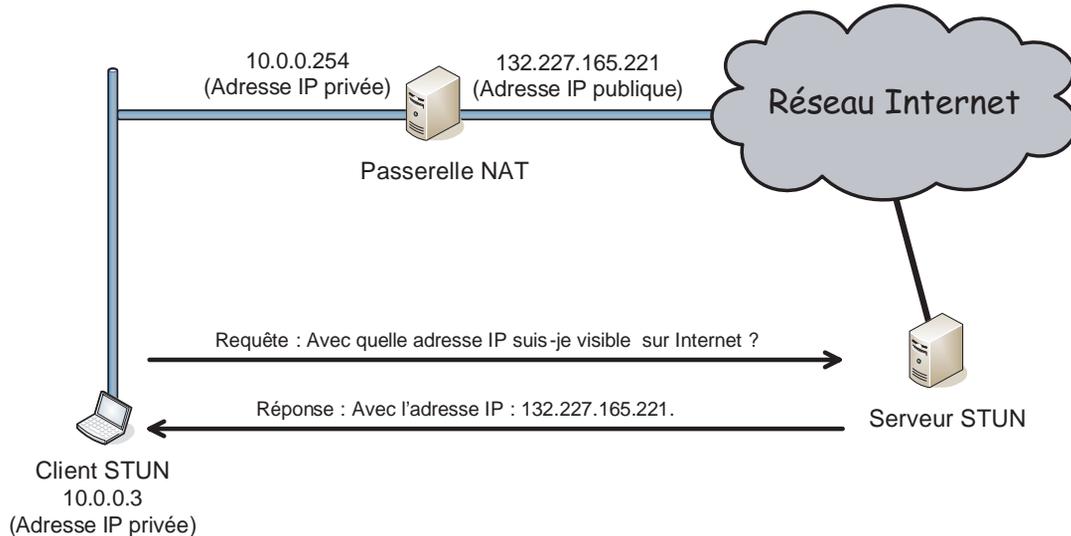


Figure 14.6

*Requête-réponse entre un client et un serveur STUN*

Le protocole STUN est conforme au modèle client-serveur et distingue deux catégories d'éléments :

- Le client STUN, entité qui émet des requêtes STUN, doit se trouver dans un domaine privé. Le client STUN est implémenté au sein d'une application devant traiter et envoyer des données dont le contenu (hors en-tête) comporte des adresses IP. C'est généralement le terminal de l'utilisateur qui a ce besoin, mais il peut aussi s'agir d'un terminal quelconque.
- Le serveur STUN, entité qui reçoit et répond aux requêtes STUN. Pour exécuter correctement sa fonction, le serveur STUN doit se trouver à l'extérieur du réseau privé, généralement dans le réseau Internet.

Le protocole STUN est standardisé, ce qui en fait une méthode à privilégier pour garantir aux applications la meilleure compatibilité possible. Son coût est modique, puisqu'il ne nécessite que l'ajout d'un serveur, placé à une position stratégique. Le serveur reçoit des demandes brèves et y répond de façon analogue, ce qui en fait une entité faiblement sollicitée par chaque utilisateur.

Cette opération n'est toutefois pas transparente pour l'utilisateur, puisque les applications doivent prendre en compte ce mécanisme dans les échanges qu'elles établissent. De plus, la méthode STUN, si elle supporte la majorité des types de NAT, ne s'applique pas au NAT symétrique. Dans ce dernier, en effet, la translation d'adresses dépend également de la destination des messages envoyés. En utilisant le protocole STUN, l'appelant contacte un serveur STUN et est vu par ce dernier avec une adresse IP qui peut être différente de celle que l'appelant utilisera plus tard pour contacter son véritable destinataire. Autrement dit, le serveur STUN retourne des informations d'adressage qui ne sont factuelles que pour lui et erronées pour tout autre correspondant. Le NAT symétrique lui échappe donc. C'est pour résoudre ce problème que la méthode TURN a été introduite.

### La méthode TURN

Le protocole TURN (Traversal Using Relay NAT) est actuellement à l'étude. Soumis à l'IETF sous forme de draft depuis octobre 2003, avec une révision en juillet 2004, il a été conçu pour lever les restrictions posées par la méthode STUN et permettre aux utilisateurs qui se trouvent derrière un réseau avec NAT ou derrière un pare-feu de communiquer quelle que soit la forme de NAT qui leur est appliquée.

Comme pour STUN, l'idée est de disposer d'un serveur à l'extérieur du réseau natté, mais, contrairement à STUN, le serveur ne se contente pas d'informer le client de la forme de NAT qui lui est appliquée. Il joue un rôle beaucoup plus actif et participe à l'acheminement des données en tant que relais. Un client qui souhaite établir une communication avec un correspondant doit envoyer toutes ses requêtes et données vers le serveur STUN, qui les transmet ensuite au correspondant.

Le serveur doit dédier des ressources à ses utilisateurs en termes de bande passante. L'accès à ce service doit nécessairement être sécurisé et restreint aux seuls utilisateurs qui sont autorisés à l'utiliser. Dans la pratique, les clients partagent avec le serveur un secret partagé permettant de les authentifier. Préalablement à leur envoi de données, les clients émettent une requête sécurisée par TLS.

Là encore, la communication doit être initiée par le terminal local au mécanisme de NAT. En outre, le terminal local au NAT ne peut communiquer qu'avec un seul correspondant distant en passant par le serveur TURN.

### La plate-forme ICE

ICE (Interactive Connectivity Establishment) n'est pas un protocole, mais une méthodologie pour le passage des flux qui sont soumis à des translations d'adresses. Elle prescrit l'utilisation des protocoles STUN et TURN et fournit un cadre d'application au cas par cas. ICE est actuellement à l'étude sous forme de draft IETF.

## En résumé

Les nombreuses solutions présentées dans cette section sont efficaces, mais ne sont pas applicables dans tous les cas. La translation d'adresses est, par exemple, parfois incompatible avec l'utilisation de tunnel VPN avec le protocole IPsec. Dans ce cas, lorsque les flux sont cryptés, il est impossible que le serveur en charge de la translation d'adresses intervienne pour modifier les paquets et assurer la liaison avec le correspondant. Du reste, l'utilisation de moyens de cryptage n'est pas toujours tolérée par les pare-feu d'entreprise, qui veulent contrôler et pour cela reconnaître les flux qui transitent sur le réseau.

Il existe cependant une solution de rechange au mécanisme de NAT. Proposée par l'IETF sous le nom de RSIP (Realm Specific Internet Protocol), son objectif est similaire au NAT : prévenir la pénurie d'adresses IP en déployant un adressage privé au sein d'un réseau local. Mais la manière de procéder est différente et plus élégante.

Avec RSIP, un terminal dans un réseau local possède une adresse IP privée qui lui est affectée pour communiquer localement comme avec le NAT. Pour pouvoir communiquer avec le réseau public, le terminal doit solliciter une passerelle RSIP. Celle-ci affecte alors à l'utilisateur une adresse IP publique (dans le cas du NAT statique ou dynamique) ou bien une adresse IP publique avec un port TCP ou UDP associé (dans le cas du NATP).

Le terminal a donc immédiatement connaissance de l'adresse IP publique avec laquelle il est visible sur le réseau, et il peut l'utiliser dans ses messages normalement. L'attribution de l'adresse IP se fait pour une durée déterminée, au-delà de laquelle la passerelle NAT s'autorise à attribuer de nouveau cette adresse IP. Néanmoins, le terminal peut négocier à tout moment une nouvelle demande pour conserver son adresse IP publique.

Par ailleurs, s'ils en font la demande, les terminaux peuvent se voir attribuer plusieurs adresses IP correspondant à plusieurs réseaux locaux et publics, afin de pouvoir communiquer parallèlement sur chaque réseau, tout en respectant la politique d'adressage propre à chaque réseau dans lequel elle communique.

Les terminaux ne peuvent cependant se passer de faire transiter les flux par la passerelle RSIP et doivent encapsuler tout leur trafic dans un tunnel (de type GRE, IPsec ou autre) établi avec leur adresse privée avec la passerelle RSIP. Celle-ci, en recevant les paquets de ce tunnel, décapsule simplement les parties d'adressage privé et envoie le paquet restant sur le réseau public. De cette façon, la passerelle RSIP reste indispensable pour garantir le masquage des éléments du réseau local.

Bien que globalement beaucoup moins utilisé que le NAT, le protocole RSIP présente le grand avantage de ne pas nécessiter, contrairement au NAT, de modifier dynamiquement les paquets au niveau de la passerelle. Dès leur émission par les terminaux dans le réseau local, les paquets sont valides dans le réseau public. Il est en outre compatible avec les mécanismes de NAT et peut cohabiter avec ces derniers.

## Conclusion

L'adoption généralisée du protocole IPv6 permettra un adressage beaucoup plus important des terminaux. En principe, tous les terminaux de la planète pourraient avoir une adresse IP fixe. Mais cela ne résoudra sans doute pas totalement les problèmes mentionnés dans ce chapitre.

Il n'est donc pas certain que le NAT sera abandonné à l'arrivée du protocole IPv6. En masquant le véritable plan d'adressage d'un réseau local, le NAT contribue à sécuriser les terminaux, qui ne sont pas directement accessibles, mais uniquement joignables en traversant une entité effectuant le NAT et pouvant assurer des fonctionnalités complémentaires de contrôle. La gestion du réseau local est en outre indépendante de toute autre contrainte émanant du fournisseur d'accès.

De plus, le besoin des utilisateurs de partager un adressage public ne sera pas forcément adapté au processus mis en place par les fournisseurs d'accès Internet. Ces derniers pourront à l'avenir continuer à ne distribuer qu'une seule adresse IPv6 à leurs abonnés et à recommander l'usage du NAT pour partager cette adresse ou bien opérer de façon transparente cette fonctionnalité à l'aide du boîtier fourni.

Les problèmes du NAT étant par ailleurs couplés aux problèmes des pare-feu, notamment en ce qui concerne l'utilisation de ports dynamiques, il est probable que les solutions de rechange présentées dans ce chapitre seront encore mises à contribution pendant de longues années.



## Partie III

# Conclusion

Pour clôturer cet ouvrage, nous évoquons dans cette partie quelques aspects notables de la téléphonie sur IP et traçons les grandes lignes des technologies et infrastructures potentielles de demain, sur lesquelles travaillent les principaux acteurs de la ToIP.

Le chapitre 15 résume les points clés permettant de fournir un service de ToIP de qualité satisfaisante et conforme aux spécificités des flux multimédias temps réel. Il esquisse également l'architecture GAN (anciennement UMA), qui est une orientation possible des réseaux mobiles de troisième génération, et qu'exploitent actuellement plusieurs opérateurs téléphoniques.

Le chapitre 16 présente une alternative à GAN particulièrement séduisante, et que poussent de très nombreux industriels : IMS (IP Multimedia Subsystem). L'IMS définit une architecture qui s'appuie sur la capitalisation des protocoles et mécanismes déjà existants dans le monde IP et s'adapte conformément aux contraintes des flux temps réel. Cette architecture a pour objectif de permettre la convergence des flux vers un réseau cœur unique, quels que soient les utilisateurs, leurs terminaux et surtout les réseaux d'accès utilisés.



## Les cinq problèmes clés de la ToIP

---

La téléphonie sur IP va inéluctablement remplacer la téléphonie numérique classique. Les enjeux sont considérables puisque l'ensemble des entreprises aura adopté cette technologie dans les dix années à venir.

Ce chapitre détaille les cinq problèmes clés auxquels il est important de réfléchir avant de décider de passer à la ToIP. Ces problèmes sont les suivants :

- **Sécurité.** Dans les versions classiques de la téléphonie, la sécurité est fortement garantie par un réseau spécifique, lequel ne peut être attaqué par l'émission de paquets d'attaque puisque le réseau n'est pas à transfert de paquets. Dans la ToIP, la confidentialité est assez simple à garantir par le biais de tunnels. Reste le problème de l'authentification de l'utilisateur, qui mérite réflexion.
- **Disponibilité.** Dans la téléphonie classique, la disponibilité est aux 5 « neuf », c'est-à-dire que le système est en état de marche 99,999 % du temps. Dans la ToIP, elle passe aux 3 « neuf », soit 99,9 %, avec un bon fournisseur de service IP et plutôt moins en général. La question est de savoir comment prendre en compte cette problématique pour revenir à des disponibilités plus acceptables.
- **Gestion.** La gestion du réseau téléphonique commuté est relativement simple, puisqu'elle consiste à maintenir des circuits téléphoniques. Avec l'intégration de la ToIP dans le réseau de données, la gestion de l'environnement téléphonique devient beaucoup plus complexe. Comment la nouvelle génération de réseaux intégrant la ToIP va-elle pouvoir répondre à cette question ?

- **Contrôle.** Comme la gestion, le contrôle de la téléphonie est assez simple dans l'environnement unique des circuits numériques. L'intégration de la ToIP dans un réseau de données global complexifie grandement le contrôle, alors même qu'il s'agit d'un service crucial compte tenu des contraintes temps réel de l'application de téléphonie.
- **Qualité de service.** La téléphonie par paquets est une application complexe, pour laquelle une excellente qualité de service est nécessaire. Cette problématique ayant été amplement commentée tout au long de l'ouvrage, nous n'y reviendrons que pour en résumer l'essentiel.

## La sécurité

La sécurité est un problème capital, bien que trop souvent délaissé pour diminuer les coûts d'investissement, et qui pose des problèmes qui ne sont pas toujours simples à résoudre.

Dans la téléphonie numérique, le système est quasiment fermé. Les commutateurs ne peuvent donc être atteints par des informations circulant dans le réseau. Cette particularité provient de l'unicité de l'application qui circule sur le réseau. Avec le multimédia et l'intégration de la téléphonie dans l'ensemble des données, il devient particulièrement complexe de sécuriser l'application de ToIP.

Les besoins concernent l'authentification, autrement dit la protection contre le piratage des identités, la confidentialité, c'est-à-dire l'impossibilité d'écouter une conversation, l'intégrité de la conversation et la défense de la vie privée (privacy) :

- **Authentification.** C'est une des toutes premières priorités si l'on veut que la confiance s'installe dans cet environnement. De nombreux mécanismes sont disponibles à cet effet, qu'ils soient normalisés ou propriétaires. La norme la plus répandue provient du groupe de travail IEEE 802.1x que nous allons décrire.
- **Confidentialité.** Les solutions résident dans le chiffrement.
- **Intégrité.** La signature électronique est une solution facilement exploitable.
- **Vie privée.** Il s'agit là encore d'un défi, bien que de premières solutions apparaissent.

### L'authentification

L'authentification consiste à identifier la personne qui se connecte ou à authentifier un utilisateur dont le nom n'est pas connu mais qui a donné les éléments permettant de le reconnaître, comme son mot de passe.

La solution d'authentification la plus répandue est fournie par la norme IEEE 802.1x. Elle est surtout employée dans le monde Wi-Fi pour obtenir l'autorisation de traverser un point d'accès.

Cette solution utilise un serveur d'authentification de type RADIUS, qui contient le nom des personnes qui ont le droit d'accéder et un secret associé, comme leur mot de passe. L'authentification d'un client qui souhaite téléphoner s'effectue simplement par l'intermédiaire de ce protocole. Pour cela, il faut que le client puisse prouver qu'il connaît une clé secrète, un mot de passe par exemple. Ce secret ne passe pas en clair dans le réseau.

L'architecture qui permet de mettre en œuvre l'algorithme associé est illustrée à la figure 15.1. La norme 802.1x définit un contrôle d'accès du réseau fondé sur les ports. Sa fonction est d'authentifier et d'autoriser des équipements attachés au port d'un réseau local.

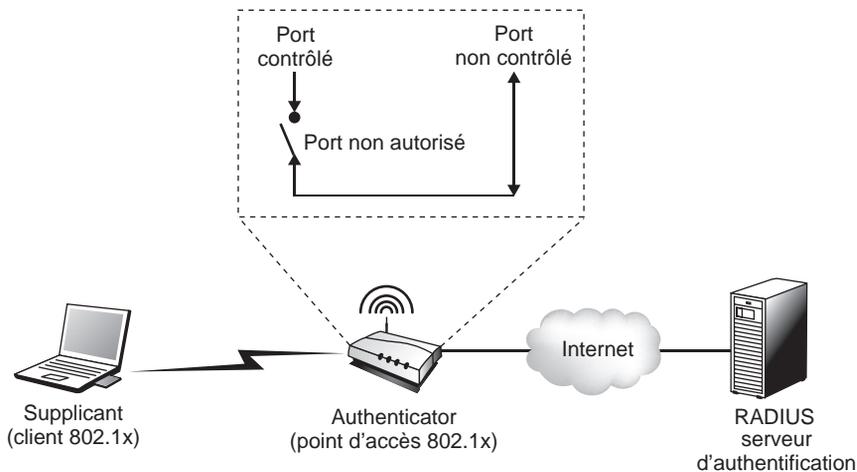


Figure 15.1

Architecture d'authentification IEEE 802.1x

Dans les réseaux sans fil 802.11, un port est une association entre une station et un point d'accès. Dans un réseau de ToIP, le port est en général l'association entre un téléphone IP et un commutateur. Le port contrôlé se comporte comme un interrupteur associé à deux états. Dans l'état *unauthorized*, seules les trames dédiées à l'authentification ne sont pas bloquées. Dans l'état *authorized*, le flux de téléphonie et plus généralement d'information transite librement.

Le protocole 802.1x définit les techniques d'encapsulation utilisées pour transporter des paquets EAP (Extensible Authentication Protocol) entre le port du client 802.1x et le port de l'équipement d'accès. Ces ports sont appelés PAE (Port Access Entity).

Le mécanisme de gestion des ports et d'encapsulation est illustré à la figure 15.2 pour une ToIP sur Wi-Fi. EAPoL indique les début et fin (optionnel) d'une session d'authentification avec les messages de notification EAPOL-START et EAPOL-LOGOFF.

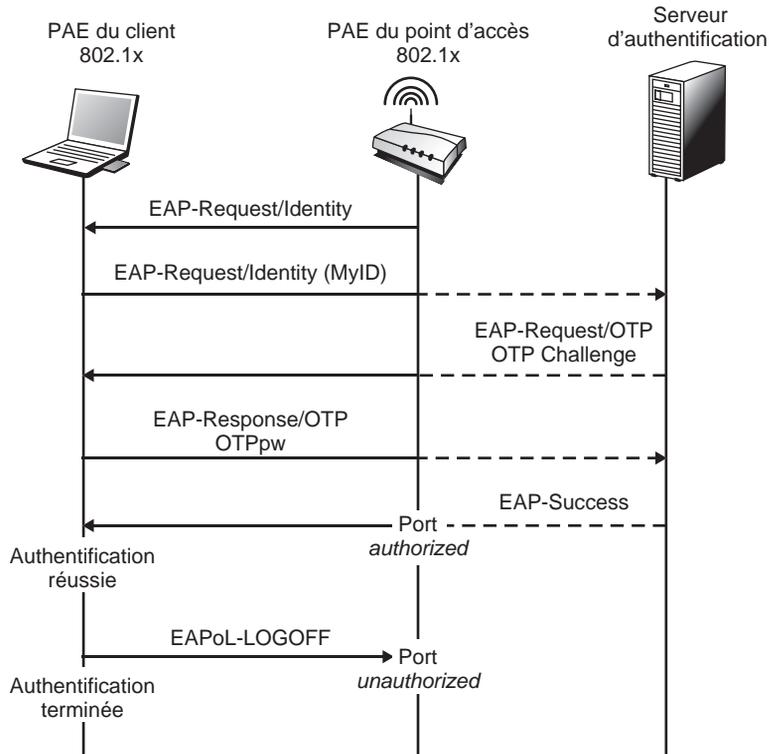
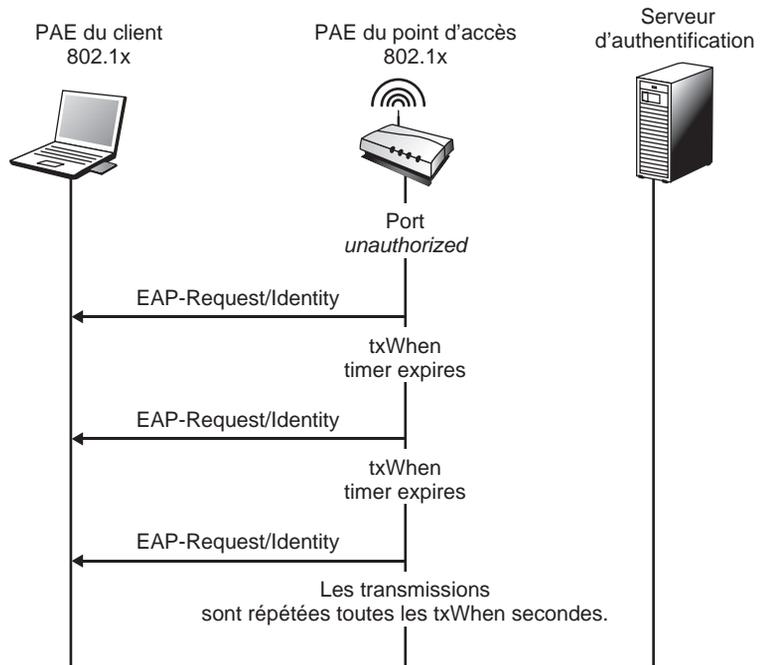


Figure 15.2

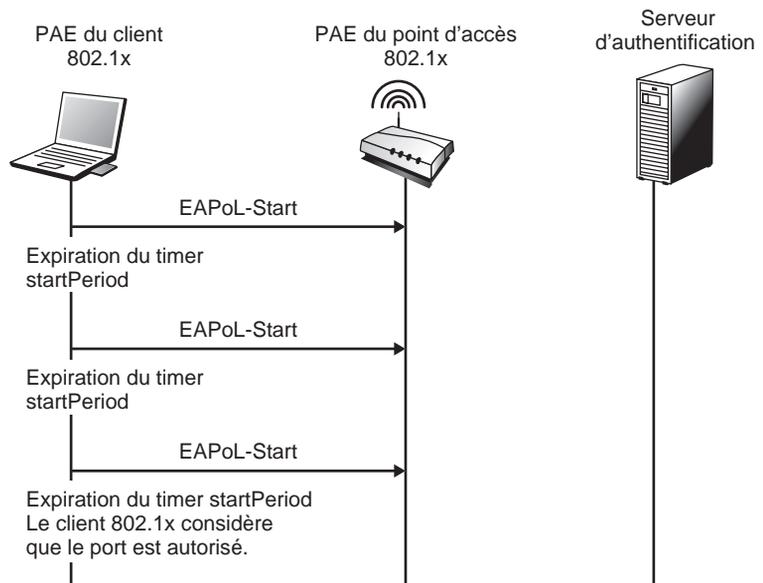
Mécanisme d'encapsulation et de gestion de ports de 802.1x

Dans l'état *authorized*, le port contrôle la durée de la session, c'est-à-dire le temps pendant lequel on considère que le client reste authentifié sans rien lui demander, à l'aide de la variable `REAUTHPERIOD`, dont la valeur par défaut est 3 600 s. En règle générale, le point d'accès retransmet les trames EAP perdues toutes les 30 s. De son côté, le client 802.1x retransmet les trames EAPOL-START non acquittées toutes les 30 s par un message EAP-REQUEST IDENTITY. Ces mécanismes sont illustrés aux figures 15.3 et 15.4.

**Figure 15.3**  
*Contrôle de l'entité PAE du point d'accès 802.1x*

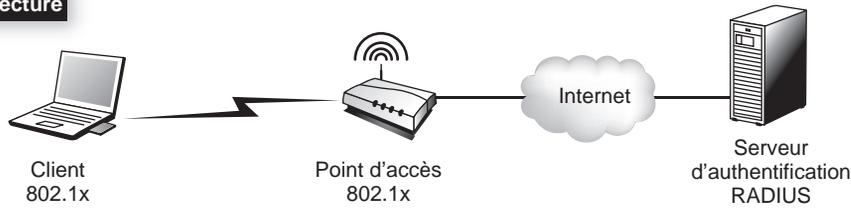


**Figure 15.4**  
*Machine d'état du PAE du client 802.1x*



Dans les réseaux, le protocole EAP est utilisé de manière transparente entre la station et le serveur d'authentification au travers d'un équipement de réseau compatible IEEE 802.1x jouant le rôle d'« authenticator ». Il est tour à tour encapsulé par le protocole RADIUS, qui est routable puisque transporté par IP. Ces encapsulations sont décrites à la figure 15.5 pour le cas d'un point d'accès jouant le rôle d'« authenticator », mais qui peut être remplacé par un équipement de réseau conforme à la norme 802.1x.

### Architecture



### Couches réseau

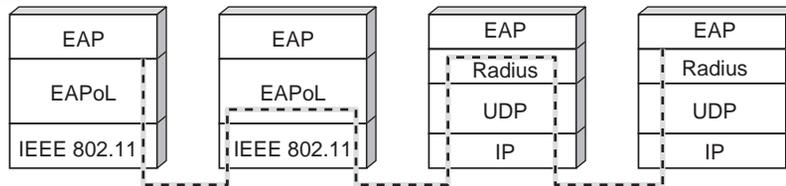


Figure 15.5

*Encapsulation des paquets sur le parcours entre l'utilisateur et le serveur d'authentification*

Schématiquement, l'insertion d'un terminal de ToIP dans un environnement 802.1x se déroule de la manière suivante :

1. Le terminal s'authentifie puis s'associe à l'équipement d'accès, par exemple un commutateur Ethernet ou un point d'accès Wi-Fi.
2. Afin de débiter l'authentification, le terminal émet toutes les 30 secondes une trame EAP-START.
3. Le point d'accès ou le commutateur transmet un message EAP-REQUEST.IDENTITY au terminal client 802.1x, qui produit en retour une réponse EAP-RESPONSE.IDENTITY comportant l'identité (EAP-ID) du terminal de ToIP.
4. À partir de ce paramètre, le point d'accès ou le commutateur déduit l'adresse (IP) du serveur d'authentification et transmet à ce dernier le message EAP-RESPONSE.IDENTITY encapsulé dans une requête RADIUS.

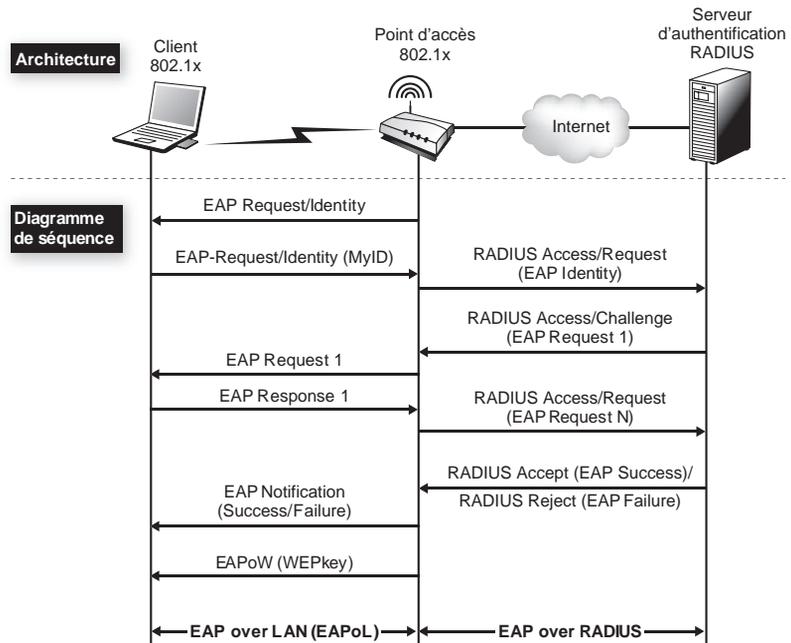
D'autres possibilités ont été implémentées, comme l'interrogation successive des serveurs RADIUS jusqu'à trouver celui qui est concerné par l'adresse IP.

5. Dès lors, des messages EAP requête et réponse sont échangés entre le serveur RADIUS et le terminal client 802.1x, le point d'accès ou le commutateur ne jouant qu'un rôle passif de relais.
6. Le serveur RADIUS indique le succès ou l'échec de cette procédure grâce à un message EAP-SUCCESS ou EAP-FAILURE. En fonction de cette information, le port transite dans l'état autorisé ou non autorisé.
7. À la fin du processus d'authentification, le message RADIUS ACCESS-ACCEPT provoque une transition dans l'état *authorized* du port concerné. Le message RADIUS ACCESS-REJECT force le port concerné à l'état *unauthorized*. Un port conserve son état courant durant une session d'authentification.
8. Dans le cas où l'authentification est réussie, le terminal client 802.1x et le serveur d'authentification calculent une clé de session, baptisée Unicast Key. Dans l'environnement Microsoft, cette valeur représente un couple de clés de deux fois 32 octets (ces attributs sont définis dans la RFC 2548 de mars 1999). Le serveur d'authentification transmet cette dernière au point d'accès ou au commutateur dans les attributs MS-MPPE-SEND-KEY et MS-MPPE-RCV-KEY du message RADIUS ACCESS-ACCEPT.
9. Le point d'accès ou le commutateur choisit alors une clé de chiffrement, dite Global Key, pour l'association de sécurité avec le terminal client 802.1x. Cette dernière est chiffrée et signée avec la clé de session reçue du serveur RADIUS puis délivrée au client 802.1x dans une trame EAP-KEY.

Le déroulement d'une authentification est illustré à la figure 15.6.

**Figure 15.6**

*Déroulement d'une authentification 802.1x*



Les faiblesses de cet algorithme sont les suivantes :

- L'identité du client passe en clair sur le réseau, ce qui est contraire à la protection de la vie privée. Des solutions sont proposées consistant à chiffrer le nom de la personne qui souhaite s'authentifier. Pour cela, il est possible d'utiliser une clé asymétrique.
- L'entité qui gère le serveur d'authentification peut connaître les noms des personnes connectées. Pour le moment, peu de solutions permettent de résoudre cette question. On se dirige actuellement vers des solutions de traçabilité fermée, grâce au chiffrement des noms des personnes connectées. Seule une autorité judiciaire possédant la clé adéquate peut déchiffrer les informations de traçabilité. Le rôle des cartes à puce est important dans ce processus, puisque l'information permettant la traçabilité y réside et ne peut être récupérée que grâce à la clé maître de l'autorité judiciaire.

### **Confidentialité et intégrité**

Pour la confidentialité de la communication, les données doivent être chiffrées. Le chiffrement s'effectue généralement par une clé symétrique, qui offre une grande rapidité du chiffrement et du déchiffrement.

La difficulté réside dans la distribution sécurisée de la clé entre les deux extrémités communicantes. Une clé asymétrique peut être utilisée pour le transport de la clé symétrique utilisée pour établir le tunnel chiffré transportant la ToIP.

L'intégrité est obtenue par une signature électronique. La signature électronique s'effectue à partir d'un *hash* chiffré de l'information à transmettre. Après déchiffrement, le destinataire doit obtenir la même valeur du *hash*.

Nous voyons donc que des solutions de sécurité efficaces peuvent être implémentées sur un réseau de ToIP. La plupart des solutions de téléphonie sur IP implémentent d'ailleurs ces fonctionnalités de façon plus ou moins ouverte.

### **La disponibilité**

La disponibilité désigne le temps pendant lequel un système est en état de marche ou, ce qui revient au même, le temps pendant lequel le système n'est pas en état de marche.

Comme expliqué précédemment, la téléphonie classique présente une disponibilité dite aux 5 « neuf », c'est-à-dire que le système est en état de marche 99,999 % du temps, ce qui représente cinq minutes de panne au total sur l'année. Un bon FAI travaille aux 3 « neuf », c'est-à-dire que son réseau est disponible 99,9 % du temps, ce qui équivaut à 8,8 heures de panne par an.

Beaucoup de FAI ne parviennent à un état de marche que de 99 %, qui équivaut à trois journées de panne par an. Le coût pour grimper d'un facteur « neuf » est approximativement de 2. Plus le nombre de « neuf » augmente, plus le coût est élevé. En gardant ce facteur 2 à l'esprit, on constate que le coût pour passer d'une disponibilité ordinaire à la disponibilité classique de la téléphonie exige une multiplication du coût par presque 10.

Il est possible d'implémenter une redondance de tous les éléments, réseau d'entreprise, accès, cœur de réseau, etc. Cette redondance est plus ou moins nécessaire suivant le cœur de métier de l'entreprise.

Pour une entreprise ou un opérateur, il est important de déterminer le taux de disponibilité visé en fonction de l'objectif à atteindre. De nombreuses solutions sont disponibles pour gagner en disponibilité, à condition que l'ensemble de la chaîne soit mis au niveau.

Pour la boucle locale, la solution optimale est de pouvoir être connecté avec une ligne professionnelle à deux opérateurs simultanément. Une ligne professionnelle apporte déjà une garantie importante du fait d'un service de prise en charge des problèmes en temps quasi réel. Les pannes peuvent néanmoins durer plusieurs minutes, ce qui équivaut à une disponibilité de 4 « neuf ». Cette disponibilité peut être augmentée en utilisant deux connexions simultanées au même opérateur si le nombre de paires métalliques est, par exemple, de huit. Le mieux est encore d'avoir des opérateurs différents afin de contourner les risques de panne d'un même concentrateur DSLAM.

Les opérateurs augmentent pour leur part la disponibilité de leur liaison entre le DSLAM et le routeur de bord de leur réseau cœur en mettant en place non pas une liaison mais deux liaisons sur deux routeurs de bord différents.

La dernière partie qui peut être fiabilisée est le réseau cœur lui-même. Une redondance des chemins est généralement utilisée, avec de nombreuses stratégies possibles. Ces stratégies dites  $N:M$  impliquent que  $N$  chemins sont protégés par  $M$  chemins supplémentaires. Le cas le plus classique est le 1:1, dans lequel un chemin possède un chemin de backup. Il est toutefois aujourd'hui démontré que cette stratégie n'est pas la meilleure, en particulier pour la protection des liaisons portant de la téléphonie, car trop onéreuse.

Pour atteindre des disponibilités aux 5 « neuf » tout en gardant des coûts acceptables, une redondance 4:2, voire 5:2, est préférable. Les deux lignes de backup peuvent d'ailleurs être utilisées par des applications moins sensibles, qui peuvent être supprimées ou dégradées si la ligne de sauvegarde doit être récupérée pour des liaisons téléphoniques en panne.

Les calculs permettant de définir le type de redondance à appliquer s'effectuent à partir des valeurs MTTF (Mean Time To Failure) et MTTR (Mean Time To Repair). Pour donner des ordres de grandeur des MTTR et MTTF utilisés pour réaliser des calculs de disponibilité au États-Unis, Telcordia a opté pour une valeur de MTTR de 2 h pour les routeurs et commutateurs et de 12 h pour une coupure de ligne passant par un domaine public, une panne de transmetteur optique tous les 3 800 jours et une panne d'un récepteur optique tous les 8 000 jours.

## La gestion

Avec la ToIP, il faut inclure les flux de parole dans le système de gestion du réseau, en particulier pour les pannes et la comptabilité.

Le protocole classique SNMP est parfaitement utilisable dans ce contexte, mais des systèmes dits d'« hypervision » doivent être adaptés pour que la détection des pannes puisse se faire dans des laps de temps suffisamment courts pour ne pas compromettre la disponibilité du système.

De même, des dispositifs de configuration automatique des routeurs peuvent être mis en place afin de configurer les conditionneurs de trafic à l'intérieur des équipements de réseau pour prendre en charge les trafics de téléphonie. Ces flux sont mis en classe EF (Expedited Forwarding), selon la technique DiffServ normalisée par l'IETF (*voir le chapitre 7*), ce qui demande une configuration spécifique des routeurs.

Une solution de configuration s'est fortement développée depuis quelques années grâce à la technique PBM (Policy-based Management), qui permet à un système central de configurer les équipements de réseau par politique.

Le contrôle par politique nécessite plusieurs composants (*voir figure 15.7*). Les nœuds du réseau prennent le nom de PEP (Policy Enforcement Point). Les politiques y sont appliquées pour gérer les flux des utilisateurs. Le PDP (Policy Decision Point) est le point qui prend les décisions et choisit les politiques à appliquer aux PEP. La communication entre le PEP et le PDP s'effectue par le biais d'un protocole *ad hoc* choisi entre plusieurs possibilités. Aujourd'hui c'est le protocole NetConf qui semble le plus utilisé.

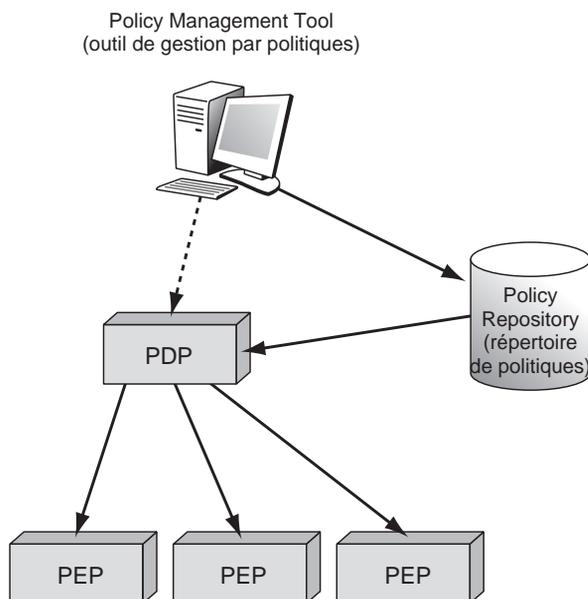
Le système comporte également une console utilisateur, qui contient les outils de gestion des politiques. Ces outils permettent notamment d'entrer les politiques dans une base de données, nommée Policy Repository, qui entrepose les règles de politique que le PDP vient rechercher pour les appliquer aux nœuds du réseau.

Des variantes de ce schéma de base peuvent inclure plusieurs PDP susceptibles de gérer un même nœud de transfert du réseau. Dans ce cas, les PDP ont des rôles différents. Une autre variante autorise une décentralisation des fonctions du PDP dans des PDP locaux, appelés LPDP (Local Policy Decision Point). En règle générale, un PDP gère un seul domaine administratif, et les règles de politique sont communes à la configuration de l'ensemble des nœuds du domaine.

Le PDP est défini comme une entité logique prenant des décisions politiques pour elle-même ou pour d'autres éléments réseau qui réclament ses décisions. Le PDP, que l'on peut aussi appeler serveur de politiques, est donc le point central qui doit décider des politiques à appliquer dans le réseau. Il s'agit en quelque sorte d'un organe de décision, qui recherche les informations dont il a besoin dans de nombreux serveurs communiquant directement avec lui de façon à prendre une décision. Ces serveurs peuvent être locaux, ce qui est le cas le plus général, mais ils peuvent aussi être distants.

**Figure 15.7**

*Architecture d'un système géré par politiques*



Les PEP sont aussi des entités logiques qui appliquent les décisions politiques prises par le PDP dont elles dépendent. Les PEP sont généralement les nœuds du réseau, qui peuvent être de différents types : routeur, commutateur ou LSR (Label Switched Router). Un PEP peut également être un pare-feu ou un équipement intermédiaire entre le client et le réseau. Le client peut lui-même posséder un client PEP sur son terminal.

Dans les réseaux de mobiles à carte à puce, il est fréquent de considérer qu'un PEP d'accès se trouve sur la carte à puce. Son rôle est essentiellement réduit à la gestion de la sécurité et non de la qualité de service, mais il est imaginable d'implémenter la gestion de QoS dans les cartes à puce dès que celles-ci seront assez puissantes pour l'implémenter.

La technique PBM (Policy-based Management) ne peut cependant être considérée comme une solution universelle, car elle dépend aujourd'hui fortement des équipementiers et de leur interface de configuration de leurs routeurs et commutateurs. L'avantage du standard NetConf, permettant les communications entre le PDP et les PEP, est d'utiliser XML. Cette propriété le rend relativement indépendant des langages de configuration. Les routeurs et commutateurs doivent toutefois être munis d'un interpréteur pour générer le code complet de configuration.

Ce mode de gestion automatisé par politiques ne concerne pour l'heure que les réseaux d'opérateurs. Pour la gestion de l'application de ToIP elle-même, des systèmes de gestion plus classiques permettent d'implémenter la surveillance des pannes, la planification, la sécurité, les performances et la comptabilité.

## Le contrôle

Le contrôle complète la gestion du réseau afin d'optimiser la configuration du réseau pour que les contraintes de la téléphonie soient satisfaites. On distingue la gestion du contrôle par les temps de réaction et la mise en œuvre dans le contrôle d'une boucle de réaction rapide.

Le contrôle exige une boucle de rétroaction capable de modifier la configuration du système si les temps de réponse deviennent inacceptables pour la téléphonie. Les premières solutions consistaient à contrôler la congestion afin que les flux restent fluides à l'intérieur du réseau. Cela revenait à maintenir le taux d'utilisation des ressources suffisamment bas pour qu'il n'y ait pas de files d'attente importantes dans les nœuds.

De nouvelles architectures, dites autonomes (*autonomic*) ont été proposées, évaluées puis réalisées. La figure 15.8 illustre l'apport principal de cette nouvelle architecture, qui réside dans l'introduction d'un « plan de connaissance » rassemblant les informations contextuelles afin de les mettre à disposition des algorithmes de contrôle présents dans le plan de contrôle. L'avantage de cette solution est de rassembler dans un plan unique les connaissances nécessaires au pilotage du réseau, notamment au pilotage des algorithmes de contrôle des flux de ToIP pour que ceux-ci passent sans problème.

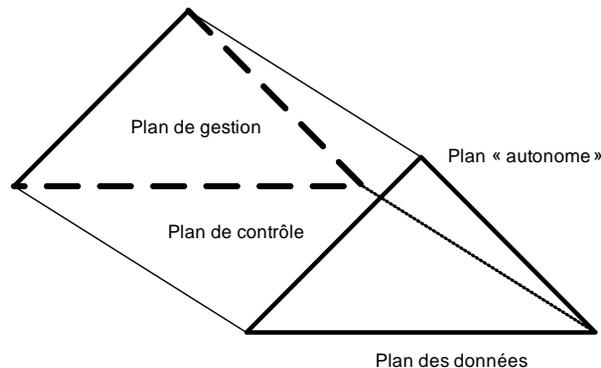


Figure 15.8

Architecture d'un environnement de contrôle « autonome »

Cette architecture décrit un ensemble de quatre plans :

- Le plan des données contient les couches basses de l'architecture des réseaux, c'est-à-dire les couches 1 à 4 du modèle de référence ou l'environnement TCP/IP.
- Le plan de contrôle contient les algorithmes de contrôle du plan des données. On y trouve les algorithmes de routage de contrôle de flux, de contrôle de la qualité de service, de la sécurité, de la mobilité, etc. L'objectif de ces algorithmes est de faire fonctionner le plan des données d'une façon plus ou moins statique.

- Le plan « autonome » a pour objectif, d'une part, de mettre en place un algorithme de pilotage et, d'autre part, de mettre à la disposition du système de pilotage des connaissances qui seront nécessaires pour alimenter les algorithmes de contrôle.
- Le plan de gestion a pour objectif d'administrer les trois autres plans.

La nouveauté de cet environnement provient du plan « autonome », qui s'occupe de rechercher les informations et de les traiter (ce qu'on appelle les « connaissances »), et du système de pilotage, qui se nourrit des connaissances et qui en fait profiter les algorithmes de contrôle qui auront été choisis et alimentés de façon adéquate.

## La qualité de service

La qualité de service est particulièrement importante dans l'application temps réel qu'est la ToIP. Comme nous l'avons vu aux chapitres 6 et 7, elle peut être obtenue par différentes méthodes, suivant que le réseau est commuté ou routé.

Si le réseau est commuté, les chemins utilisés pour le transport des paquets peuvent être dimensionnés par des techniques d'ingénierie de trafic. Si le réseau est routé, des solutions comme DiffServ permettent de classer le trafic et de surdimensionner le réseau par rapport au trafic prioritaire constitué essentiellement des paquets de parole.

Lorsque des techniques de commutation sont utilisées, il s'agit généralement de réseaux MPLS (MultiProtocol Label-Switching). Ces réseaux concernent plutôt les très grandes entreprises et les opérateurs. Dans ce cas, des chemins sont tracés dans le réseau. Associées à ces chemins, des qualités de service suivent le plus souvent la définition DiffServ.

Les chemins de plus haute priorité, EF (Expedited Forwarding), garantissent la qualité de service grâce à un calcul de trafic réalisé lors de l'ouverture de la connexion par le paquet de signalisation. La réservation de ressources effectuée permet de garantir les temps de transit nécessaires à la ToIP. Parfois, certains opérateurs préfèrent ouvrir plusieurs chemins avec la qualité EF afin de différencier de façon encore plus précise la ToIP d'applications métier qui se trouveraient dans la même classe. Dans ce cas, la garantie est encore plus forte.

Dans le cas d'un réseau de routage, que l'on trouve plutôt dans les petites, moyennes et grandes entreprises, la qualité de service est assurée par la classe EF, mais la garantie est apportée non par une réservation de ressources mais par un surdimensionnement du réseau par rapport à la quantité de trafic EF. Ce calcul se complexifiant en fonction de la taille de l'entreprise, on passe à des techniques de réservation sur des chemins dès que le trafic global devient trop élevé.

Le calcul s'effectue sur les clients de la classe EF, qui provient de la ToIP et de certains trafics métier. Il faut réserver, pour ne pas avoir de surprise dans la suite, 100 Kbit/s par voie téléphonique, qu'elle soit compressée ou non. En effet, nous avons vu que la taille minimale de la trame Ethernet était de 64 octets ; si la parole est très compressée, il n'y aura que peu d'octets téléphoniques dans la trame, et si la parole n'est pas compressée, il n'y aura pratiquement que des octets téléphoniques dans la trame.

Il faut faire attention au cas des réseaux Ethernet à 1 Gbit/s, dont les trames ont une longueur minimale de 512 octets et dans lesquelles une voie de parole peut représenter jusqu'à 500 Kbit/s si aucun élément ne permet de multiplexer plusieurs voies de ToIP dans une même trame.

Ajoutons, pour conclure sur la qualité de service, que le protocole RTP/RTCP est une solution encore très utilisée aujourd'hui pour compresser plus ou moins le flot téléphonique en fonction de l'état du réseau. L'application s'adapte ainsi au réseau, alors que les solutions plus modernes visent à ce que le réseau s'adapte à l'application.

## GAN

GAN (Generic Access Network), anciennement UMA (Unlicensed Mobile Access), désigne une architecture de contrôle des réseaux permettant de passer d'un réseau disposant d'une couverture locale (un Local Area Network, ou LAN) vers un réseau disposant d'une couverture plus large (un Wide Area Network, ou WAN).

Le modèle GAN s'inscrit dans l'objectif de convergence entre les réseaux fixes et mobiles, qu'on appelle CFM (convergence fixes mobiles), en anglais FMC (Fixed Mobile Convergence). GAN permet de réaliser un basculement instantané (appelé hand-over, si l'on reste dans le réseau de l'opérateur, ou roaming, si l'on passe sur le réseau d'un autre opérateur) d'un réseau IP vers un réseau non-IP, et inversement, sans rupture de la communication.

Par exemple, l'utilisateur dispose d'un réseau d'accès IP sans licence (Wi-Fi, Bluetooth, ou autre), qu'on appelle UMA. En s'y connectant, il peut accéder à une passerelle connectée au réseau IP, appelée UMA Network Controller. Cette passerelle lui donne accès au réseau téléphonique de son opérateur.

Lorsque l'utilisateur le désire, et sans qu'il y ait de coupure dans la communication en cours, il peut s'affranchir de son réseau d'accès UMA pour passer à un réseau avec licence, ou LMA (Licenced Mobile Acces), de type GSM, qui dispose d'une couverture géographique plus large, mais coûte plus cher.

Ainsi, l'utilisateur est libre d'exploiter deux supports d'accès pour accéder à ses services : un réseau IP, sans licence, ou bien un réseau non-IP, avec licence. Bien sûr, cela suppose que le terminal de l'utilisateur soit bi-mode, c'est-à-dire qu'il supporte à la fois les connexions IP et les connexions mobiles.

Plusieurs opérateurs se sont d'ores et déjà lancés dans l'exploitation du GAN, comme Orange, avec son offre UNIK, ou BT (British Telecom) avec Fusion.

En exploitant son réseau personnel IP comme réseau d'accès, l'utilisateur fait économiser à son opérateur l'usage d'une infrastructure de réseau d'accès mobile de type GSM, qui est bien plus coûteuse, ce qui s'en ressent dans sa facture. La fourniture de ce service est donc moins chère mais elle est aussi plus réduite, car l'utilisateur est restreint aux seules zones de couverture IP auxquelles il a accès.

La combinaison des deux supports, IP et non-IP, constitue un ensemble cohérent pour assurer la connectivité de l'utilisateur à tout moment, en tout lieu et dans les meilleures conditions.

## Perpectives

Dans les principaux pays développés, la ToIP se développe à une vitesse accélérée, et plusieurs d'entre eux, parmi lesquels le Japon, la Corée, la Finlande et la France, ont déjà annoncé leur passage total à la ToIP pour 2010 ou 2011.

Aujourd'hui, les taux de pénétration de la ToIP sont de l'ordre de 50 %, avec une croissance de 15 % par an. Les différentes applications de ToIP d'entreprise, grand public et d'opérateurs croissent de concert à des vitesses assez similaires.

Une des questions posées par cet essor fulgurant de la ToIP est le manque à gagner pour les opérateurs. Les communications téléphoniques vers tous les grands pays développés sont aujourd'hui quasiment gratuites. La compétition est telle pour attirer les clients que les gains, qui furent si importants dans la téléphonie classique, tendent vers zéro.

Les bénéfices doivent donc s'obtenir par d'autres services, éventuellement associés à la parole téléphonique, ou par une meilleure productivité. Dans ces conditions, les opérateurs doivent s'adapter et offrir de plus en plus de services pour pallier la disparition de la téléphonie de type circuit.

Les questions qui se posent sont encore nombreuses entre l'adoption d'une ToIP d'opérateur ou de type Skype, Wengo ou Vonage, ces dernières présentant l'avantage d'être gratuites, mais au prix d'une qualité de communication souvent insuffisante pour des usages professionnels. La qualité de service s'améliore petit à petit à mesure qu'apparaissent des réseaux plus performants.

De nombreux progrès pourraient être faits dans le domaine professionnel, mais les entreprises rechignent devant les coûts de déploiement. En particulier, la mise en place d'une politique de sécurité saine et la mise à niveau des équipements de routage et de commutation apportant une qualité de service suffisante aux flux de téléphonie, ajoutées à l'investissement dans des téléphones IP de qualité, impliquent des coûts élevés, auxquels n'ont pas encore consenti les entreprises.

Les problèmes techniques qui restent à résoudre sont nombreux, en particulier l'intégration des environnements fixes et mobiles avec une facture unique, quel que soit le terminal utilisé. La génération UMA, poussée par de nombreux opérateurs pour réaliser les handovers verticaux, n'est pas bien adaptée au monde IP, devenu majoritaire. Avec cette technologie, un réseau Wi-Fi se comporte comme s'il était en GSM, ce qui ne va guère dans le sens de l'histoire.

Le protocole IP devrait sortir vainqueur et s'imposer comme l'architecture native du monde aussi bien fixe que mobile. L'IMS (IP Multimedia Subsystem), solution de

rechange à l'architecture UMA pour l'intégration fixe-mobile, se présente sous de bien meilleurs auspices puisque ses protocoles de base sont IP et SIP (Session Initiation Protocol). Cette solution devrait s'imposer vers la fin de la décennie pour permettre une convergence relevant davantage de l'univers Internet que de celui des télécommunications. C'est l'objet du prochain chapitre.

## Conclusion

Globalement, le passage d'une situation où plusieurs réseaux portent chacun un média spécifique à un réseau unique multiplexant les médias a grandement complexifié les solutions. Comme nous l'avons vu dans ce chapitre, il existe des solutions à cette complexité, qui commencent à se répandre dans le domaine de la ToIP. Les freins à leur plein déploiement ne sont plus que le montant de l'investissement de départ, qui peut vite se révéler important pour de grands sites, et la disponibilité de téléphones IP permettant une qualité de service, ce que les softphones n'offrent pas.

L'avenir de la ToIP est tout tracé. Lente à mûrir, elle s'impose dorénavant partout, aussi bien chez les particuliers que chez les opérateurs. En se projetant plus loin dans le temps, elle va probablement se fondre dans les environnements multimédias et s'enrichir de l'image et de services évolués, la vidéo sur IP ayant des contraintes semblables en cas d'interactivité, mais à un débit beaucoup plus élevé.

Un autre axe de développement sera apporté par la téléphonie en tout lieu et à tout moment, qui permettra d'entrer en contact avec n'importe qui, où qu'il soit. Conçue au départ pour remplacer la téléphonie terrestre, la ToIP est aujourd'hui parfaitement accessible depuis un portable ou un PDA, voire un SmartPhone en y intégrant une application de téléphonie telle que Skype. Le trafic de téléphonie utilise alors une voie de données intégrée dans le transport de données, que ce soit sur un réseau Wi-Fi ou WiMax ou sur un canal de données à une vitesse suffisante dans un réseau de mobiles. Si la couverture n'est pas assurée par une des technologies disponibles, le satellite peut prendre le relais, toujours par l'intermédiaire d'un canal de données.

D'autres applications de parole téléphonique possédant des contraintes plus fortes passeront à terme en ToIP. Citons, à titre d'exemple, les applications d'urgences ou celles correspondant à des services en milieu difficile, comme la téléphonie entre une tour de contrôle et les avions, qui ne peut être coupée et dont la qualité doit être suffisante pour une bonne compréhension. La disponibilité doit alors atteindre les 6 « neuf » et la qualité être toujours au minimum celle de la téléphonie classique.

# 16

## L'architecture IMS

---

Non content de s'être imposé dans les réseaux filaires, le protocole IP continue sa progression et son déploiement et commence à tisser sa toile dans les réseaux cellulaires, au détriment de la téléphonie fixe traditionnelle. Rien ne semble pouvoir freiner son inexorable et prodigieuse ascension.

Pour que le protocole IP gagne et s'impose définitivement face au modèle à commutation de circuits, il semble que ce ne soit qu'une affaire de temps, et d'évolution des mœurs. Encore trop souvent, les utilisateurs rechignent à tenter le pari d'un tout-IP. Progressivement, la pratique, les coûts et les fonctionnalités nouvelles font cependant tomber les dernières réticences vers un monde IP.

Cela étant, cette solution de rechange à la téléphonie standard n'est qu'un premier pas vers la révolution des télécommunications qui se met en place sous nos yeux. En effet, maintenant que les technologies IP sont éprouvées et démontrent qu'elles sont suffisamment fiables et matures, l'étape suivante consiste à faire converger tous les réseaux d'accès (IP ou télécoms) vers une architecture unique.

Le monde IP ne se limite pas à remplacer la téléphonie traditionnelle : il l'enrichit d'une gamme de services évolués, diversifiés et à moindre coût, qui a de quoi inspirer le monde des télécoms. Par exemple, une conversation sur Internet peut, dynamiquement et très facilement, faire usage de la voix, de la vidéo, de textes, de partages de documents (textes, images, sons et vidéos), d'applications en réseau, de sécurisation des communications, etc.

C'est ce marché du possible, déjà conquis dans le monde Internet, mais qui échappe encore au monde des télécoms, que vient combler la nouvelle architecture appelée IMS (IP Multimedia Subsystem).

L'architecture IMS marque le développement d'un nouveau type de réseau qui s'inscrit à la croisée d'IP et des télécoms. Comme son nom l'indique, IMS définit un sous-réseau multimédia ou plutôt un réseau parallèle aux réseaux actuels. Son ambition est le traitement des flux et services multimédias dans une optique de convergence, quel que soit le réseau d'accès utilisé (GSM, Wi-Fi, Ethernet, UMTS, WiMAX, etc.), quelle que soit sa nature (fixe ou mobile), et quel que soit le type de terminal considéré (ordinateur, PDA, téléphone, etc.).

Cette plate-forme révolutionne le monde des télécoms et accélère la profonde mutation de ce secteur en ce qu'elle exploite le protocole IP pour gérer les communications entre les entités mises en place. La téléphonie n'est plus *le* service, mais *un* service, parmi beaucoup d'autres.

L'objectif de ce chapitre est de présenter l'IMS en détaillant les motivations qui conduisent les acteurs des télécoms à faire évoluer leur réseau vers IP, puis en décrivant historiquement le développement de l'IMS. Nous présenterons ensuite l'architecture que propose IMS en introduisant ses composants, avant d'entrer dans le détail des communications.

Nous concluons ce chapitre en évoquant les obstacles qui restent à franchir pour l'adoption généralisée d'une architecture IMS.

## Motivations et objectifs de l'IMS

C'est presque sous la forme d'un scénario publicitaire que l'on pourrait présenter l'objectif de l'IMS. Avec son téléphone comment faire pour :

- disposer d'un service de présence, c'est-à-dire connaître la disponibilité de ses contacts avant même qu'on ne les appelle ?
- envoyer une photo, un numéro de téléphone ou une adresse postale durant une communication ?
- faire un achat en saisissant son numéro de Carte bleue de manière sécurisée ?
- recevoir le bilan financier de son comptable pendant qu'on discute avec ce dernier ?
- accéder à l'intranet de son entreprise ?
- regarder une bande-annonce de film durant la réservation d'une place de cinéma ?
- initier une conférence audio, puis, pendant la communication, basculer en mode conférence vidéo, le tout en faisant varier dynamiquement le nombre de participants ?
- modifier une présentation pendant que son correspondant la diffuse et la commente par téléphone ?

Les pistes pour certaines de ces actions ne manquent pas. Mais le constat général est flagrant : sur Internet, toutes ces tâches sont triviales, interactives et sans coupure, alors que, avec un téléphone, elles relèvent de l'exploit, coûtent relativement cher et, surtout, sont tout sauf simples à mettre en œuvre.

Certes, le terminal y est pour beaucoup, mais, de ce côté-là, les utilisateurs ont de plus en plus la possibilité de s'équiper de terminaux très évolués et très puissants, à la frontière des ordinateurs. À des coûts progressivement raisonnables, les ventes de PdaPhones et SmartPhones (avec en particulier le succès de l'iPhone) corroborent cet attrait de la part des utilisateurs pour des appareils de plus en plus sophistiqués.

Manifestement, le problème ne se situe pas au niveau des terminaux, puisque l'équipement existe et qu'il se démocratise rapidement. Si les services télécoms restent à développer, l'objectif est de les placer à la hauteur de ceux disponibles dans le monde IP. Pour cela, l'idée de l'IMS est d'installer le cœur du réseau télécoms dans le monde IP, afin de lui faire partager ses acquis et de permettre le développement rapide de nouveaux services universels, accessibles quel que soit l'endroit où l'on se trouve, quel que soit le réseau que l'on exploite et quel que soit le terminal que l'on utilise.

### ***Le point de vue opérateur***

#### **Le problème : plusieurs réseaux d'accès, mais des besoins identiques partout**

Considérons les réseaux RTC (pour la téléphonie traditionnelle sans IP), DSL (pour la téléphonie IP) et GSM (pour la téléphonie radio).

Ces trois réseaux sont spécifiques, et surtout autonomes. Ils n'interagissent pas entre eux. Dans ce cadre, développer un nouveau service réseau revient inévitablement à le penser de manière exclusive pour l'un de ces réseaux.

Pourtant, ces trois réseaux sont susceptibles de fournir les mêmes services, à commencer par la téléphonie, qui est disponible dans chacun d'eux, mais avec une architecture différente, des serveurs différents et un mode de fonctionnement différent et incompatible, alors même que le service est identique.

#### **La solution : mutualiser les ressources pour réduire les coûts et simplifier la gestion des infrastructures des opérateurs**

Aujourd'hui, les opérateurs mettent en place une infrastructure distincte pour chaque réseau d'accès. Ainsi, on trouve un serveur d'authentification pour les accès dans les réseaux RTC et un autre pour les réseaux GSM. L'opérateur doit donc maintenir et gérer ces deux serveurs. On peut faire le même constat pour les serveurs de facturation, ou plus largement les serveurs d'applications, qui sont propres à un certain type de réseau d'accès.

On comprend facilement la motivation à l'origine de ce phénomène. Les réseaux d'accès ont été conçus indépendamment les uns des autres, et ils ont connu un développement parallèle, de sorte qu'il était difficile d'imaginer leur unification. La question n'en demeure pas moins : pourquoi continuer à maintenir deux architectures qui remplissent les mêmes fonctions ?

L'architecture IMS vise à déployer une plate-forme dans laquelle chaque service peut être géré par un seul serveur, et ce quel que soit le réseau d'accès. La vision de l'IMS consiste à affirmer haut et fort que les réseaux d'accès, s'ils diffèrent, ne sont jamais que des points d'entrée et de sortie permettant de joindre les terminaux des correspondants. Au centre, le réseau cœur est différent, alors que rien ne le justifie. L'idée est de faire converger tous les réseaux d'accès vers un même réseau cœur, et donc, quel que soit le réseau d'accès utilisé, de disposer des mêmes couches de transport de l'information, de contrôle des flux et d'accès aux services.

La vision d'opérateur verticale, dans laquelle les composants sont monolithiques et indissociables, est remplacée par une vision horizontale, dans laquelle les composants sont flexibles et modulaires. Les fonctions communes à l'ensemble des réseaux (comme l'authentification, la facturation et la fourniture des services) sont mutualisées dans des ressources partagées et centralisées.

### **Un service intelligent**

Mutualiser les ressources ne signifie pas perdre les particularités de chaque réseau d'accès.

Par exemple, si une requête parvient à un serveur, ce dernier ne répondra pas de la même manière selon que le terminal du client est un téléphone mobile ou un PC. Il doit impérativement faire la distinction entre les réseaux d'accès pour adapter ses réponses et fournir des résultats pertinents, par exemple selon le débit dont dispose l'utilisateur.

## ***Le point de vue utilisateur***

### **Le problème : des scénarios d'utilisation différents mais des besoins analogues**

Aujourd'hui, l'utilisateur a autant de terminaux que de profils d'utilisateur. Or, pour un même terminal, il n'est pas possible de faire la distinction entre une ligne professionnelle et une ligne privée. Il est donc nécessaire d'avoir deux terminaux pour simplement différencier les paramètres d'utilisation, alors même que, fonctionnellement, un seul terminal téléphonique suffirait.

L'utilisateur a de surcroît souvent besoin d'avoir dans les deux terminaux des paramètres communs, comme ses contacts ou son calendrier, dont il peut avoir besoin à tout moment, ce qui l'oblige à multiplier les manipulations fastidieuses et redondantes sur des appareils qui, bien souvent, se configurent différemment.

### **La solution : un terminal unique**

Le même terminal pour la ligne fixe, à la maison comme au travail, et pour la ligne mobile, privée comme professionnelle : voilà ce que nous promet l'IMS.

Nos lignes téléphoniques, et tous les services qui y sont associés, nous suivent, sans qu'il soit besoin de multiplier les terminaux, ce qui permet d'économiser et de simplifier considérablement les usages.

## Historique de l'IMS

Puisque l'IMS préfigure l'architecture des réseaux de demain, il était logique que son développement s'appuie sur les différents réseaux existants et que sa constitution tienne compte des différentes technologies en s'imprégnant des tendances en cours, des besoins et des recommandations.

Au départ, la question était de savoir comment faire évoluer les réseaux cellulaires, c'est-à-dire de déterminer les protocoles et architectures des futurs réseaux de télécommunications. Le problème est que, suivant l'endroit où l'on se trouve, les standards utilisés dans les réseaux cellulaires sont différents. Par exemple, en Europe la norme évolue du GSM vers l'UMTS tandis qu'en Amérique du Nord et partiellement en Asie, les produits s'orientent vers le CDMA 2000. Il s'agit donc de prévoir des évolutions vers des standards distincts.

Pour cela, deux groupes de travaux ont été formés, chacun dévolu à l'une de ces normes : le 3GPP (Third Génération Partnership Project), en charge de réfléchir aux évolutions de la norme GSM, et le 3GPP2 (Third Génération Partnership Project 2), en charge de réfléchir aux évolutions de la norme CDMA 2000. Tous deux font partie de l'organisme de normalisation international IMT-2000 (International Mobile Telecommunications 2000), chargé par l'UIT (Union internationale des télécommunications) de définir et de standardiser les réseaux de mobiles de la prochaine génération.

Créé en 1998, le 3GPP regroupe différents partenaires régionaux de la standardisation des télécommunications : l'ETSI (European Telecommunications Standards Institute) en Europe, le Comité T1 aux États-Unis, le CCSA (China Communications Standards Association) en Chine, le TTC (Telecommunications Technology Committee) et l'ARIB (Association of Radio Industries and Business) au Japon et le TTA (Telecommunications Technology Association) en Corée. À ces partenaires, se sont ajoutés d'autres organismes et grands groupes industriels qui s'y sont affiliés pour réfléchir à des projets communs (UMTS Forum, IPv6 Forum, etc.).

Le 3GPP ne référence pas ses documents de travail sous le nom de standards, mais sous celui de spécifications techniques, ou TS (Technical Specifications), et de rapports techniques, ou TR (Technical Reports). Lorsqu'un ensemble de TS et de TR est complet et cohérent pour former un tout décrivant exhaustivement les fonctionnalités attendues, l'ensemble est classé sous le nom de Release. L'IMS a été défini dans la Release numéro 5 du 3GPP.

En 2004, quatre groupes de travaux de l'ETSI, nommés SPAN, Signaling System N°7, R1 et Typhon, ont fusionné pour donner naissance au groupe de travail TISPAN (Telecommunications and Internet converged Services and Protocols for Advanced Networking), qui œuvre dans le cadre du 3GPP pour l'IMS.

En 2005, TISPAN a proposé une extension importante de l'IMS à destination des réseaux filaires, lesquels, avant cela, n'étaient pas concernés par l'IMS. La première version de l'IMS adoptée par l'ETSI a été proposée cette même année. Ces travaux ont été repris et incorporés dans l'IMS dès 2006, avec la publication de la Release 6 de

l'IMS. La deuxième version a été proposée en 2007 (ajout de la VoD, d'IPTV et de la QoS). La troisième (ajout de la mobilité généralisée et de la convergence) verra le jour en 2009.

L'initiative 3GPP2 a été lancée en 1998 comme un projet collaboratif pour la conception d'un réseau de troisième génération partant sur la base des réseaux cdma2000 mis en place dans les réseaux cellulaires d'Amérique du Nord et d'Asie. Il regroupe cinq partenaires régionaux : TIA (Telecommunications Industry Association) en Amérique du Nord et les groupes ARIB, CCSA, TTA et TTC déjà cités. À ces membres se sont joints des partenaires commerciaux et des industriels.

Bien que les réseaux considérés au départ soient différents, ces deux instances de l'UIT ont les mêmes objectifs de convergence des communications vers un réseau tout-IP. Pour s'orienter vers un modèle IP, et plutôt que de reconstruire des protocoles, le 3GPP et le 3GPP2 ont exploité ce qui existait déjà et y ont ajouté ce qui devenait nécessaire. Ils travaillèrent pour cela de concert avec l'IETF afin de standardiser les protocoles utilisés et d'y apporter des modifications : ces travaux concernaient en particulier les protocoles SIP, COPS, Diameter, DNS et IPv6.

Au final les deux organismes, 3GPP et 3GPP2, proposent deux versions différentes, quoique relativement proches, de l'IMS.

Parallèlement au développement de l'architecture IMS, un autre groupe, créé en 2002 et nommé OMA (Open Mobile Alliance), s'est impliqué dans le développement de l'IMS pour la définition de services plus évolués que ceux prévus par le 3GPP et le 3GPP2.

## Introduction à l'IMS

Cette section présente des généralités qu'il est important d'avoir à l'esprit avant d'entrer dans l'architecture de l'IMS.

### *Protocoles utilisés dans l'IMS*

Les trois principaux protocoles utilisés dans l'IMS sont les suivants :

SIP est le protocole fédérateur de l'architecture IMS. Il est en quelque sorte la glue qui permet aux différents composants de communiquer entre eux de manière homogène.

Son choix, par rapport à tout autre protocole de signalisation, n'est pas anodin puisqu'il pérennise SIP au détriment de H.323, jugé trop lourd et trop coûteux. Il marque ainsi la confiance des industriels et du monde de la recherche dans le protocole SIP.

Bien sûr, son utilisation implique celle des protocoles associés, en particulier SDP, pour la description des sessions, et RTP/RTCP, pour le transport en temps réel des flux de données multimédias.

Diameter, décrit dans la RFC 3588 de l'IETF, est un protocole de type AAA (Authentication, Authorization, Accounting). Successeur du protocole RADIUS (Remote Authentication Dial-In User Service), son nom est un jeu de mots visant à symboliser l'évolution de la technologie puisque le diamètre (en anglais *diameter*) est le double du rayon (en anglais *radius*). Il est employé pour la sécurisation des communications, notamment lors de l'enregistrement des utilisateurs et lorsque les profils utilisateur sont transférés d'une base de données vers les serveurs de traitement (nous décrivons plus loin ces entités). Il fonctionne en mode client/serveur, donc sous la forme de requêtes et de réponses. Diameter est également présent dans IPsec et TLS.

COPS (Common Open Policy Service) est un protocole flexible permettant la mise en place de politiques par une entité centrale appelée PDP (Policy Decision Point), qui sont appliquées sous formes de règles dans des entités appelées PEP (Policy Enforcement Point). COPS sert notamment à permettre aux opérateurs de garantir une qualité de service dans une architecture IMS.

### **Notions de réseau visité et de roaming**

L'architecture IMS s'appuie sur des réseaux d'accès très étendus. Non seulement il est possible d'utiliser de multiples technologies d'accès, mais, en outre, les zones d'accès géographiques doivent être aussi larges que possible dans la vision IMS. Autrement dit, le terminal IMS de l'utilisateur — on parle également de UE (User Equipment) — doit pouvoir se connecter de partout.

Un opérateur possédant une infrastructure physique forcément limitée, il convient de prendre en compte la possibilité d'exploiter des réseaux tiers, c'est-à-dire de permettre à un abonné d'un opérateur d'utiliser des réseaux d'accès physiques qui appartiennent à d'autres opérateurs que le sien.

Pour cela, on introduit le concept de réseau visité : lorsqu'un utilisateur utilise un réseau qui n'est pas celui de l'opérateur auquel il a souscrit un contrat de services, on dit que le réseau en question est visité.

La connexion de l'utilisateur à un réseau visité est soumise aux accords qui lient son opérateur à l'opérateur qui exploite le réseau visité. En effet, pour qu'une connexion soit permise, si l'opérateur qui exploite un réseau ne peut pas identifier (et facturer) un utilisateur, il est indispensable qu'il existe des engagements contractuels avec un autre opérateur (que l'utilisateur devra mentionner lors de sa connexion).

On parle de *roaming* pour désigner la possibilité qu'a un utilisateur de se connecter à un réseau qui n'est pas celui de son opérateur. De même, les accords de roaming décrivent les conditions contractuelles qui lient un opérateur, exploitant une base de données de clients, à un opérateur, exploitant un réseau d'accès avec une couverture géographique.

## Le modèle à quatre couches de l'architecture IMS

IMS propose une approche modulaire qui permet de distinguer des niveaux de traitements différents.

Quatre couches (ou « plans ») peuvent être identifiées, chacune d'elles étant liée à un domaine spécifique (la figure 16.1 illustre un schéma global de l'architecture en couches de l'IMS) :

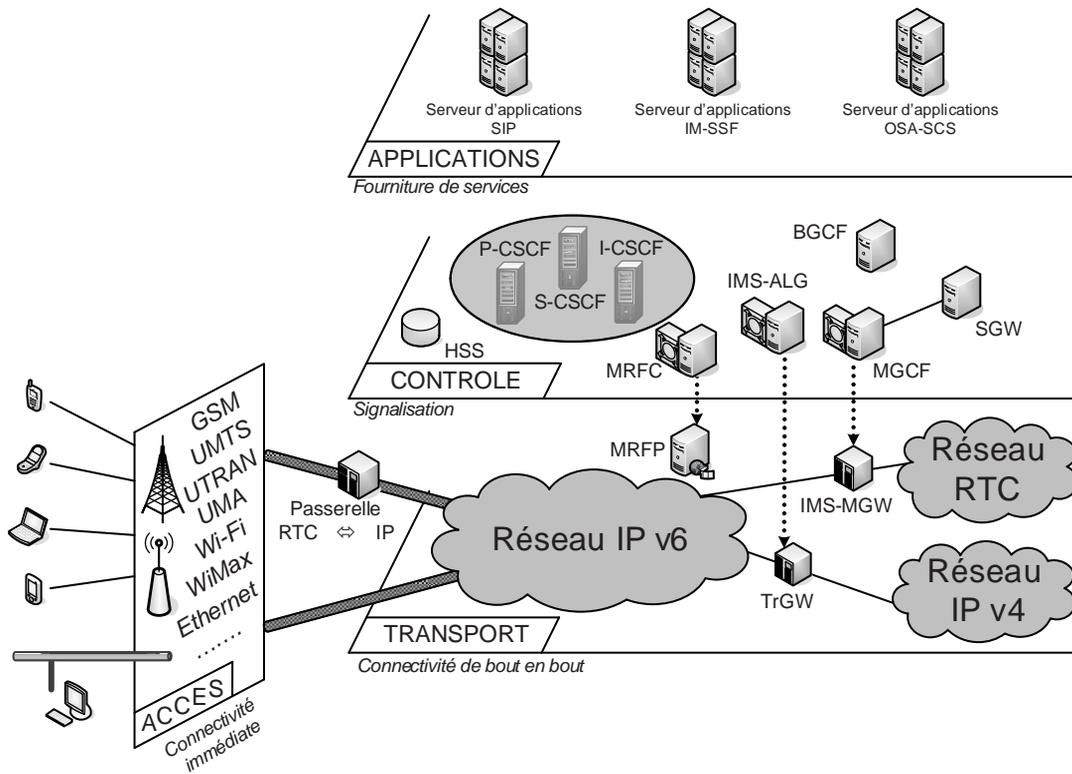


Figure 16.1

Architecture de l'IMS

- La couche accès.** Elle définit la manière dont l'utilisateur se connecte au réseau. Parmi les réseaux d'accès, on peut citer les plus classiques : GSM (Global System for Mobile communications), UMTS (Universal Mobile Telecommunications System), CDMA 2000 (Code Division Multiple Access 2000), UMA (Unlicensed Mobile Access), xDSL (xDigital Subscriber Line), Wi-Fi (Wireless Fidelity), WiMax (Worldwide Interoperability for Microwave Access) et les réseaux fixes, comme Ethernet, ATM, la fibre optique et le câble, par exemple. Cette liste non exhaustive est susceptible de contenir

n'importe quel réseau d'accès, qu'il soit filaire ou non. Remarquons que les réseaux supportés ne s'orientent ni vers IP, ni vers les réseaux cellulaires en particulier, mais visent une large étendue de possibilités.

En regroupant différents types de réseaux d'accès au sein de cette couche, IMS offre un niveau d'abstraction à la manière dont l'utilisateur se connecte au réseau. Cette vision est à l'origine de l'idée de convergence des réseaux vers un réseau unique, prônée par l'IMS. Autrement dit, IMS est indépendant du réseau d'accès, qui n'est qu'un élément assurant la connectivité de l'utilisateur au réseau cœur.

- **La couche transport.** Elle permet la connectivité de bout en bout entre les différents interlocuteurs. Alors que la couche d'accès se contente de connecter un utilisateur au réseau IMS, la couche transport se charge de l'acheminement des données de l'utilisateur jusqu'à son (ou ses) correspondant(s). Cela comprend le transport de l'information par les routeurs et le choix de la route empruntée dans le réseau. C'est le réseau IP qui est utilisé dans cette couche.
- **La couche contrôle.** Elle assure la gestion et le contrôle du réseau. Elle est en charge de tous les messages de signalisation dans le réseau, permettant d'ouvrir, de maintenir, de modifier et de terminer une session entre des utilisateurs. C'est la partie intelligente du modèle, qui offre toutes les fonctionnalités de gestion des utilisateurs et constitue le véritable socle de l'IMS. Nous la détaillons plus loin, en présentant l'ensemble des entités qui s'inscrivent dans cette architecture.
- **La couche application.** Elle consiste en la fourniture des services, qu'ils soient audio, vidéo ou textuels. Cette couche implémente tous les services que l'on peut proposer aux utilisateurs. Elle est la partie la plus ouverte du modèle, puisque le réseau IMS ne spécifie pas les services eux-mêmes, mais offre une plate-forme de déploiement unifiée, simple, rapide, productive et sécurisée pour la mise en place de nouveaux services. Sa description est assez sommaire, mais elle est susceptible d'être enrichie à l'infini, selon les nouveaux services que l'on souhaite apporter et toutes les propositions amenées par des développeurs. On peut, notamment, mentionner les applications classiques de présence, de messagerie instantanée et de push-to-talk. Nous verrons plus loin les différentes catégories de services que l'IMS permet de distinguer.

Chaque couche est indépendante, de sorte qu'il est possible, par exemple, d'ajouter librement de nouveaux services dans la couche applicative, sans tenir compte du réseau d'accès que les utilisateurs ont employé, ni du terminal qu'ils ont utilisé. Cela dit, il est souhaitable que les serveurs d'applications adaptent leur réponse en fonction de ces critères : par exemple, une page Web ne doit pas contenir les mêmes informations ni être formatée de la même manière selon qu'elle est destinée à être visualisée sur un ordinateur portable ou sur un téléphone portable.

### ***La couche de transport***

La couche de transport est une couche générique IP. Elle est formée d'un maillage de commutateurs et de routeurs qui assurent, dans le réseau IP, le routage des données multimédias, à l'exclusion des informations de signalisation, qui sont à la charge de la couche de contrôle, décrite plus loin.

Elle dispose en outre de deux sous-systèmes particuliers :

- NASS (Network Attachment SubSystem) pour la configuration du réseau. Il permet de disposer dans le réseau de l'équivalent d'un serveur DHCP, afin d'attribuer une adresse IP à l'utilisateur, et d'un client, pour les authentifications de l'utilisateur, réalisées sur son profil.
- RACS (Ressource Admission Control SubSystem) pour le contrôle d'admission. Il permet d'allouer les ressources sollicitées par les applications en effectuant, à la demande, des réservations.

### ***La couche de contrôle***

La couche de contrôle est constituée de différentes entités logiques que nous allons détailler.

Toutes ces entités constituent le socle de l'architecture IMS et, à ce titre, sont indispensables pour le fonctionnement d'un réseau IMS. Elles sont des entités logiques, ce qui signifie que, malgré leur distinction fonctionnelle, rien n'empêche de les implémenter (toutes ou certaines) au sein d'un même équipement.

Ces entités sont les suivantes :

- HSS et SLF
- CSCF (décliné en P-CSCF, I-CSCF et S-CSCF)
- MRF (réparti en MRFC et MRFP)
- BGCF
- MGW, MGCF et SGW
- TrGW et IMS-ALG
- Serveur d'applications (SIP, IM-SSF et OSA-SCS)

Les sections suivantes détaillent chacune d'entre elles.

## ***HSS (Home Subscriber Server) : la base de données des utilisateurs***

Le serveur HSS (Home Subscriber Server) est une entité centralisée qui stocke une base de données contenant l'ensemble des profils des utilisateurs. Il comporte, entre autres et pour chaque utilisateur, les informations suivantes :

- Localisation des terminaux des utilisateurs abonnés.
- Identité complète des utilisateurs susceptibles d'accéder au réseau IMS.
- Paramètres permettant leur authentification et les informations d'autorisation d'accès.
- Ensemble des services auxquels ils ont souscrits.
- Préférences de services (on peut imaginer toutes sortes de préférences, par exemple la redirection vers la messagerie d'un utilisateur en fonction de l'heure).
- Serveur S-CSCF, qui traite les demandes de l'utilisateur (attribué dynamiquement lors de la connexion de l'utilisateur).

Toutes les données relatives à un même compte utilisateur doivent être stockées sur un même HSS. Néanmoins, lorsque le nombre d'utilisateurs est important, il est possible de les répartir au sein de plusieurs HSS. Dans ce cas, il est nécessaire de mettre en place une entité complémentaire, appelée SLF (Subscription Locator Function), qui a pour rôle de déterminer le HSS contenant les données relatives à un utilisateur.

Dans la suite, nous nous contenterons de parler d'un serveur HSS, même s'il peut y en avoir plusieurs, auquel cas, il s'agirait d'un serveur SLF qui relaie les requêtes vers le HSS qui convient.

Les serveurs HSS et SLF communiquent avec les autres entités du réseau au moyen du protocole Diameter.

## ***CSCF (Call State Control Function) : les trois gestionnaires d'appels***

Les CSCF sont des serveurs de contrôle de sessions, qui utilisent le protocole de signalisation SIP pour communiquer. Ils sont les organes de traitement des références dans l'architecture IMS et réalisent les fonctions permettant d'orienter et de contrôler une session.

Concrètement, lorsqu'un utilisateur se connecte, il peut accéder à un réseau qui n'est pas celui de l'opérateur chez qui il a souscrit un service. Il faut donc que l'utilisateur emprunte le réseau pour assurer sa connectivité, tout en cherchant à se relier à des entités appartenant à son opérateur, pour avoir accès aux services de son contrat.

Il existe trois types de CSCF, représentés à la figure 16.2. Chacun de ces serveurs peut se trouver en nombre dans un réseau IMS, notamment pour répondre à la charge des demandes.

Nous détaillons dans les sections qui suivent le rôle de chacun de ces serveurs.

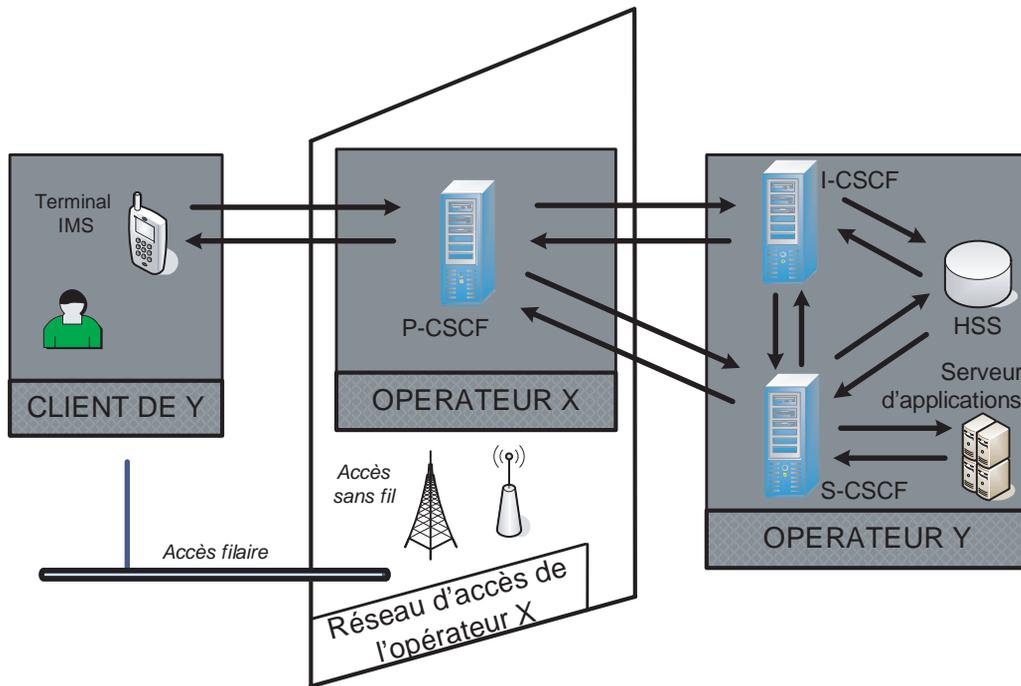


Figure 16.2

*Les serveurs CSCF*

### P-CSCF (Proxy-CSCF)

Le serveur P-CSCF constitue le point d'entrée dans le réseau IMS pour tout ce qui concerne la signalisation d'une session d'un utilisateur. Il est situé dans le réseau auquel le terminal se connecte, même si ce réseau n'est pas celui de l'opérateur auquel l'utilisateur est abonné.

Dès qu'une requête de signalisation est envoyée depuis un terminal, elle est réceptionnée par un P-CSCF. Ce dernier est chargé de prendre en charge les requêtes d'un utilisateur et de les rediriger vers les serveurs concernés (c'est-à-dire les serveurs appartenant à l'opérateur auquel l'utilisateur a souscrit un service). De même, après avoir envoyé la requête de l'utilisateur au serveur concerné, c'est le P-CSCF qui relaie la réponse obtenue vers le terminal. Le P-CSCF est donc un intermédiaire imposé entre le terminal et les autres serveurs.

Dans la pratique, lorsqu'un terminal est connecté au réseau IMS, il initie une étape d'activation de contexte PDP lui permettant de négocier les paramètres avec lesquels ses services vont être lancés. Cette étape est indispensable pour pouvoir ensuite émettre et recevoir des requêtes SIP. Lors de cette étape, le terminal découvre l'adresse du serveur

P-CSCF. Celui-ci sert à aiguiller chacune des requêtes émises par le terminal. Autrement dit, le terminal s'adresse au P-CSCF pour lui formuler ses requêtes, et c'est le P-CSCF qui est chargé de les mener à bien.

Le P-CSCF est déterminé et attribué pour toute une session, d'une connexion à la déconnexion d'un utilisateur. Par contre, il peut changer lors de la session suivante de l'utilisateur.

Le serveur P-CSCF n'est pas nécessairement la propriété de l'opérateur auquel le client a souscrit un service. Au contraire, et c'est tout son intérêt, il peut être un intermédiaire entre le terminal client IMS et les serveurs de l'opérateur de l'utilisateur.

Considérons un utilisateur qui se trouve physiquement dans une région où son opérateur ne dispose d'aucune infrastructure pour permettre sa connexion, par exemple à l'étranger. Dans ce cas, et pour peu que l'opérateur de l'utilisateur ait un contrat avec l'opérateur qui gère cette région, l'utilisateur peut quand même se connecter en utilisant l'infrastructure du réseau visité, matérialisé par le P-CSCF de l'opérateur de la région. Le terminal est dans ce cas « visiteur » dans le P-CSCF qu'il sollicite.

Outre sa nécessité physique, le P-CSCF facilite les facturations entre opérateurs. Bien sûr, le P-CSCF peut aussi être celui de l'opérateur auquel l'utilisateur a souscrit un abonnement.

Ainsi, dans tous les cas, le terminal possède un interlocuteur virtuel privilégié, auquel il délègue le soin d'aiguiller ses demandes. De cette manière, le terminal n'a qu'une vision minimale des serveurs du réseau IMS, et c'est au serveur P-CSCF que revient la charge de localiser et contacter les autres entités.

Le réseau de signalisation est de la sorte protégé par le P-CSCF, qui agit comme un garde-barrière. Il peut même avoir les fonctionnalités suivantes :

- Contrôle de syntaxe des requêtes SIP.
- Contrôle d'intégrité des données.
- Contrôle d'admission en refusant des appels lorsque le réseau est trop chargé, afin de ne pas dégrader la qualité des communications existantes.

À ce titre, le rôle du P-CSCF est comparable à celui que joue le proxy SIP dans l'architecture SIP.

Le P-CSCF doit également permettre l'aboutissement des appels d'urgence. De plus, il est aussi responsable de la gestion de la qualité de service pour l'utilisateur. En ce sens, il agit comme le PEP d'une architecture de contrôle COPS.

Outre sa fonctionnalité d'intermédiaire dans l'acheminement des méthodes SIP entre le terminal de l'utilisateur et le serveur approprié, le P-CSCF offre des fonctionnalités de compression/décompression des messages SIP, afin de réduire le temps d'établissement d'une connexion.

Le terminal IMS a la possibilité de compresser ses données, ce qui permet de les transmettre plus rapidement au P-CSCF qui les décompresse avant de les transmettre aux

entités sollicitées. C'est d'autant plus utile que les réseaux radio (utilisés le plus souvent dans la couche d'accès) ont généralement une bande passante réduite. L'objectif de la compression n'est pas vraiment de ne pas gaspiller la bande passante, mais plutôt sa conséquence immédiate, qui est d'accélérer la transmission du message.

### **I-CSCF (Interrogating-CSCF)**

Le serveur I-CSCF constitue, pour une session d'un utilisateur, le point d'entrée dans le réseau de l'opérateur auquel l'utilisateur a souscrit un contrat de service. Par exemple, si un utilisateur est dans un réseau qui n'appartient pas à son opérateur, il peut emprunter ce réseau en utilisant un intermédiaire (le serveur P-CSCF), lequel redirige ses requêtes vers une entité du réseau de son opérateur (le serveur I-CSCF). Cette redirection est réalisée, même si l'utilisateur est dans le réseau de son opérateur, mais, dans ce cas, son intérêt est moindre.

Le serveur I-CSCF n'est que le point d'entrée du réseau d'un opérateur. Il ne réalise pas de traitements pour répondre aux requêtes des utilisateurs, mais se contente de désigner des entités qui le font (les serveurs S-CSCF, décrits dans la section qui suit). Le rôle du I-CSCF est donc essentiellement de masquer le réseau de l'opérateur en fournissant un simple point d'entrée et d'aiguiller les requêtes de ses abonnés vers les serveurs adéquats.

Concrètement, c'est grâce au P-CSCF que le serveur I-CSCF peut être sollicité : l'utilisateur, qui connaît uniquement le domaine de son opérateur, envoie cette information vers le P-CSCF auquel il est connecté. Avec le domaine mentionné par l'utilisateur, le P-CSCF peut effectuer une requête DNS pour déterminer l'adresse IP du serveur I-CSCF et peut alors contacter le serveur I-CSCF.

Le I-CSCF assure également des fonctionnalités de facturation et de pare-feu pour protéger le réseau de l'opérateur puisqu'il se place à l'entrée de celui-ci.

Le I-CSCF peut être relié à d'autres I-CSCF appartenant à d'autres opérateurs, afin de permettre les communications entre les utilisateurs d'opérateurs distincts.

### **S-CSCF (Serving-CSCF)**

Le serveur S-CSCF doit assurer la gestion de la session d'un utilisateur. Plus précisément, tous les messages de signalisation SIP émis ou reçus par le terminal IMS de l'utilisateur traverseront le S-CSCF attribué à la session du terminal et seront traités par ce dernier.

Le serveur S-CSCF est déterminé grâce au serveur I-CSCF (et toujours par l'intermédiaire du P-CSCF) : après que le P-CSCF a contacté le I-CSCF, ce dernier assigne un S-CSCF pour servir l'utilisateur, en fonction des S-CSCF disponibles, afin de répartir au mieux la charge supportée par chacun d'entre eux. Il en informe alors le P-CSCF, de manière que ce dernier puisse ultérieurement s'adresser directement au S-CSCF. Parallèlement, le S-CSCF enregistre dans le HSS la position de l'abonné dans le réseau et

indique au HSS son adresse, afin qu'une entité cherchant à joindre l'abonné sache à quel S-CSCF s'adresser.

Une fois désigné pour servir la session d'un utilisateur, le premier rôle du S-CSCF est de récupérer auprès de la base HSS, et avec le protocole Diameter, l'ensemble des paramètres de profil de l'utilisateur pour l'enregistrer, l'authentifier et vérifier ses droits d'accès. De plus, il met à jour, dans cette même base HSS, l'état courant des sessions dont il a la charge, afin de pouvoir localiser l'utilisateur dans ses déplacements (en sauvegardant leur adresse IP), modifier les préférences de ce dernier, mais aussi avoir l'historique de ses communications (utile à la facturation, par exemple).

Remarquons que le terminal de l'utilisateur lui-même ne connaît jamais l'adresse du I-CSCF, ni celle du S-CSCF. Ces informations sont masquées et ne sont connues que du P-CSCF qui sollicite les serveurs concernés pour le terminal.

Le rôle du serveur P-CSCF doit être précisé puisque lui aussi fait transiter tous les messages SIP émis ou reçus par le terminal : en fait, le serveur P-CSCF ne fait essentiellement que relayer tous les messages qui lui parviennent vers le I-CSCF ou le S-CSCF, il n'est donc qu'un intermédiaire. Au contraire, le S-CSCF est responsable des traitements à réaliser pour l'utilisateur.

Le S-CSCF est de plus l'entité chargée d'effectuer le routage des appels. En particulier, si l'adresse de destination n'est pas une adresse SIP, mais, par exemple, un numéro de téléphone standard, le serveur S-CSCF fournit les fonctionnalités de conversion pour joindre les passerelles téléphoniques.

Enfin, le S-CSCF est chargé d'effectuer l'interaction avec les serveurs d'applications en commutant les requêtes de l'utilisateur vers les serveurs adéquats.

Tout comme le P-CSCF, le S-CSCF n'est pas déterminé statiquement pour servir un utilisateur, mais lorsqu'une session démarre, le S-CSCF choisi ne change pas tout au long de la session de l'utilisateur. En particulier, lors de la phase d'enregistrement, le S-CSCF stocke temporairement les informations relatives au profil de l'utilisateur (qu'il aura récupéré auprès du HSS) de manière à traiter plus rapidement et directement les droits et préférences de ce dernier.

### ***MRF (Multimedia Resource Function) : pour la gestion des conférences***

L'entité MRF (Multimedia Resource Function) permet d'établir un pont de conférence entre les utilisateurs d'un réseau IMS. Son rôle est de gérer la signalisation de et vers tous les utilisateurs d'une conférence, en offrant des facilités d'exploitation, comme la sélection des types de flux — par exemple, si sa bande passante est faible, il est possible de demander uniquement le flux audio sur son poste, tandis que d'autres utilisateurs disposeront de la vidéo — et la conversion des formats de codecs — il est possible d'utiliser des codecs différents pour chaque utilisateur, le MRF adaptant chaque flux aux exigences de chacun.

Le MRF a un fonctionnement semblable à celui de l'entité MCU (Multipoint Control Unit), que nous avons introduite au chapitre 3, consacré à H.323, si ce n'est qu'il manipule une signalisation SIP.

Comme ce dernier, le MRF se décompose en deux entités logiques :

- MRFC (Multimedia Resource Function Controller) : pour la partie signalisation, et plus précisément la négociation des paramètres sollicités par chaque utilisateur pour la mise en œuvre de la conférence.
- MRFP (Multimedia Resource Function Processor) : pour la partie traitement des flux de données, c'est-à-dire l'application des demandes formulées par l'utilisateur dans les flux.

### ***BGCF (Breakout Gateway Control Function) : pour la liaison avec le réseau RTC***

BGCF (Breakout Gateway Control Function) est un serveur qui communique en utilisant le protocole SIP. Il sert à préciser le routage des appels initiés par des terminaux IMS vers des terminaux fonctionnant en mode commutation de circuits (réseaux filaires traditionnels ou réseau GSM, par exemple).

En considérant qu'un terminal IMS cherche à joindre un correspondant dans le réseau RTC, deux cas peuvent se produire, selon la compatibilité de l'appareil :

- Si l'appareil qui se connecte est compatible avec le réseau du correspondant qu'il cherche à joindre, le BGCF se charge de mettre en relation les deux entités.
- Sinon, comme l'appareil n'est pas compatible avec le réseau du correspondant qu'il cherche à joindre, le BGCF indique des passerelles spécifiques (de type MGCF, présentées plus loin) responsables d'effectuer la liaison et auxquelles le BGCF relaie toute la signalisation qu'il reçoit.

### ***IMS-MGW, MGCF et SGW : pour l'interfonctionnement avec le réseau RTC***

Avec l'IMS, il est indispensable d'être ouvert aux réseaux les plus classiques, ce qui inclut bien évidemment le réseau commuté RTC.

Pour ce faire, les trois entités illustrées à la figure 16.3 sont introduites. Il s'agit de l'IMS-MGW, du MGCF et du SGW.

Ces trois entités se répartissent sur deux plans : un plan de données (couche transport) et un plan de signalisation (couche contrôle). Nous allons décrire chacune d'elles.

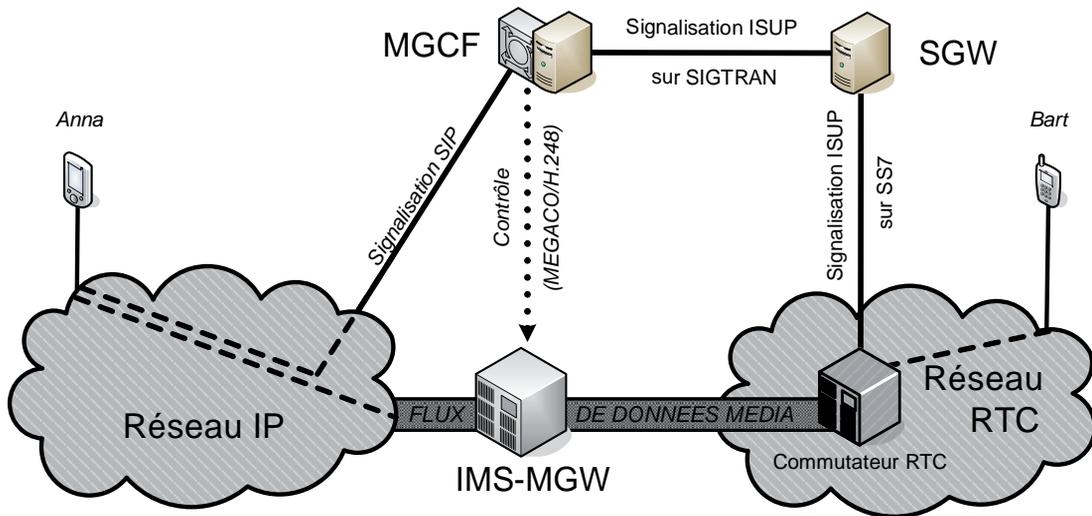


Figure 16.3

*Interfonctionnement avec le réseau RTC*

### La passerelle de flux multimédia IMS-MGW (IMS-Media GateWay)

Cette entité assure le transport des flux de données d'un réseau IP vers un réseau RTC, en effectuant la liaison entre un terminal du côté IP et son correspondant du côté RTC, avec le transcodage correspondant (d'un flux RTP vers un flux TDM). C'est donc une passerelle de flux de données multimédias qui convertit les flux de données multimédias codés en RTP (dans le réseau IP) en flux de données multimédias codés en TDM (dans le réseau RTC). Elle se situe sur la couche transport.

Généralement, cette passerelle possède des fonctionnalités complémentaires de traitement du média, comme la suppression d'écho.

### Le contrôleur de passerelle MGCF (Media Gateway Control Function)

Cette entité, qui agit au niveau de la signalisation, assure le passage du mode paquet (IP) au mode circuit (RTC) en ouvrant, maintenant et fermant des connexions entre les réseaux IP et RTC et en assurant la conversion des protocoles de signalisation propres à ses réseaux.

Le serveur doit donc convertir un message SIP qui provient du réseau IP en un message ISUP dans le réseau RTC. Le message ISUP correspondant sera remis à l'entité SGW, présentée plus loin.

Le MGCF détermine le transcodage des flux de données et contrôle la passerelle de flux multimédias IMS-MGW. La communication entre le serveur MGCF et le serveur IMS-MGW se fait au moyen du protocole de signalisation MEGACO/H.248.

Le serveur MGCF doit également informer le serveur S-CSCF des messages de signalisation qu'il génère pour que ce dernier connaisse les communications et opérations en cours.

### **La passerelle de signalisation SGW (Signaling GateWay)**

Comme la précédente, cette entité concerne la partie signalisation. Sa fonction est de transporter, un message de signalisation ISUP d'un transport SS7 vers SIGTRAN.

Le serveur SGW joue le rôle de passerelle pour la signalisation ISUP. Il est en contact avec l'entité MGCF qui se charge de remplacer la signalisation ISUP en signalisation SIP.

## ***TrGW et IMS-ALG : pour le support transparent des adresses IPv4***

### **D'IPv4 à IPv6**

Le protocole IPv6 est recommandé pour l'IMS. En effet, si l'on considère que chaque terminal téléphonique est, potentiellement, un client IMS, il faut fournir une adresse IP à chacun d'entre eux ; or le nombre d'adresses IPv4 est insuffisant pour couvrir le besoin d'adressage de tous les terminaux téléphoniques.

IPv4 dispose d'un adressage sur 32 bits, ce qui laisse  $2^{32}$  adresses disponibles au maximum (sur les 32 bits, certains sont réservés à des usages ou distinctions spécifiques, tels que l'adressage privé ou le broadcast), soit environ 4 milliards d'adresses IP. De plus, de nombreux utilisateurs disposent de plusieurs terminaux IP (un ordinateur personnel, un autre professionnel, un téléphone portable personnel, un autre fixe, etc.), soit autant d'adresses IP potentiellement nécessaires.

Face à cette pénurie d'adresses IPv4, il sera donc nécessaire de s'orienter vers IPv6, qui offrira 128 bits pour l'adressage, soit environ  $2^{128}$  adresses disponibles et un nombre relativement large de terminaux pouvant être simultanément connectés et compatibles avec le protocole IP.

Néanmoins, si la version 6 du protocole IP existe depuis 1995 et que sa maturité est rarement contestée, son exploitation dans les réseaux IP demeure, encore aujourd'hui, particulièrement lente, voire incertaine.

Une des raisons à cela est l'investissement à réaliser pour les opérateurs afin de rendre leur infrastructure compatible avec le nouveau protocole. En outre, l'absence de réel besoin immédiat de cette technologie freine son évolution. Seuls quelques efforts ponctuels ont été initiés pour œuvrer à l'essor d'IPv6. Par exemple, l'autorité mondiale de régulation des adresses dans l'Internet, l'ICANN (Internet Corporation for Assigned Names and Numbers) a attendu l'année 2008 pour rendre compatibles avec IPv6 six de ses treize serveurs DNS (Domain Name Server). En revanche, l'opérateur français Free a déployé la technologie dans son réseau pour tous ces abonnés.

IMS pousse en faveur du protocole IPv6, car si la pénurie d'adresses IP pour les ordinateurs n'est pas encore d'actualité, il est certain que le déploiement d'un réseau IMS va rapidement engendrer un grand besoin d'adresses IP afin d'y connecter tous les terminaux téléphoniques.

### Compatibilité assurée avec des passerelles IPv4-IPv6

Le système IMS doit considérer à la fois un réseau IPv6, inévitablement attendu, et un réseau IPv4 existant. Pour permettre une évolution en douceur vers la version 6, IMS assure la compatibilité avec IPv4 et met en œuvre des entités spécialement dédiées aux communications entre des stations qui communiquent avec IPv4 et des stations qui communiquent avec IPv6. De cette manière, le réseau IMS se charge d'effectuer la transition entre les deux protocoles de manière transparente.

Pour cela, les deux entités illustrées à la figure 16.4 sont introduites : TrGW (TRansition GateWay) et IMS-ALG (IMS-Application Layer Gateway).

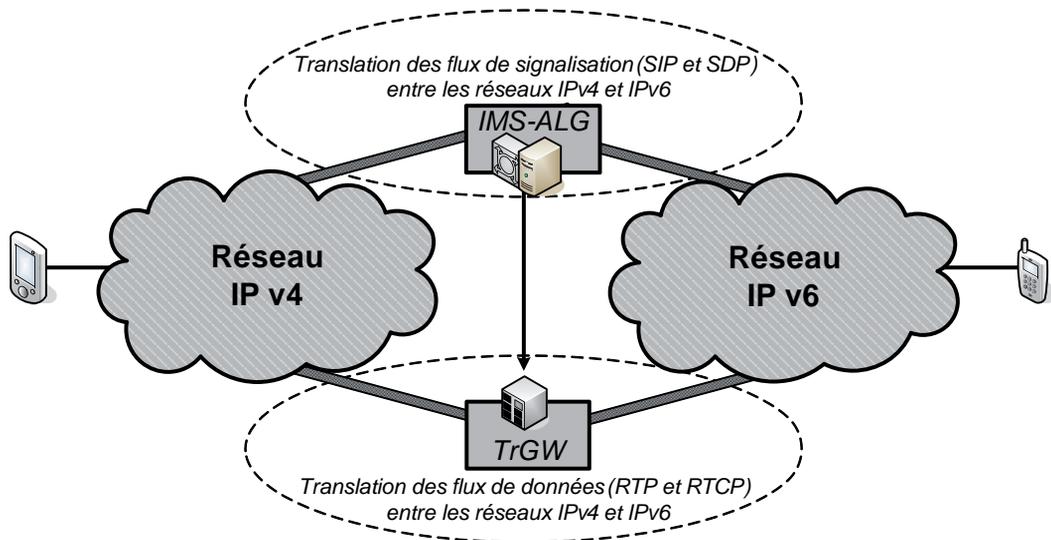


Figure 16.4

Les serveurs IMS-ALG et TrGW

Lorsqu'une station IPv6 communique avec une station IPv4, cette dernière n'est, *a priori*, pas capable de comprendre le message qui lui parvient. L'idée consiste à mettre en place une entité intermédiaire entre ces deux stations afin d'effectuer la conversion du protocole IP entre les versions 4 et 6.

Comme les messages qu'une station peut envoyer à une autre sont de deux types (flux de données et signalisation), deux entités fonctionnelles sont dévolues à ces tâches : la

passerelle TrGW, qui effectue la translation d'adresses sur les flux de données, et la passerelle IMS-ALG, qui effectue la translation d'adresses au niveau de la signalisation.

Autrement dit, la passerelle TrGW agit sur les messages de données encapsulés par RTP (et les retours RTCP), tandis que la passerelle IMS-ALG agit sur les messages de signalisation SDP et SIP.

## Les serveurs d'applications

Les serveurs d'applications — souvent abrégés en AS (Application Server) — sont des entités SIP fournissant différents types de services aux utilisateurs. Ils sont connectés au serveur S-CSCF, qui joue l'intermédiaire entre l'utilisateur et les services.

On distingue trois grandes familles de serveurs d'applications : SIP AS (SIP Application Server), IM-SSF (IP Multimedia-Service Switching Function) et OSA-SCS (Open Service Access-Service Capability Server).

### *SIP AS (SIP Application Server)*

Ces serveurs permettent l'exécution des services nativement implémentés pour fonctionner avec SIP.

Les services les plus classiques (service de présence, push-to-talk, messagerie instantanée, etc.) sont généralement implémentés au sein de ces serveurs.

### *IM-SSF (IP Multimedia-Service Switching Function)*

Pour permettre la mobilité de l'abonné tout en lui garantissant la fourniture de ses services — même s'il se trouve dans une infrastructure qui n'appartient pas à son opérateur de services (on parle de roaming pour désigner la connexion d'un utilisateur à un réseau qui n'est pas celui de son opérateur) —, il est nécessaire d'avoir une passerelle, appelée IM-SSF, afin de connecter l'abonné au serveur d'applications de son opérateur.

La communication entre le serveur IM-SSF et le serveur d'applications de l'opérateur distant s'effectue au moyen d'une technologie de signalisation décrivant l'interfaçage avec les applications. Cette signalisation s'appelle INAP (Intelligent Network Application Part) dans le cadre des réseaux fixes et CAMEL (Customized Applications for Mobile network Enhanced Logic) dans celui des réseaux sans fil GSM.

Les communications entre le serveur IM-SSF et les serveurs distants utilisent le protocole CAP (Camel Application Part), spécifié par le 3GPP sous la référence TS 29.278, en janvier 2005.

IM-SSF permet ainsi d'accéder à des services distants en réalisant l'interface entre, d'un côté, le serveur S-CSCF (qui communique avec le protocole SIP) et, de l'autre côté, des serveurs distants (en communiquant avec le protocole CAP).

### **OSA-SCS (Open Service Access-Service Capability Server)**

Ces serveurs fournissent le moyen d'interagir avec les serveurs d'applications OSA. OSA a été défini par le 3GPP et l'ETSI comme une architecture de gestion des services dans un réseau téléphonique de troisième génération. Son objectif est de proposer une vision très abstraite du réseau, n'imposant aucune architecture en particulier et s'adaptant facilement, quelle que soit l'architecture considérée.

Grâce à OSA, les développeurs peuvent concevoir des applications sans prendre en compte les spécificités architecturales du réseau considéré, ce qui facilite le processus de déploiement et évite d'avoir à repenser plusieurs fois les mêmes applications selon les réseaux cibles.

Pour cela, OSA fournit une API qui facilite le développement des services. Cette API, ouverte, librement téléchargeable et indépendante des technologies utilisées, a été nommée OSA Parlay. Elle tire son nom du consortium Parlay (<http://www.parlay.org>), qui regroupe plusieurs industriels, tels que France Telecom, IBM, Telcordia, Sprint ou Ericsson, et qui a développé les API, avant de les soumettre au 3GPP pour normalisation. Le 3GPP y fait référence, de manière plus neutre, sous le nom d'API OSA (document de janvier 2005, référencé TS 29.278). Cette souplesse dans le mode de développement des applications, ajoutée à la possibilité d'interfaçage avec ces applications, sont accessibles avec l'IMS grâce au serveur OSA-SCS.

Ainsi, OSA-SCS permet d'accéder à des services génériques en réalisant l'interface entre, d'un côté, le serveur S-CSCF (qui communique avec le protocole SIP) et, de l'autre côté, des serveurs implémentant des applications pensées sur le modèle générique de l'architecture OSA (en communiquant avec les API OSA). Cela permet l'émergence de services tiers et favorise le développement de nouveaux services.

Notons pour finir la notion d'applications dites *enabler*, c'est-à-dire définissant un composant de base standardisé réalisant une fonction précise et qui est facilement réutilisable et exploitable pour construire des services plus sophistiqués. Les services de présence, de localisation ou de messagerie instantanée (SMS ou MMS notamment) sont des exemples d'applications *enabler*.

## **Les identités IMS**

Le réseau IMS reprend la notion d'identités publique et privée, qui est propre aux réseaux à commutation de circuits des télécoms.

L'identité publique d'un utilisateur désigne un URI (SIP ou de type téléphonique) affecté publiquement à l'utilisateur et utilisé par les correspondants de ce dernier pour le

désigner, et donc le joindre. Dans la terminologie télécoms, ce concept est l'équivalent du MSISDN (Mobile Subscriber ISDN Number).

Par exemple, si un utilisateur dispose d'une adresse professionnelle et d'une adresse privée, il dispose de deux identités publiques correspondant à chacune de ses adresses. De plus, même si l'utilisateur ne dispose que d'une adresse privée, celle-ci peut se présenter à la fois sous la forme d'un URI SIP et d'un URI téléphonique. La première forme (au format *sip:identifiant@opérateur*) est adaptée au protocole SIP qu'exploite l'IMS et est donc préférable de manière générale. La seconde (au format *tel:+952955295*) est indispensable pour être joignable à partir de terminaux non-IMS, qui ne disposent que de chiffres pour joindre leurs correspondants et ne sont joignables que par un numéro d'appel.

L'identité privée d'un utilisateur désigne un identifiant au format NAI (Network Access Identifier) qui est de type *identifiant@opérateur*. Elle est affectée secrètement à l'utilisateur et est utilisée par l'opérateur pour désigner ce dernier. Dans la terminologie télécoms, ce concept est l'équivalent de l'IMSI (International Mobile Subscriber Identifier).

Par exemple, si un utilisateur dispose de plusieurs identités publiques, il peut toutes les gérer à partir d'un compte unique (d'une identité privée) et donc d'un terminal unique, lequel recevra les appels destinés à l'utilisateur, et ce quelle que soit l'identité publique avec laquelle les appels sont adressés. L'identité privée est avant tout utile pour permettre l'identification et l'authentification de l'abonné par l'opérateur.

De la même manière qu'une identité privée peut être liée à plusieurs identités publiques, une identité publique peut être liée à plusieurs identités privées. Par exemple, considérons un utilisateur qui dispose de deux terminaux IMS, le premier étant professionnel, le second privé, et de deux identités publiques, là aussi, une professionnelle et l'autre privée. Chacun des terminaux de cet utilisateur sera configuré avec une identité privée. Il peut décider de ne recevoir, sur le premier terminal, que les appels professionnels (c'est-à-dire liés à son profil public professionnel), tandis que, sur le second terminal, il décidera de recevoir tous les appels (c'est-à-dire à la fois ceux liés à son compte professionnel et ceux liés à son compte personnel).

Ainsi certaines identités privées sont-elles liées à plusieurs identités publiques, et certaines identités publiques à plusieurs identités privées.

### **Gestionnaire d'identités**

Les terminaux IMS sont identifiés par la présence d'une carte à puce appelée UICC (Universal Integrated Circuit Card), qui comporte un ensemble d'informations d'identification et de personnalisation, et sans laquelle seuls les appels d'urgences sont possibles.

L'interface de communication entre la carte UICC et le terminal est standardisée, mais la carte UICC est générique et peut comporter toutes sortes d'applications, notamment les applications classiques suivantes :

- **SIM (Subscriber Identity Module).** Cette application spécifiée dans les documents 3GPP TS 11.11 et 3GPP TS 51.011 est bien connue pour son usage massif et généralisé dans les réseaux GSM. En fait, le 3GPP ne fait ici que reprendre une application existante et formaliser son usage. Elle fournit, en particulier, des informations relatives au réseau de l'utilisateur, son identité, ses préférences, ses paramètres d'authentification et de cryptage, ainsi que des informations de stockage (typiquement la sauvegarde de SMS ou de son carnet d'adresses).
- **USIM (Universal Subscriber Identity Module).** Cette application spécifiée dans le document 3GPP TS 31.102 est indispensable pour opérer la connexion d'un terminal dans un réseau de troisième génération (UMTS, par exemple). La nature des informations stockées est semblable à celle de la SIM, mais la manière dont l'information est stockée diffère.
- **ISIM (IP multimedia Services Identity Module).** Cette application spécifiée dans le document 3GPP TS 31.103 stocke notamment les identités publiques et privées d'un utilisateur, le domaine de l'utilisateur (c'est-à-dire le réseau appartenant à son opérateur) et une clé secrète utilisée pour des besoins de sécurité et d'authentification. Ces informations ne sont accessibles qu'en lecture seule, ce qui prévient toute modification par l'utilisateur. C'est la forme la plus adaptée pour gérer l'ensemble des paramètres de l'utilisateur dans un réseau IMS.

Ces trois applications ne sont évidemment pas les seules. Même si elles peuvent contenir des informations redondantes, elles peuvent être utilisées en même temps par le terminal pour fournir plusieurs de ces données enregistrées, selon les connexions (SIM pour les réseaux GSM, USIM pour les réseaux UMTS, etc.).

Par contre, il est indispensable de disposer au moins de l'application ISIM ou USIM pour se connecter dans un réseau IMS (le niveau de sécurité de SIM est insuffisant pour être possible avec IMS) du 3GPP. En ce qui concerne la version IMS du 3GPP2, la notion d'UICC est absente, mais ces mêmes applications USIM et ISIM peuvent être fournies comme une configuration du terminal de l'utilisateur.

## Les extensions du protocole SIP pour IMS

IMS reprends le protocole SIP dans sa syntaxe et dans son fonctionnement. Néanmoins, il définit un ensemble d'extensions supplémentaire dans les en-têtes du protocole SIP pour répondre à quelques besoins spécifiques. Ces extensions sont optionnelles pour le protocole SIP ; elles sont préfixées par la lettre P pour indiquer qu'elles sont privées (en anglais *private*), suivie d'un tiret.

Elles adoptent la même syntaxe que les en-têtes conventionnels, à savoir le format suivant :

*nom\_en\_tête:valeur\_en\_tête*

Le tableau 16.1 récapitule les en-têtes les plus utilisés.

**Tableau 16.1 – En-têtes SIP supplémentaires pour l'IMS**

Nom de l'en-tête	Description
P-Asserted-Identity	<p>Cet en-tête permet aux entités réseau (typiquement le P-CSCF) de préciser l'identité publique d'un utilisateur appelant. Il est utilisé par une entité du réseau qui aura préalablement identifié l'utilisateur (ou bien qui est une entité de confiance), ce qui rend l'identité certaine.</p> <p>L'en-tête spécifie à sa suite un ou deux URI. S'il y a un seul URI, il est au format SIP ou SIPS (éventuellement précédé du nom de l'utilisateur à qui l'URI appartient) ou bien détermine un numéro de téléphone. S'il y a deux URI, le premier est nécessairement au format SIP ou SIPS, et le second nécessairement au format téléphonique. On aura par exemple :</p> <p><i>P-Asserted-Identity : "Hanna"&lt;sip : HannaM@nina.com&gt; tel : +908070605</i></p>
P-Preferred-Identity	<p>Cet en-tête est spécifié par le terminal appelant (User Agent Client) et permet aux entités réseau (typiquement le P-CSCF) de connaître l'identité publique que préfère utiliser un utilisateur appelant. Il peut y avoir une ou deux valeurs de ce type, de la même manière que pour l'en-tête précédent <i>P-Asserted-Identity</i>.</p>
P-Called-Party-ID	<p>Cet en-tête permet à l'utilisateur appelé de savoir avec quelle identité publique l'appelant l'a contacté. Par exemple, un utilisateur peut être joignable par son adresse professionnelle et par son adresse privée. Ce champ d'en-tête lui permet de savoir laquelle de ses adresses a été utilisée, de manière à pouvoir personnaliser sa gestion (sonnerie particulière, envoi en messagerie, etc.). L'en-tête est généralement ajouté lors de l'envoi d'une requête d'invitation (INVITE) par un serveur SIP.</p>
P-Access-Network-Info	<p>Cet en-tête permet au terminal IMS de donner des informations sur le réseau qu'il utilise, par exemple « IEEE-802.11b », si l'utilisateur se connecte sur un réseau Wi-Fi à la norme 802.11b.</p>
P-Visited-Network-ID	<p>Cet en-tête permet aux entités du réseau appartenant à l'opérateur de l'abonné (typiquement le S-CSCF) de connaître les réseaux que son abonné utilise. Cette information est essentielle pour que l'abonné puisse se connecter au réseau qu'il visite : lorsque le réseau de son opérateur va connaître le réseau que l'abonné est en train d'utiliser, il va permettre ou non (selon que les opérateurs ont un accord de roaming entre eux) la connexion de l'abonné.</p>
P-Associated-URI	<p>Cet en-tête permet aux entités du réseau appartenant à l'opérateur de l'abonné (typiquement le serveur S-CSCF) de retourner l'ensemble des adresses (ou URI) associées à l'identité publique de l'utilisateur. On trouve cet en-tête dans la réponse 200 OK à une requête d'enregistrement (REGISTER).</p>
P-Charging-Function-Addresses	<p>Cet en-tête permet de déterminer les adresses de l'entité chargée de la facturation des abonnés. Ajouté par un serveur SIP, il est supporté à la fois comme un champ de requête et comme un champ de réponse. Une seule instance de ce champ est tolérée. De plus, deux niveaux de facturation sont distingués : CCF (Charging Collection Function) pour les coûts hors connexion (par exemple une carte de connexion prépayée) et ECF (Event Collection Function) pour les coûts pendant la connexion (par exemple une connexion payée à la durée).</p>
P-Charging-Vector	<p>Cet en-tête permet de mettre en relation des informations de tarification concernant un utilisateur. Typiquement, en accédant à un service, un serveur va prendre en charge la tarification, mais comme la session peut solliciter d'autres services, d'autres serveurs de facturation peuvent être exploités, et il est nécessaire de les regrouper (dans ce que l'on appellera un vecteur de facturation) pour simplifier l'offre à l'utilisateur et lui fournir une facture unique.</p>

Ces en-têtes sont définis dans les RFC 3325 (pour les en-têtes *P-Asserted-Identity* et *P-Preferred-Identity*) et 3455 (pour tous les autres en-têtes) de l'IETF.

## Les commandes Diameter

Le protocole Diameter est utilisé dans l'IMS pour l'authentification, l'autorisation et la journalisation des activités de l'utilisateur (AAA, pour Authentication, Authorization, Accounting).

Spécifié dans la RFC 3588 de l'IETF, il normalise un certain nombre de commandes pour permettre son utilisation.

Le tableau 16.2 détaille la liste des commandes qui sont définies sur l'interface de commutation nommée Cx, qui est l'interface de communication la plus utilisée, entre la base de données HSS et les serveurs I-CSCF et S-CSCF.

Tableau 16.2 – Commandes sur l'interface Cx

Nom de la commande	Abrégé	Code numérique	Description
User-Authorization-Request	UAR	300	Requête d'enregistrement de l'utilisateur, émise du I-CSCF vers le HSS.
User-Authorization-Answer	UAA	300	Retourne soit le serveur S-CSCF déjà en charge de la session de l'utilisateur, soit la liste des S-CSCF disponibles, soit un message interdisant la communication de l'utilisateur (si son réseau d'accès ne bénéficie pas d'accord de roaming par exemple).
Server-Assignment-Request	SAR	301	Une fois que l'authentification s'est déroulée avec succès, le serveur S-CSCF informe le HSS que l'utilisateur est bien enregistré au réseau IMS. Cette même requête est également utilisée lorsque l'utilisateur n'est plus connecté afin de désassocier l'adresse du S-CSCF dans la base HSS.
Server-Assignment-Answer	SAA	301	En retour à la validation d'authentification, le HSS note que l'utilisateur devient joignable et fournit au S-CSCF le profil complet de l'utilisateur qui servira à identifier précisément les services et les préférences de l'utilisateur sans plus solliciter le HSS.
Location-Info-Request	LIR	302	Pour mettre en relation deux utilisateurs, il faut d'abord que l'appelant localise le serveur S-CSCF qui est en charge de la session de l'appelé. Cette demande du I-CSCF auprès du HSS permet de déterminer l'adresse SIP du serveur S-CSCF en charge de la session du terminal que l'on cherche à contacter.
Location-Info-Answer	LIA	302	Cette réponse est fournie par le HSS au I-CSCF et indique la localisation du serveur S-CSCF de l'utilisateur que l'on veut contacter (s'il est connecté au réseau).
Multimedia-Auth-Request	MAR	303	Lorsqu'un serveur S-CSCF a été désigné par le I-CSCF pour gérer la session d'un utilisateur, il demande au HSS d'associer son adresse dans le profil de l'utilisateur. Cette association permettra à tout autre entité voulant contacter l'utilisateur de s'adresser d'abord au S-CSCF désigné.

Tableau 16.2 – Commandes sur l'interface Cx (*suite*)

Nom de la commande	Abrégé	Code numérique	Description
Multimedia-Auth-Answer	MAA	303	Le HSS confirme l'association du S-CSCF à la session d'un utilisateur, et fournit également au S-CSCF les informations qui permettront d'authentifier l'utilisateur.
Registration-Terminal-Request	RTR	304	À n'importe quel moment, la base HSS peut demander au S-CSCF de déconnecter l'utilisateur avec cette requête (à des fins administratives notamment).
Registration-Terminal-Answer	RTA	304	Le serveur S-CSCF confirme au HSS qu'il a pris en compte la demande et enverra aux entités concernées (P-CSCF et terminal) une requête SIP de NOTIFY pour les en informer.
Push-Profile-Request	PPR	305	Si le profil d'un utilisateur a été modifié (avec de nouveaux services dynamiquement ajoutés, par exemple), la base HSS peut effectuer cette requête auprès du S-CSCF afin de mettre à jour les données de profil stockées sur ce dernier.
Push-Profile-Answer	PPA	305	Le serveur S-CSCF répond à la requête de mise à jour de profil d'un utilisateur.

La section suivante donne des exemples d'utilisation de ces requêtes.

## Communications avec IMS

Le protocole SIP est utilisé pour l'établissement, la gestion et la terminaison des sessions. Les messages sont donc semblables à ceux qui ont été décrits dans le chapitre sur SIP, à quelques extensions près.

Nous allons voir comment les messages SIP sont envoyés dans deux cas d'application classique :

- l'enregistrement d'un terminal dans le réseau ;
- la mise en relation de deux utilisateurs.

### *Enregistrement d'un terminal dans le réseau*

L'enregistrement d'un utilisateur dans le réseau est la première action réalisée par un terminal, dès sa mise en route. Elle est indispensable puisqu'elle permet à la fois d'appeler et d'être joignable par ses correspondants.

La méthode associée à cette fonctionnalité est REGISTRER, du protocole de signalisation SIP. Nous allons détailler le processus mis en œuvre avec SIP, qui diffère avec l'IMS du fait de son architecture particulière.

La figure 16.5 illustre les étapes temporelles liées au scénario d'enregistrement. Nous allons considérer que l'enregistrement s'effectue avec succès.

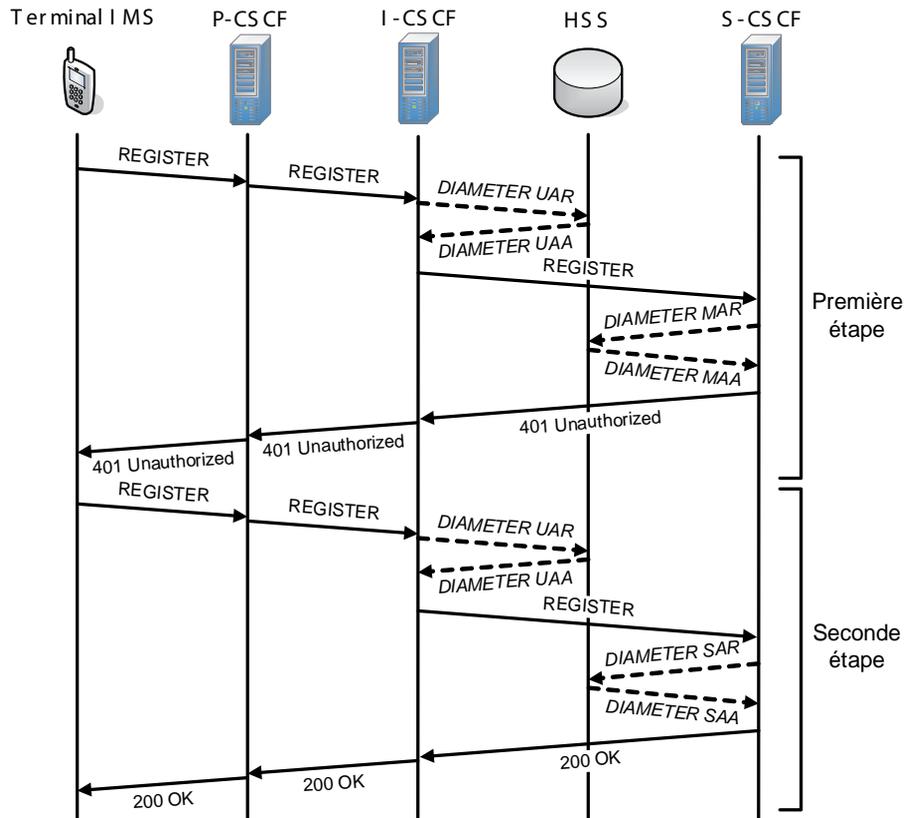


Figure 16.5

Processus d'enregistrement d'un terminal avec l'IMS

On observe globalement deux étapes SIP dans ce scénario, correspondant à la succession de deux requêtes et de leurs réponses.

### 1. Le terminal envoie une requête SIP REGISTER, et reçoit en retour une réponse 401 UNAUTHORIZED

Le terminal IMS envoie sa requête d'enregistrement au serveur P-CSCF. Celui-ci ne connaît pas nécessairement le serveur I-CSCF (dans le cas général, le P-CSCF peut appartenir à un autre opérateur que celui dont dépend l'abonné). Pour localiser le I-CSCF, le serveur P-CSCF procède à une requête DNS à partir du nom de domaine que l'utilisateur a fourni.

Une fois localisé, le P-CSCF joint le I-CSCF en lui fournissant la requête de l'utilisateur, dans laquelle il a ajouté un champ d'entête P-VISITED-NETWORK-ID. Ce champ contient un identifiant du réseau dans lequel le P-CSCF se trouve. Il permettra au S-CSCF de vérifier que le réseau visité bénéficie d'un accord de roaming avec l'opérateur de l'utilisateur. Un autre champ ajouté par le P-CSCF est le champ d'en-tête PATH qui spécifie l'adresse SIP du P-CSCF. Cette information permettra de retourner la réponse à cette requête *via* ce même serveur P-CSCF.

Lorsque le I-CSCF reçoit la requête, il ignore si elle concerne un utilisateur qui est déjà enregistré ou s'il s'agit d'un nouvel enregistrement. Le I-CSCF utilise le protocole Diameter pour contacter la base de données HSS et lui demander, à partir des identités publiques et privées contenues dans le message de requête REGISTER, d'authentifier l'utilisateur (requête UAR).

Si l'utilisateur a déjà été enregistré, un serveur S-CSCF lui a déjà été attribué, et l'adresse de ce serveur est stockée dans la base HSS. Dans ce cas, la base HSS fournit dans sa réponse (message UAA) l'adresse du S-CSCF qui est en charge de la session de l'utilisateur. Autrement, et s'il s'agit d'une première connexion pour l'utilisateur, aucun serveur S-CSCF ne lui a été attribué et la réponse UAA de la base HSS propose l'ensemble des S-CSCF disponibles avec leurs caractéristiques propres, de manière que le serveur I-CSCF puisse choisir l'un d'entre eux, selon un critère qu'il détermine librement (généralement pour répartir la charge des utilisateurs équitablement entre tous les serveurs S-CSCF).

Considérons dans notre scénario qu'il s'agit d'une première authentification : le serveur I-CSCF a donc choisi un serveur S-CSCF pour traiter la session courante de l'utilisateur, et il lui relaie la requête d'enregistrement de l'utilisateur. Le S-CSCF contacte alors la base HSS pour l'informer qu'il a été désigné pour prendre en charge la session de l'utilisateur (requête MAR). Il reçoit en retour une réponse MAA qui, d'une part, confirme l'enregistrement du S-CSCF affecté à l'utilisateur et, d'autre part, retourne des informations propres à l'utilisateur, qui vont permettre de générer un challenge pour authentifier l'utilisateur (ces informations sont appelées vecteurs d'authentification).

Lorsque le S-CSCF les reçoit, il répond au serveur I-CSCF par un message SIP 401 UNAUTHORIZED, contenant le challenge sous la forme d'un champ d'entête appelé WWW-AUTHENTICATE. Ce message de réponse est relayé conformément au modèle SIP client/serveur, c'est-à-dire de proche en proche, en passant par tous les émetteurs de requêtes : le I-CSCF d'abord, le P-CSCF ensuite et le terminal client enfin. Cette liste de serveurs est facilement déterminée car, lors de l'envoi de la requête REGISTER, les serveurs traversés ont enregistré leur adresse SIP, grâce au champ d'en-tête PATH.

## 2. Le terminal envoie une nouvelle requête SIP REGISTER et reçoit en retour une réponse 200 OK

Lorsque le client reçoit le message de réponse 401, il détecte le challenge qui s'y trouve et prépare automatiquement une réponse adaptée (sans solliciter l'utilisateur, mais en exploitant les données contenues dans la carte à puce UICC, si le terminal client en dispose, ou bien avec des données équivalentes stockées dans le terminal).

Cette réponse est générée dans une nouvelle requête d'enregistrement REGISTER. Elle est envoyée en suivant le même processus d'acheminement que le premier message de requête : le terminal client ne connaît que le serveur P-CSCF, qui, lui-même, ne peut s'adresser qu'au I-CSCF, qui, à son tour, sous-traite la demande auprès du serveur S-CSCF.

Remarquons, d'une part, que le serveur I-CSCF contacté peut être différent du premier, car son adresse est déterminée par une requête DNS effectuée par le serveur P-CSCF. Or il est classique que le DNS opère une répartition de charges entre les serveurs I-CSCF et réponde différemment à chaque requête. D'autre part, le serveur I-CSCF effectue la même requête (UAR) auprès de la base HSS. En effet, comme on vient de l'indiquer, non seulement il peut s'agir d'un autre serveur I-CSCF que celui qui a relayé le premier message, mais, de plus et surtout, même s'il s'agit du même serveur I-CSCF, ce dernier n'a gardé aucune trace de la requête précédente et ignore donc à quel serveur S-CSCF adresser cette requête : c'est la base HSS qui le renseigne dans sa réponse (message UAA) en lui fournissant l'adresse du serveur S-CSCF en charge de la session courante.

À réception de la requête, le serveur S-CSCF vérifie l'authentification de l'utilisateur, en reprenant les vecteurs d'authentification que lui a fournis le HSS à l'étape précédente et qu'il a stockés. Si les paramètres d'authentification sont valides, l'authentification est un succès, et le serveur S-CSCF en informe le HSS (par un message SAR). Dès lors, l'utilisateur peut appeler et devient joignable à l'adresse qui figure dans le champ d'en-tête CONTACT de la requête REGISTER, et ce pendant toute la durée mentionnée dans la valeur de l'en-tête EXPIRES.

Le HSS répond ensuite au S-CSCF en lui envoyant le profil complet de l'utilisateur, qui est stocké temporairement et servira à paramétrer et personnaliser les services de ce dernier.

Pour terminer, le serveur S-CSCF envoie un message de réponse 200 OK au terminal pour lui notifier le succès de l'opération d'enregistrement. Ce message comporte notamment un champ P-ASSOCIATED-URI, qui liste les adresses SIP que l'utilisateur a enregistrées auprès du HSS (et donc avec lesquelles il est joignable), et un champ SERVICE-ROUTE, qui précise explicitement l'adresse URI SIP du serveur S-CSCF que l'utilisateur pourra contacter directement dans ses requêtes suivantes (bien qu'elles transiteront obligatoirement avant par le serveur P-CSCF).

Comme précédemment, le message de réponse traverse les entités qui ont acheminé la requête, c'est-à-dire le I-CSCF, le P-CSCF puis le terminal IMS.

## Mise en relation de deux utilisateurs

Une communication implique une première étape de recherche des abonnés, de vérification des autorisations d'accès, puis de mise en relation entre les correspondants.

La méthode associée à cette dernière fonctionnalité est INVITE avec le protocole de signalisation SIP. C'est aussi la méthode que nous allons employer, mais avec une mise en œuvre particulière, dont un scénario est illustrée à la figure 16.6.

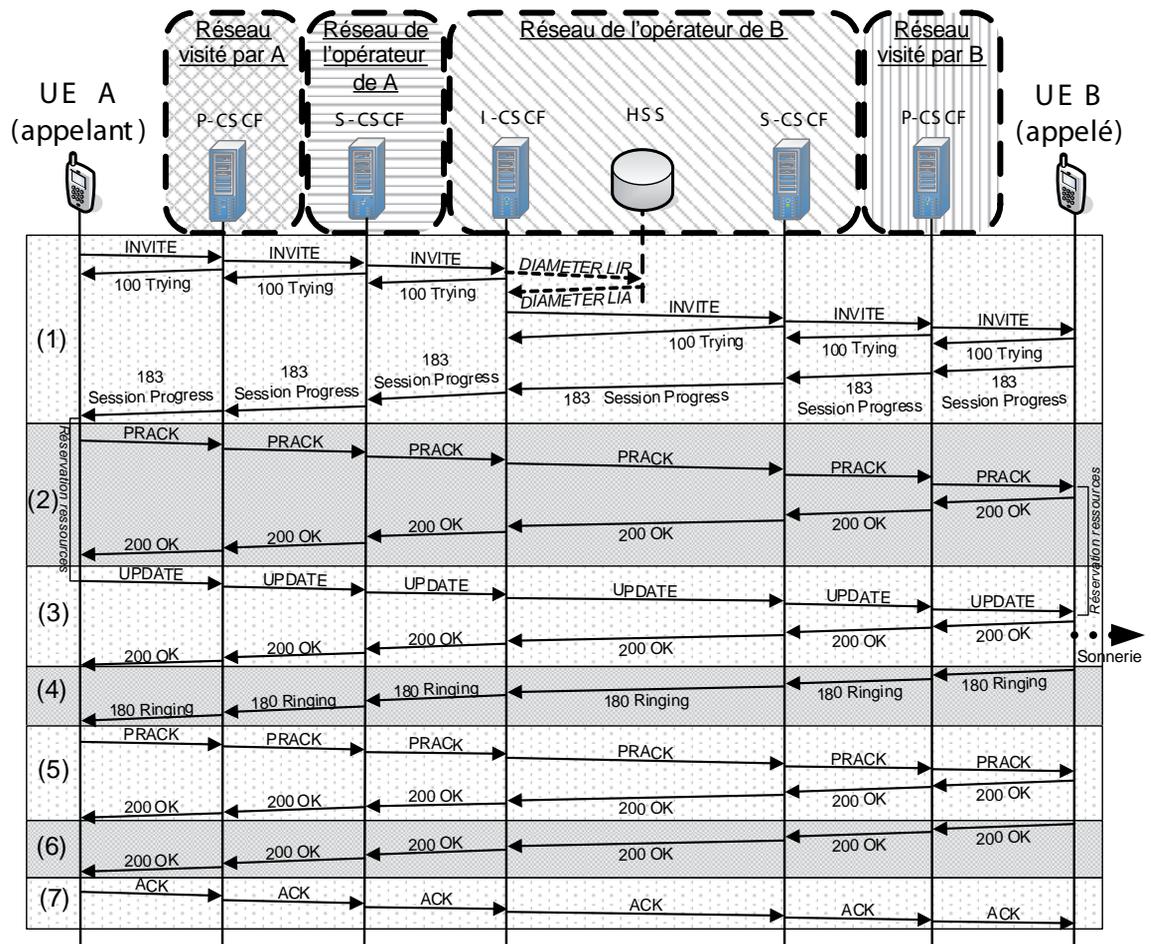


Figure 16.6

Mise en communication de deux terminaux

Nous considérons que l'appelant et l'appelé ont des opérateurs différents et sont localisés dans des réseaux visités, c'est-à-dire qui n'appartiennent pas forcément à leur opérateur respectif. C'est le cas le plus général.

Ce scénario de mise en relation de deux terminaux, ou UE (User Equipment), peut être découpé en sept grandes étapes, que nous allons brièvement présenter :

1. Message d'invitation de A vers B, avec deux réponses temporaires : une réponse 100 pour indiquer la tentative et une réponse 183 pour proposer de négocier les paramètres de la communication. L'appelant sollicite le terminal de B et, pour cela, s'adresse au serveur I-CSCF de B, qui le localise après une requête Diameter.
2. Pour s'assurer que l'émetteur A a bien reçu la réponse 183, celui-ci doit impérativement envoyer un acquittement temporaire. Comme toute requête SIP, et conformément au modèle client/serveur, une réponse est envoyée.
3. Le terminal A doit négocier les paramètres de qualité de service pour garantir sa communication dans le réseau. Cette étape permet aux utilisateurs d'établir une communication avec une bande passante garantie, et donc un service de qualité. Elle est nouvelle par rapport au processus habituel d'établissement de connexion qu'utilise SIP, car la qualité de service est un élément essentiel avec l'IMS. Le terminal B vérifie que lui aussi a réservé les ressources nécessaires à la communication dans le réseau et valide la requête par sa réponse.
4. Dès ce moment, le terminal de B commence à sonner. Cette étape complète les réponses temporaires à la requête d'invitation par une réponse 180, elle aussi temporaire.
5. Pour s'assurer que cette réponse est bien reçue du terminal A, ce dernier doit confirmer la réception par une requête d'acquiescement, qui attend elle-même une réponse.
6. Dès que l'utilisateur du terminal B a répondu (ou que sa messagerie s'est enclenchée), la réponse définitive 200 est envoyée à la requête initiale d'invitation.
7. La requête d'acquiescement finale valide l'initialisation de la communication, qui peut dès lors débiter pour permettre aux terminaux de s'échanger des flux de données multimédias.

## Conclusion

Alors que, dans le monde IP, la téléphonie n'est qu'un service banalisé parmi d'autres, elle constitue le cœur de marché des télécoms. À cet égard, le réseau télécoms paraît bien pauvre en comparaison du réseau Internet.

Les opérateurs doivent réfléchir à ce que seront les réseaux télécoms de demain, et décider si et comment la prochaine génération de réseaux télécoms proposera une gamme de services plus large, incluant la téléphonie, mais aussi la vidéo et toutes les applications réseau que l'on connaît, ou s'ils resteront confinés dans les communications téléphoniques,

réduisant leurs marges de bénéfices potentiels avec les services à valeur ajoutée dans un marché hautement concurrentiel.

Cette seconde hypothèse réduit évidemment au minimum les efforts et évolutions technologiques et impacte vers le bas les bénéfices économiques des opérateurs. C'est donc dans la première perspective que s'inscrit l'IMS.

Concurrent du GAN, l'IMS est un élément constitutif central des réseaux de nouvelle génération, dits NGN (Next Generation Network), lesquels désignent plus généralement le passage d'un réseau à transfert de circuits (RTC), vers un réseau à transfert de paquets (IP).

L'orientation prise s'inscrit dans la notion de réseaux intelligents, ou IN (Intelligent Network), qui consiste à répartir l'intelligence entre toutes les entités d'un réseau, à l'opposé — ou à la jonction — du modèle télécoms, où toute l'intelligence et le contrôle se situent dans le réseau cœur (avec des terminaux basiques), et du modèle Internet, où toute l'intelligence de contrôle se situe dans les terminaux (avec un réseau cœur très simple).

Les réseaux IN permettent de développer un plan de contrôle qui allège les commutateurs en les cantonnant à leur fonctionnalité première, qui est de connecter et transporter des flux. Cela engendre des gains dans la vitesse de déploiement des services, dans la gestion du réseau et dans les capacités de modification, de flexibilité et d'évolution.

L'IMS se pose à la frontière entre deux paradigmes : le monde des réseaux IP et celui des réseaux télécoms. En décrivant une architecture unique, l'IMS permet la cohésion de deux réseaux qui étaient, jusqu'à présent, indépendants, voire concurrents sur de nombreux services, comme la téléphonie.

Telle est la prétention de l'IMS : quel que soit le réseau d'accès, faire émerger un réseau cœur unique, qui mutualise toutes les ressources, standardise les échanges entre les opérateurs, unifie les services des utilisateurs, simplifie la gestion des terminaux et facilite le développement de nouveaux services. C'est un pas nécessaire dont la grande majorité des acteurs reconnaissent la nécessité. C'est néanmoins une étape que peu d'acteurs osent encore franchir.

En effet, si son utilité n'est pas à démontrer, le business plan de l'IMS (et donc sa rentabilité) reste à étudier. Par exemple, la première des applications souvent mentionnées pour l'IMS est le service de présence, comme sur les messageries instantanées. Mais, avec ce simple service, les opérateurs peuvent perdre énormément d'appels, en particulier ceux qui aboutissent sur messagerie. Or l'infrastructure d'un réseau IMS a un coût relativement important pour les opérateurs qu'il faut, au contraire, rentabiliser.

Toutefois, les possibilités sont si grandes et le service rendu aux utilisateurs si important qu'il est assez probable qu'une nouvelle forme de consommation va émerger avec l'IMS. Nul doute qu'un tel réseau donnera une forte valeur ajoutée aux opérateurs qui le proposeront et que, très vite, les concurrents devront suivre le mouvement pour ne pas se laisser distancer sur ce terrain très prometteur.

# Partie IV

## Annexe



# Références

---

- J. R. ABRAHAMS, M. LOLLO, *CENTREX or PBX: The Impact of Internet Protocol*, Artech House, 2003
- T. ALLEN, M. CARLING, B. DUNSMORE, *Internetworking Technologies Handbook*, Cisco Press, 2003
- U. BLACK, *Voice Over IP*, Prentice Hall, 2002
- K. CAMP, *IP Telephony Demystified*, McGraw-Hill, 2002
- G. CAMARILLO, M. GARCIA-MARTIN, *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*, Wiley, 2006
- S. CHAKRABORTY, J. PEISA, T. FRANKKILA, P. SYNNERGREN, *IMS Multimedia Telephony over Cellular Systems: VoIP Evolution in a Converged Telecommunication World*, Wiley, 2007
- J. D. CIOARA, *Cisco IP Telephony*, Cisco Press, 2006
- J. DOHERTY, N. ANDERSON, *Internet Phone Services Simplified (VoIP)*, Cisco Press, 2006
- B. DOUSKALIS, *Putting VoIP to Work: Softswitch Network Design and Testing*, Prentice Hall, 2001
- L. DANG, C. JENNINGS, D. KELLY, *Practical VoIP Using VOCAL*, O'Reilly, 2002
- J. DAVIDSON, J. PETERS, M. BHATIA, S. KALIDINDI, S. MUKHERJEE, *Voice over IP Fundamentals*, Cisco Press, 2006
- D. COLLINS, *Carrier Grade Voice Over IP*, McGraw-Hill Professional, 2002
- J. F. DURKIN, *Voice-Enabling the Data Network: H.323, MGCP, SIP, QoS, SLAs, and Security*, Pearson Education, 2002
- D. ENDLER, M. COLLIER, *Hacking Exposed VoIP: Voice Over IP Security Secrets & Solutions*, McGraw-Hill, 2006
- M. FINNERAN, *Voice Over WLAN*, Newnes, 2008

- S. GHERNAOUTI-HÉLIE, A. DUFOUR, *Ingénierie des réseaux locaux d'entreprise et des PABX*, Masson, 1995
- R. GRIGONIS, *Voice Over DSL*, CMP Books, 2002
- D. GOMILLION, B. DEMPSTER, *Building Telephony Systems with Asterisk*, Packt Publishing, 2006
- W. C. HARDY, *VoIP Service Quality: Measuring and Evaluating Packet-Switched Voice*, McGraw-Hill, 2003
- L. HARTE, *Introduction to IP Telephony: Why and How Companies are Upgrading Private Telephone Systems to use VoIP Services*, Althos, 2006
- L. HARTE, D. BOWLER, *Introduction to SIP IP Telephony Systems: Technology Basics, Services, Economics, and Installation*, Althos, 2004
- O. HERSENT, J.-P. PETIT, D. GURLE, *Beyond VoIP Protocols: Understanding Voice Technology and Networking Techniques for IP Telephony*, Wiley, 2005
- O. HERSENT, D. GURLE, J.-P. PETIT, *La voix sur IP*, Dunod, 2005
- B. JACKSON, C. CLARK, *Asterisk Hacking*, Syngress, 2007
- K. JAYASWAL, *Data Centers Servers Storage and Voice over IP*, Wiley Administering, 2005
- A. JOHNSTON, *SIP: Understanding the Session Initiation Protocol*, Artech House, 2003
- A. JOHNSTON, D. M. PISCITELLO, *Understanding Voice over IP Security*, Artech House, 2006
- R. KAZA, S. ASADULLAH, *Cisco IP Telephony: Planning, Design, Implementation, Operation, and Optimization*, Cisco Press, 2005
- B. KHASNABISH, *Implementing Voice over IP*, Wiley, 2003
- P. MAHLER, *VoIP Telephony with Asterisk*, Signate, 2006
- M. A. MILLER, *Voice Over IP Technologies: Building the Converged Network*, Wiley, 2002
- D. MINOLI, *Voice Over IPv6: Architectures for Next Generation VoIP Networks*, Newnes, 2006
- M. NELSON, A. SMITH, D. DEEL, *Developing Cisco IP Phone Services: A Cisco AVVID Solution*, Cisco Press, 2002
- F. OHRTMAN, *Softswitch: Architecture for VoIP*, McGraw-Hill, 2002
- F. OHRTMAN, *Voice over 802.11*, Artech House, 2004
- A. PERRY, *Fundamentals of Voice-Quality Engineering in Wireless Networks*, Cambridge University Press, 2006

- T. PORTER, J. KANCKIRZ, A. ZMOLEK, *Practical VoIP Security*, Syngress, 2006
- J. PULVER, *The Internet Telephone Toolkit*, Wiley, 1996
- A. RAAKE, *Speech Quality of VoIP: Assessment and Prediction*, Wiley, 2006
- J. F. RANSOME, J. RITTINGHOUSE, *Voice over Internet Protocol (VoIP) Security*, Digital Press, 2004
- W. ROB, *Computer Telephone Integration*, Artech House, 1993
- W. ROB, *Computer Mediated Communications: Multimedia Application*, Artech House, 1995
- C. RIGAULT, *Principes de la commutation numérique: Du téléphone au multimédia*, Hermes, 1998
- T. RUSSELL, *The IP Multimedia Subsystem (IMS): Session Control and Other Network Operations*, McGraw-Hill, 2007
- S. SHEPARD, *IMS Crash Course*, McGraw-Hill, 2006
- H. SINNREICH, A. B. JOHNSTON, R. J. SPARKS, *SIP Beyond VoIP: The Next Step in the IP Communications Revolution*, VON Publishing LLC, 2005
- H. SINNREICH, A. B. Johnston, *Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol*, Wiley, 2006
- Z. SUN, *Satellite Networking Principles and Protocols*, Wiley, 2005
- M. STAFFORD, *Signaling and Switching For Packet Telephony*, Artech House, 2004
- A. SULKIN, *PBX Systems for IP Telephony*, McGraw-Hill, 2002
- P. THERMOS, A. TAKANEN, *Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures*, Addison-Wesley Professional, 2007
- J. G. VAN BOSSE, F. U. DEVETAK, *Signaling in Telecommunication Networks*, Wiley-Interscience, 2006
- J. VAN MEGGELEN, J. SMITH, L. MADSEN, *Asterisk: The Future of Telephony*, O'Reilly, 2005
- B. H. WALKE, S. MANGOLD, L. BERLEMANN, *IEEE 802 Wireless Systems: Protocols, Multi-Hop Mesh/Relaying, Performance and Spectrum Coexistence*, Wiley, 2007
- K. WALLACE, *Voice over IP First-Step*, Cisco Press, 2005
- T. WALLINGFORD, *Switching to VoIP*, O'Reilly, 2005
- T. WALLINGFORD, *VoIP Hacks, Tips & Tools for Internet Telephony*, O'Reilly, 2005
- R. WALTERS, *Computer Telephony Integration*, Artech House, 1998

## Liens web

### *Sites de vulgarisation de la ToIP*

<http://www.voip-info.org> (anglais)

Un site incontournable, qui s'impose par la diversité des articles proposés. Cette immense source d'informations mélange l'actualité de la ToIP et les tutoriaux dans tous les domaines de la VoIP. Le site s'adresse à tous les utilisateurs, du débutant, qui y trouvera les bases fondamentales pour commencer son approche, aux professionnels, qui y trouveront des ressources conséquentes pour approfondir leur étude.

<http://www.en.voipforo.com> (anglais)

De nombreuses fiches traitant de sujets divers, comme la numérisation de la voix, la qualité de service, les protocoles H.323 et SIP, les softphones ou encore le PBX Asterisk.

<http://www.voipplanet.com> (anglais)

Actualités, solutions, tests et tutoriaux pour les spécialistes de VoIP qui veulent connaître et approfondir les notions pratiques de la téléphonie.

<http://www.voipfr.org>

Un site non technique présentant des informations de manière claire et précise sur les évolutions du marché de la ToIP et des offres commerciales.

<http://www.testyourvoip.com> (anglais)

Un outil permettant de tester la qualité d'une communication téléphonique.

<http://www.multimedia.cx> (anglais)

Une base de liens dédiée au multimédia. Un wiki décrit de manière complète la liste des codecs audio et vidéo.

<http://www.01net.com>

Un site sur toutes les technologies informatiques traitées de bout en bout, de la promesse à l'actualité, avec des retours d'expérience sur le terrain. Un bon moyen de suivre les tendances de l'informatique en général et de la téléphonie en particulier, qui y est un thème largement couvert.

<http://www.reseaux-telecoms.net>

L'actualité informatique des télécoms, avec les évolutions du marché analysées par des experts.

<http://fr.wikipedia.org>

La célèbre encyclopédie en ligne, libre et gratuite, qui contient énormément d'informations et de sources sur les thèmes les plus divers de la téléphonie, depuis les protocoles jusqu'aux logiciels clients, en passant par les normes de codage et les principaux acteurs

du marché. Les fiches sur les protocoles H.323, SIP et Jabber sont une bonne première approche.

## Protocoles de ToIP

<http://www.itu.int> (anglais)

Le site de l'ITU (International Telecommunication Union)

<http://www.ietf.org> (anglais)

Le site de l'IETF (Internet Engineering Task Force)

<http://www.ietf.org/html.charters/sip-charter.html> (anglais)

La page de référence du protocole SIP, qui présente les travaux réalisés et en cours.

<http://www.packetizer.com/voip/> (anglais)

Une très bonne base de documents, classée par protocoles et organisée en sections (documents de spécification des standards, groupes de travail, mailing lists, forum de discussion, liens connexes, etc.)

<http://www.h323forum.org> (anglais)

Forum H.323 soutenu par l'IMTC (International Multimedia Telecommunications Consortium). On y trouve des discussions techniques autour des thèmes centraux relatifs à H.323, ainsi que la liste exhaustive des standards, incluant les annexes.

<http://www.sipforum.org> (anglais)

Le forum SIP, annonçant notamment les principaux événements internationaux concernant SIP.

<http://www.sipcenter.com> (anglais)

Un site couvrant tous les aspects de SIP, de ses spécifications protocolaires à ses évolutions futures, en passant par des outils de test d'interopérabilité d'une implémentation avec SIP, une comparaison avec H.323, son interaction avec MGCP ou encore le passage des pare-feu.

<http://www.cs.columbia.edu/sip/> (anglais)

Un site très complet sur les acteurs promouvant SIP, avec un historique exhaustif de la normalisation, des outils de tests, d'analyse et de performance du protocole SIP, ainsi que de très nombreux liens, tutoriaux et informations pratiques.

<http://www.tech-invite.com> (anglais)

Portail technique dédié au protocole SIP et à l'IMS.

<http://www.xmpp.org> (anglais)

Liste des travaux et documents de travail sur XMPP (Extensible Messaging and Presence Protocol), le protocole mis en œuvre dans la plate-forme Jabber.

<http://www.3gpp.org> (anglais)

Site Web du 3GPP, qui a pour objectif de standardiser l'IMS en partant de la technologie GSM (en Europe).

<http://www.3gpp2.org> (anglais)

Site Web du 3GPP2, qui a pour objectif de standardiser l'IMS en partant de la technologie CDMA 2000 (aux États-Unis et en Asie).

## **Softphones et dérivés**

<http://www.voipproducts.eu> (anglais)

L'actualité des softphones et une comparaison, actualisée régulièrement, des tarifs proposés. On y retrouve, entre autres, les incontournables WLM, Yahoo! Messenger Voice et Google Talk.

<http://www.wengophone.fr/index.php>

Site de téléchargement du logiciel Wengo, avec les sources du logiciel ouvert.

<http://www.skype.com>

Site de téléchargement du logiciel Skype.

<http://developer.skype.com/> (anglais)

Cette page recense les informations permettant aux développeurs d'intégrer certaines fonctionnalités de Skype à leur application, et d'en concevoir de nouvelles avec les API fournies.

<http://www.windowlive.fr/messenger/>

Site de téléchargement du logiciel WLM de Microsoft.

<http://fr.messenger.yahoo.com>

Site de téléchargement du logiciel Yahoo! Messenger.

<http://developer.yahoo.com/messenger/> (anglais)

Téléchargement et documentation du kit de développement (ou SDK) permettant aux développeurs indépendants de concevoir leurs propres outils et plug-in pour Yahoo! Messenger.

<http://www.google.com/talk/intl/fr/>

Site de téléchargement du logiciel Google Talk.

<http://www.jabber.org> (anglais)

Le site de la Fondation Jabber. Il contient des listes très complètes de clients ou de serveurs compatibles avec la plate-forme Jabber. Le site contient également des

bibliothèques de codes source de l'implémentation du protocole. On y trouve aussi des explications sur le protocole XMPP, et les origines et ambitions de la Fondation.

<http://www.jabberfr.org>

Un site très complet et résolument pratique qui permet de se familiariser avec Jabber. Aux débutants comme aux experts, ce site propose des tutoriaux très clairs ainsi qu'un forum pour s'entraider et une liste de liens pour poursuivre son étude.

<http://www.ekiga.org> (anglais)

Site de téléchargement du logiciel Ekiga (anciennement GnomeMeeting), l'une des applications libres et gratuites les plus réputées pour la gestion des conférences multimédias. Le logiciel supporte à la fois le protocole H.323 et SIP.

<http://www.counterpath.com> (anglais)

Site de téléchargement du logiciel X-Lite permettant de communiquer en utilisant le protocole SIP. Le softphone émule un terminal téléphonique et intègre des fonctions classiques, comme l'accès à sa messagerie pour peu que la configuration avec le serveur soit paramétrée.

<http://www.sjlabs.com> (anglais)

SJPhone, le softphone compatible H.323 et SIP très riche en fonctionnalités et disposant de guide d'utilisation pour utilisateurs débutants et avancés.

<http://www.bewip.com>

Softphone à installer sur des appareils communicants, de type PDA ou SmartPhone, et permettant de sélectionner intelligemment le réseau GSM ou Wi-Fi selon les disponibilités et tarifs.

<http://gizmoproject.com/intl/fr/>

Un softphone qui permet la mise à disposition d'une boîte vocale pour les utilisateurs, et l'avantage d'utiliser le standard SIP, permettant notamment de l'utiliser conjointement avec le logiciel Asterisk.

<http://www.jajah.com>

Ce site propose un moyen original de téléphoner. L'utilisateur saisit sur une page Web son numéro de téléphone puis celui de son correspondant. Les deux téléphones sonnent et sont mis en relation pour que la communication débute. En réseau cœur, c'est la technologie IP qui est utilisée, tandis qu'aux extrémités, une passerelle assure la conversion du signal vers un réseau téléphonique classique.

<http://www.mylivio.com>

Ce nouveau venu dans le paysage du Web propose aux webmasters d'intégrer à leurs pages Web des boutons d'appel qui permettent aux visiteurs de leur téléphoner, que ce soit sur un téléphone fixe ou un portable. Le service est totalement gratuit et financé par la publicité.

## PBX Asterisk

<http://www.asterisk.org> (anglais)

Le site de référence pour télécharger le logiciel Asterisk ainsi qu'un ensemble de modules additionnels et installer le tout sur une distribution quelconque. La section Support fournit une source d'information très utile lorsqu'on rencontre des difficultés avec Asterisk.

<http://www.voip-info.org/wiki/index.php?page=Asterisk>

Une base très complète et parfaitement documentée, avec de nombreux tutoriaux, sur des aspects très divers de la configuration à l'optimisation du serveur Asterisk. Ce site donne également accès au manuel des commandes d'Asterisk. À ce titre, le moteur de recherche est particulièrement utile pour connaître ou vérifier la syntaxe exacte des commandes

<http://www.asterisknow.org> (anglais)

Asterisk Now est une distribution complète pour installer un serveur Asterisk en un temps record de trente minutes, avec des interfaces de configuration et des outils de gestion du serveur.

<http://www.asteriskguru.com> (anglais)

Mis à jour assez fréquemment, ce site propose des tutoriaux pour se familiariser avec Asterisk ou approfondir certaines notions. De nombreuses informations sont fournies sur les nouveautés apportées au logiciel. Des utilitaires sont disponibles pour améliorer le fonctionnement d'Asterisk. Un forum très actif permet de trouver de l'aide en cas de problème.

<http://www.asterisk-france.net>

Des informations diverses sur l'actualité du logiciel phare Asterisk mais aussi des conseils et explications techniques et pratiques pour comprendre et optimiser l'utilisation du logiciel. On y trouve également des tutoriaux pour l'installation et les paramètres de fonctionnement d'Asterisk. Un forum d'entraide est proposé.

<http://www.freepbx.org> (anglais)

Un outil graphique pour paramétrer Asterisk sous Linux, selon ses besoins et grâce à des interfaces Web. La documentation fournie est dense et couvre tous les aspects pratiques.

<http://www.digium.com> (anglais)

Le site de la société mère du logiciel Asterisk.

<http://www.eikonex.net>

Site de formation au logiciel Asterisk et de vente de matériel compatible avec le produit et adapté à différents besoins.

<http://www.sipfoundry.org> (anglais)

SipX, une solution de rechange à Asterisk très robuste mais dédié exclusivement à SIP.

---

## ***Salons sur la VoIP en France***

*<http://www.conventionvoip.com>*

*<http://www.voipexpo.fr>*

Salons présentant les enjeux, solutions et acteurs de la ToIP.



# Index

---

## Numériques

3Com 131, 307  
3GPP (Third Génération Partnership Project) 419  
3GPP2 (Third Génération Partnership Project 2) 419  
4ESS 308  
9Box 373

## A

A Record (Address Record) 52  
AAL1 (ATM Adaptation Layer de type 1) 187  
AAL5 (ATM Adaptation Layer de type 5) 364  
ACR (Absolute Category Rating) 36  
AD-PCM (Adaptive Differential-Pulse Code Modulation) 32  
adressage SIP 94  
ADSL  
  accès xDSL 359  
  ATM 361  
  concentrateur 363  
  dégroupeage 362  
  DSLAM 362  
  Ethernet 361, 362  
  HDSL 360  
  L2TP 365  
  Lite 361  
  modem  
    G-Lite 361  
    VDSL 366  
    xDSL 360  
  parole et vidéo sur xDSL 366  
  POP (Point of Presence) 363  
  protocoles 363  
  Quadruple-Play 372

RADSL 360  
  répartiteur 363  
SDSL 360  
trame  
  ATM 364  
  Ethernet 363  
  PPP 364  
Triple-Play 16  
U-ADSL 361  
VDSL 360  
VoDSL 367  
ADSL (Asymmetric Digital Subscriber Line)  
  *Voir* ADSL  
affichage du numéro 50  
AGI (Asterisk Gateway Interface) 352, 355  
Ajax (Asynchronous JavaScript and XML) 229  
Alcatel 48, 131, 307  
ALG (Application Layer Gateway) 389  
AMR-WB (Adaptative Multi-Rate-WideBand) 33  
APINC (association pour la promotion de l'Internet non commercial) 288  
architecture IMS 415  
architectures physiques 183  
ASA (Automatic Service Agent) 268  
Asterisk 305  
  AGI (Asterisk Gateway Interface) 352, 355  
  améliorer les services téléphoniques 311  
  assurer le nomadisme des utilisateurs 311

cible et usage 309  
compatibilité 308  
conférences 347  
configuration 318  
  catégories d'éléments 318  
  définition des utilisateurs 321  
  organisation des fichiers 319  
  plan de numérotation 324  
décompresser les sources 314  
directive d'inclusion 339  
fichier  
  asterisk.conf 319  
  extensions.conf 324  
  h323.conf 321  
  iax.conf 321, 324  
  mgcp.conf 321  
  sip.conf 321  
  skinny.conf 321  
  voicemail.conf 349  
fonctionnalités 308  
IAX (Inter Asterisk eXchange) 308, 356  
installation 311  
interface  
  FXO 313  
  FXS 313  
IVR (Interactive Voice Response) 308  
lancement du serveur 315  
logique de programmation 339  
mise en œuvre de la plateforme 312  
modules du logiciel 314  
optimisation  
  des traitements 338  
  du routage avec les contextes 342

- présentation 307
  - problèmes éventuels avec les modules 345
  - procédure de call-back 310
  - réduire les coûts en appelant de l'extérieur au tarif domestique 309
  - se connecter
    - au réseau téléphonique traditionnel 312
    - en mode client 316
  - serveur 316
  - service de messagerie audio 349
  - sous Windows 356
  - standard vocal automatique 346
  - télécharger les composants utiles 313
  - tester la configuration d'un client 336
  - Trixbox 355
- ATM** 187
- avantages de la ToIP 10
- Avaya 307
- AVT (Audio Video Transport) 41
- AVT-WG (Audio Video Transport-Work Group) 161
- B**
- BAS (Broadband Access Server) 364
- BellCoRe 131
- BGCF (Breakout Gateway Control Function) 430
- BGP (Border Gateway Protocol) 178
- Bluetooth 372
- boucle locale 15
  - haut débit 359
- Bouygues Telecom (iMode) 264
- BRI (ISDN4Linux) 308
- BT (British Telecom) 372
  - Fusion 412
- C**
- CATV
  - IEEE 802.14 369
  - MCNS-DOCSIS 369
  - modem câble 367
  - multiplexage en fréquence 368
  - plages de fréquences 369
- CCMP (Counter with CBC MAC Protocol) 215
- CDMA 2000 419
- CELP (Code Excited Linear Prediction) 32
- Centrex 11, 187
- CFM (convergence fixes mobiles) 412
- Cisco 46, 48, 307
  - SCCP (Cisco Skinny) 308
  - SGCP 131
  - SIP Proxy Server 94
  - Vocal 357
- CLERC (Competitive Local Exchange Carrier) 366
- client de messagerie Web 229
- CLIP (Connected Line Identification Presentation) 50
- CLIR (Connected Line Identification Restriction) 50
- codage 27, 30
- codec (codeur-décodeur) 7, 8
- Colt 240
- combiné hybride GSM/Wi-Fi 227
- conférence
  - H.323 64
  - multimédia 64
- contrôle 19, 40, 157, 410
  - architecture d'un environnement de contrôle « autonome » 410
  - au niveau réseau 167
  - d'erreur 158
  - de congestion 158
  - de flux 158
  - de séquence 158
- convergence des services 221
- Conversagent 268
- CoolTalk 46
- COPS (Common Open Policy Service) 357
- CounterPath 336
- CSCF (Call State Control Function) 425
- D**
- débit (caractéristiques du) 37
- dégroupage 362
- délai de transit 196
- dépaquetisation 5
- DHCP (Dynamic Host Configuration Protocol) 8
- Dialpad 275
- Diameter 439
- DMS 421
- DiffServ 167, 170, 411
  - CATV 370
- Digium 307, 313, 356
- disponibilité 18, 406
- DMS100 308
- DNS (Domaine Name Server) 51
- Double-Play 16
- DTMF (Dual-Tone Multi-Frequency) 49
- E**
- E&M 308
  - Wink 308
- EAP (Extensible Authentication Protocol) 402
- Early H.245 81
- eBay 238
- échantillonnage 27, 28
- écho 7, 23
  - suppresseur 7
- EFM (Ethernet-in-the-First-Mile) 362
- EFMF (EFM Fiber) 362
- Eikonex 307
- endpoint 54
- ENUM (tElephone NUmber Mapping) 52
- E-PON (Ethernet Passive Optical Network) 371
- Ericsson 48, 131
- Ethereal 278
- Ethernet 183, 362
- EuroDOCSIS 370
- EuroISDN 308
- F**
- FastConnect 48, 82
- FAX (Fax over IP) 41
- fibre optique 362, 370
  - E-PON 371
  - France Télécom 371
  - Free 371
  - FTTB 370
  - FTTC 371
  - FTTH 370
- filtrage des flux 375

- Finarea SA 228  
 FMC (Fixed Mobile Convergence) 412  
 Frame Relay 189  
 France Télécom 363  
   fibre optique 371  
   Orange 294  
   Quadruple-Play 372  
   Orange Messenger by Windows Live 264  
   Unik 227  
 Free 128, 229, 371, 372  
   compte SIP 229  
   Freephonie 229  
 FreeBox 372  
 FTTB (Fiber To The Building) 370  
 FTTC (Fiber To The Curb) 371  
 FTTH (Fiber To The Home) 370  
 FXO (Foreign eXchange Office) 313  
 FXS (Foreign eXchange Subscriber) 313
- G**
- GAN (Generic Access Network) 412  
 gatekeeper 8, 50, 57  
   alternatif et affecté 83  
 GEF (Generic Extensibility Framework) 50  
 gestion des réseaux 19, 408  
   NetConf 409  
   PBM (Policy-based Management) 409  
 gigue 196  
 Google 220, 238, 243  
   Calendar 296  
   Chat 296  
   Mail 296  
   Talk 281, 295  
   ajouter des contacts 298  
   avec un logiciel alternatif 302  
   gestion des connexions multiples 302  
   intégration avec le service Google Mail 299  
   Jabber/XMPP 297  
   lancer une communication 298  
   utilisation 297  
 GR-303 308
- Groundstart 308  
 GSM 52, 372  
 GSM/Wi-Fi 227
- H**
- H.225.0 68  
   signalisation d'appel et d'enregistrement 69  
 H.235 52  
 H.235.x 79  
 H.245 68  
   signalisation de contrôle de connexion 75  
   tunneling 82  
 H.248 51  
 H.248/MeGaCoP 131  
 H.264 52  
 H.26x 79  
 H.323 42, 43, 308, 356  
   architecture et fonctionnalités 53  
   conférences 64  
   contrôleur et exécutant 65  
   modes de diffusion 66  
   enregistrement d'un terminal auprès d'un gatekeeper 73  
   exemple de scénario d'une communication complète 79  
   gatekeeper 53, 57  
   alternatif et affecté 83  
   localisation d'un terminal 74  
 MCU 53, 64  
 messages 68  
 NAT 384  
 normalisation UIT 46  
 normes d'interopérabilité 47  
 passerelle 53, 60  
 pont multipoint 53  
 principaux protocoles 78  
 procédure  
   Early H.245 81  
   FastConnect 82  
   H.245 tunneling 82  
 sécurité 82  
 signalisation d'appel 69  
   d'enregistrement 71  
   de contrôle de connexion 75  
 système 54  
 terminal 55  
 versions 48  
 zone 54  
 H.325 155  
 H.361 52  
 H.450.x 49, 79  
 H.460 52  
 H.460.x 79  
 H.510 79  
 H.GCP (Gateway Control Protocol) 131  
 HDSL (High bit rate DSL) 360  
 HSS (Home Subscriber Server) 425  
 HTTP (HyperText Transfer Protocol) 87
- I**
- IANA (Internet Assigned Numbers Authority) 377  
 IAX (Inter Asterisk eXchange) 308, 356  
 iBasis 240  
 IBM 46, 294  
 ICE (Interactive Connectivity Establishment) 393  
 ICMP (Internet Control Message Protocol) 212  
 IDCP (Internet Device Control Protocol) 131  
 IEEE 802.11 198  
 IEEE 802.11e 202  
 IEEE 802.14 369  
 IEEE 802.16 205  
 IEEE 802.1p 370  
 IEEE 802.1x 214, 400  
 IEEE 802.ah 362  
 iLBC 52  
 ILEC (Incumbent Local Exchange Carrier) 366  
 IMPP (Instant Messaging and Presence Protocol) 128, 285  
 IMS (IP Multimedia Subsystem) 415  
   architecture 422  
   commandes Diameter 439  
   communications 440  
   extensions du protocole SIP 437  
   historique 419  
   identités 435  
   motivations et objectifs 416  
   notions de réseau visité et de roaming 421  
   protocoles utilisés 420  
   serveurs d'applications 434

IM-SSF (IP Multimedia-Service Switching Function) 434  
 IMT-2000 (International Mobile Télécommunications 2000) 419  
 IMTC (International Multimedia Teleconferencing Consortium) 41  
 ingénierie de trafic 177  
 iNOW! Consortium 48  
 intégration voix-données 183  
 Intel 46, 161  
 Internet Box 15  
 Internet Multimedia Conferencing Suite 87  
 introduction aux PBX 305  
 IntServ 167, 168  
 IP multicast 367  
 IPTel (IP Telephony) 41  
 IRC (Internet Chat Relay) 223  
 ISAKMP (Internet Security Association and Key Management Protocol) 214  
 ISSLL (Integrated Services Over Specific Link Layers) 170  
 IVR (Integrated Voice Responder) 346

**J**

Jabber 281  
 ajouter des contacts WLM ou Yahoo! 293  
 architecture 282  
 choix  
 du client 289  
 du serveur 287  
 configuration d'un client 290  
 coopération entre serveurs 294  
 Erlang Jabber Daemon (ejabberd) 289  
 Google Talk 297  
 JID (Jabber ID) 288  
 Jingle 286  
 monter son propre serveur avec ejabberd 289  
 serveurs connus 288  
 services complémentaires 293  
 utilisation 287  
 XEP 282, 285  
 XMPP 284  
 JavaScript 296  
 Jingle 287

JSF (Jabber Software Foundation) 282

**K**

Kewlstart 308  
 Kopete 287

**L**

L2F (Layer 2 Forwarding) 365  
 L2TP (Layer 2 Tunneling Protocol) 365  
 latence 7, 23  
 LCS (Live Communications Server) 128  
 LDP (Label Distribution Protocol) 177, 178  
 Level 3 Communications 131, 240  
 LinPhone 336  
 LiveBox 372  
 LMA (Licenced Mobile Acces) 412  
 localisation des abonnés 57  
 Loopstart 308  
 Lotus 128  
 LPC (Linear Predictive Coding) 32  
 Lucent 48, 131  
 Lucent 5E 308

**M**

masquage du numéro 50  
 MCI 263  
 MCNS-DOCSIS (Multimedia Cable Network System-Data Over Cable Service Interoperability Specification) 369  
 MCU (Multipoint Control Unit) 53, 64  
 MDCP (Media Device Control Protocol) 131  
 Meebo 229  
 MeGaCo (Media Gateway Control) 42, 131  
 message en attente 50  
 messagerie audio 349  
 MF and DTMF 308  
 MFC-R2 308  
 MGCP 129, 308  
 architecture et fonctionnement 132  
 Call Agent 133

exemple d'utilisation chez les FAI 137

passerelles multimédias 134  
 avantages et inconvénients 138  
 H.248/MeGaCoP 131  
 héritages protocolaires 132  
 historique 130  
 messages 142  
 adressage des endpoints 142  
 identifiant de transaction 144  
 ligne d'état 147  
 paramètres  
 d'en-tête 145  
 généraux pour les requêtes et les réponses 145  
 réponses 151  
 requêtes 147  
 NAT 384  
 principes d'établissement d'une communication 139  
 MGCP (Media Gateway Control Protocol) 50, 87  
*Voir* MGCP  
 MIC (modulation, impulsion et codage) 28, 56  
 Microsoft 46, 128, 161, 220, 238, 243  
 Encarta Réponses Instantanées 268  
 LCS (Live Communications Server) 128, 260  
 MSN (MicroSoft Network) 260  
 partenariat avec Yahoo! 277  
 Passport .Net 260  
 Robots Contest 268  
 Windows Live Messenger 259  
 Windows Live Services 260  
 MIME (Multipurpose Internet Mail Extension) 87  
 mise en attente 50  
 MMUSIC (Multiparty Multimedia Session Control) 41  
 mobilité 196  
 mobilité/SIP 128  
 modem ADSL 11  
 MOS (Mean Opinion Score) 36  
 MPLS (MultiProtocol Label-Switching) 177, 411  
 MRF (Multimedia Resource Function) 429  
 MSN 16

- MTTF (Mean Time To Failure) 407
- MTTR (Mean Time To Repair) 407
- multicast 56, 67
- N**
- NAP (Network Access Provider) 363
- NAPT (Network Address Port Translation) 381, 383
- NASS (Network Attachment Sub-System) 424
- NAT 52, 239
  - adresses privées et publiques 376
  - avantages 379
  - catégories de 381
  - dynamique 381
  - filtrage applicatif des données 388
  - gestion par le client 391
  - mécanisme de 376
  - méthode
    - STUN 391
    - TURN 393
    - UPnP 391
  - NAPT 383
  - partager une adresse IP privée 377
  - passerelles de niveau applicatif 389
  - plate-forme ICE 393
  - problèmes engendrés 384
  - protocoles sensibles au 384
  - recevoir une connexion derrière un NAPT 384
  - RSIP (Realm Specific Internet Protocol) 394
  - sécurité 385
  - statique 381
  - tunneliser les applications 389
- NAT (Network Address Translation)
  - Voir* NAT
- National ISDN2 308
- NetConf 409
- Netfilter 389
- Netscape 46
- Neuf Cegetel 225
- Neuf Telecom 128, 372
- News Corporation 238
- NFAS 308
- nomadisme 11, 196, 311
- Nortel 131, 307
- notification d'appel 50
- NPT (Network Port Translation) 381
- NTP (Network Time Protocol) 163
- NTT 371
- O**
- OMA (Open Mobile Alliance) 420
- OpenWengo 286
- opérateur
  - alternatif 13, 363
  - câblo-opérateur 367
  - historique 13, 366
- Orange
  - bi-mode GSM/Wi-Fi 372
  - UNIK 412
- OSA-SCS (Open Service Access-Service Capability Server) 435
- OSPF (Open Shortest Path First) 177, 178
- P**
- PABX (Private Automatic Branch eXchange) 183, 305
- paquetisation 5
- pare-feu 52, 239
  - avec état (statefull) 386
  - passage des 386
  - sans état (stateless) 386
- Partysip SIP Proxy Server 93
- passage des pare-feu 386
- passerelle 9, 60
  - architecture 63
  - de niveau applicatif 389
  - H.323 62
  - MGCP 132
  - multimédia (MGCP) 129, 134
  - normes d'interconnexion de réseaux 63
  - RSIP 394
  - SIP 94
  - softphones 222
- PBM (Policy-based Management) 408
- PBX-IP 11, 187
  - Asterisk 305
  - SIP-X 358
  - Vocal 357
- PCM (Pulse Code Modulation) 28, 55
- peer-to-peer 237, 360
  - Skype 238
- Penta-Play 17
- Philips 264
- Phone Gaim 91
- Pingtel 358
- PINT (PSTN IP Internetworking) 41
- PON (Passive Optical Network) 362, 366
- PPP (Point-to-Point Protocol) 361
- PPPoE (PPP over Ethernet) 364
- PPTP (Point-to-Point Tunneling Protocol) 365
- présence (service de) 11
- protocoles de transport 157
- PSI 287
- Q**
- Q.931 (signalisation d'appel) 69
- Quadruple-Play 17, 372
  - Bluetooth 372
  - FreeBox 372
  - GSM 372
  - LiveBox 372
  - Neuf Telecom (TWIN) 372
  - téléphone bi-mode GSM/Wi-Fi 372
- qualité de service 19, 35, 157, 197, 411
  - contrôle et protocoles de transport 157
  - contrôles au niveau réseau 167
  - DiffServ 170, 411
    - AF (Assured Forwarding) 173
    - architecture d'un nœud 175
    - EF (Expedited Forwarding) 172
  - ingénierie de trafic 177
  - IntServ 168
  - RTP et RTCP 161
  - TCP et le transport de données multimédias temps réel 158
  - UDP et le transport de données multimédias temps réel 160
- quantification 27

## R

RACS (Ressource Admission Control SubSystem) 424  
 RADIUS (Remote Authentication Dial-In User Server) 215  
 RADSL (Rate Adaptive DSL) 360  
 RAS (Registration Admission Status) 69, 71  
 redirection d'appel 49  
 redondance 407  
 relais de trames 189  
 resynchronisation de la parole téléphonique 26  
 Robbed-bit Signaling (RBS) 308  
 RSIP (Realm Specific Internet Protocol) 394  
 RSVP (Resource reSerVation Protocol) 87, 178  
 RSVP-TE (RSVP-Traffic Engineering) 180  
 RTCP (Real-time Transport Control Protocol) 78, 87, 166  
 catégories de paquets 166  
 RTP (Real-time Transport Protocol) 9, 51, 78, 87, 161  
 exemple de mise en paquet et overhead 163  
 extensions et limitations 165  
 fonctionnalités 161  
 format des paquets 162  
 RTSP (Real-time Streaming Protocol) 87

## S

salon virtuel de conférence 347  
 sans-fil 195  
 SAP (Session Advertisement Protocol) 87  
 SASL (Simple Authentication and Security Layer) 302  
 SBC (Sub-Band Coding) 32  
 scalabilité 169  
 SCTP (Stream Control Transmission Protocol) 52  
 SDP (Session Description Protocol) 87, 107  
 SDSL (Symmetric DSL) 360  
 sécurité 18, 82, 183, 209, 400  
 attaques 210  
 authentification 400  
 confidentialité et intégrité 406

infrastructures de 214  
 téléphonie par Wi-Fi 215  
 service de présence 11  
 Jabber 284  
 SGCP (Simple Gateway Control Protocol) 131  
 SGW (Signaling GateWay) 432  
 Siemens 48, 277, 307  
 signalisation 42  
 d'appel 69  
 d'enregistrement 71  
 filtrage des flux 375  
 H.323 43  
 hors bande 86  
 messages H.323 68  
 routée et directe 59  
 SIP 86  
 SIGTRAN (Signal Translation) 42  
 SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) 128, 284  
 SIP 85, 308  
 adressage 94  
 exemples d'adresses 97  
 format des adresses 96  
 localisation et résolution d'une adresse 98  
 URI 95  
 architecture 89  
 clients 91  
 compatibilité 86  
 enregistrement d'un terminal 122  
 Free 229  
 IMS 420  
 initialisation  
 d'appel directe 125  
 d'une communication avec un serveur proxy 122  
 d'une communication directe 120  
 LinPhone 336  
 localisation  
 de l'abonné 91  
 par un serveur de redirection 125  
 messages 100  
 abréviation des en-têtes 105  
 champ VIA pour détecter les boucles lors du routage 103  
 corps 106

différence entre Call-Id et CSeq 104  
 en-têtes 101  
 exemple complet commenté 109  
 méthodes d'extension du protocole 115  
 paramètres généraux pour les requêtes et les réponses 101  
 réponses 116  
 requêtes 112  
 transaction 100  
 mise en place de serveurs 93  
 mobilité 128  
 modification d'une communication 126  
 modularité 87  
 NAT 384  
 passerelles 94  
 scénarios de communication 120  
 SDP (Session Description Protocol) 107  
 se connecter à des réseaux non-IP 94  
 serveur  
 d'enregistrement 91  
 de localisation 91  
 de redirection 92  
 proxy 92  
 simplicité 88  
 SJPhone 336  
 standardisation 85  
 terminaison d'une communication 127  
 terminal 90  
 UAC (User Agent Client) 90  
 UAS (User Agent Server) 90  
 Wengo 225  
 X-Lite 336  
 SIP (Session Initiation Protocol) 9, 42  
*Voir* SIP  
 SIP AS (SIP Application Server) 434  
 SIP Express Router 93  
 SIPFoundry 358  
 SIP-X 358  
 SJPhone 336  
 Skype 16, 220, 237, 345  
 aller plus loin 247

- architecture 238
    - limiter les ressources 239
    - traverser les pare-feu 239
  - commandes textuelles 253
  - intégration dans ses pages Web et e-mails 254
  - norme d'interopérabilité 241
  - offres 240
    - partenariats technologiques et commerciaux 241
  - options en ligne de commande 252
  - outils 246
    - indicateur de présence 246
    - salons de discussions 247
    - transfert de fichiers 247
  - ouvrir plusieurs instances 248
  - peer-to-peer 238
  - recommandations et résolution de problèmes 256
  - sécurité 242
  - utilisation 243
    - appeler 244
    - prérequis 243
  - utiliser une image de statut 255
  - SNMP (Simple Network Management Protocol) 149
  - SoftBank 371
  - softphone 219
    - Beautiful Phone 227
    - clients de messagerie Web 229
    - en entreprise 225
    - Finarea SA 228
    - Google Talk 281
    - introduction 220
    - Jabber 281
    - LinPhone 336
    - services proposés 220
      - liste de contacts, présence et disponibilité 223
      - messagerie instantanée 223
      - téléphonie 221
      - vidéo et transfert de fichiers 224
    - SJPhone 336
    - Skype 220, 237
    - téléphoner gratuitement d'un PC vers un téléphone fixe 228
    - WengoPhone 225
    - X-Lite 336
  - SOX 345
  - SPIM (SPam over Instant Messaging) 224
  - SSL (Secure Sockets Layer) 215
  - standard vocal automatique 346
  - STUN (Simple Traversal of UDP through NAT) 391
  - Sun 294
  - suppresseur d'écho 7, 24
- T**
- T.120 79
  - T.38 79
  - Telcordia 131, 407
  - Teleglobe 240
  - téléphonie
    - chez les fournisseurs d'accès 359
    - numérique 27
    - par ADSL 220
    - par circuit 4, 8
    - par paquets 4
    - problématique de base 7
    - sur ATM 187
      - AAL2 187
    - sur CATV 367
    - sur Ethernet 183
      - intégration voix-données 183
    - sur fibre optique 370
    - sur le relais de trames 189
    - sur Quadruple-Play 372
    - sur réseaux sans fil 195
      - contraintes 195
      - IEEE 802.11 198
      - qualité de service 197, 202
    - sur softphone 220
    - sur WiMax 205
  - temps
    - de latence 23, 196
    - de synchronisation 26
    - réel 40
  - Ten 264
  - théorème d'échantillonnage 27
  - TINS (Transport for Initiating and Negotiating Sessions) 287
  - TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks) 41, 48, 131
  - TISPAN (Telecommunications and Internet converged Services and Protocols for Advanced Networking) 419
  - TKIP (Temporal Key Integrity Protocol) 215
  - TLS (Transport Layer Security) 302
  - ToDSL (Telephony over DSL) 366
  - ToWLAN (Telephony over Wireless LAN) 195
  - transfert d'appel 49
  - translation d'adresses 379
  - transport de données multimédias temps réel 158
  - traverser les pare-feu 239
  - Triple-Play 16, 137
  - Trixbbox 355
  - TURN (Traversal Using Relay NAC) 393
  - TWIN 373
- U**
- U-ADSL (Universal ADSL) 361
  - UMA (Unlicensed Mobile Access) 372, 412
  - unicast 66
  - Unik 227, 372
  - UPnP (Universal Plug-and-Play) 391
  - UTF-8 (Unicode Transformation Format) 88
- V**
- V.150.1 79
  - VDSL (Very high bit rate DSL) 360, 366
  - Verizon Communications (MCI) 263
  - vidéo mobile 17
  - VoATM (Voice over ATM) 367
  - Vocal (Vovida Open Communication Application Library) 357
  - VocalTec 48
  - VoD (Video on Demand) 367
  - VoDSL (Video over DSL) 367
  - Vovida 357
  - VPN (Virtual Private Network) 184, 389
  - VTOA (Voice and Telephony Over ATM) 187

**W**

- Web 2.0 229
- Wengo 91, 345
- WengoPhone 225
  - dialoguer entre logiciels concurrents 226
- WEP (Wired Equivalent Privacy) 215
- WG MMUSIC (Work Group Multiparty Multimedia Session Control) 86
- Wi-Fi 21, 195, 215
  - Quadruple-Play 372
- WiMax 17, 21, 205
  - classes de services pour la ToIP 208
  - fixe 205
  - format de la trame 208
- WiMax-Mobile 207
- WiMedia 21
- Wi-Mobile 207
- Windows Live Call 263
- Windows Live Messenger 220, 259, 261
- afficher des images de ses contacts 265
- ajouter
  - un contact 266
  - un robot intelligent 268
- censure automatique 269
- exporter et importer sa liste de contacts 267
- extensions et améliorations 270
- fonctionnalités 263
- gamme de services unifiés Live 260
- intégrer WLM sur ses pages Web 264
- partenariat Microsoft-Yahoo! 277
- périphériques compatibles 264
- service de téléphonie 263
- utilisateurs bloqués et hors connexion 267
- utilisation 262
- Web Messenger 270
- Windows Live Contacts 266
- WiRAN (Wireless Regional Area Network) 373
- Wireshark 278
- WLM
  - Voir* Windows Live Messenger
- WLS (Windows Live Services) 260

**X**

- X.680 79
- X.691 79
- XCOM Technologies 131
- XEP (XMPP Enhancement Proposals) 282, 285
- X-Lite 336
- X-Lite Free 91
- XMPP (eXtensible Messaging and Presence Protocol) 284
- XTEN 336

**Y**

- Yahoo! 220, 238, 243
  - Yahoo! Messenger 271

# Téléphonie sur IP

2<sup>e</sup> édition



**Laurent Ouakil**

**Laurent Ouakil** est Docteur de l'Université Pierre et Marie Curie (Paris 6). Ingénieur puis chef de projet dans l'industrie depuis plusieurs années, il a encadré d'importants projets de téléphonie sur IP. Il est auteur de nombreuses publications scientifiques et enseigne les réseaux et télécoms à l'Université Paris 6 et à l'Université Paris 8. Ses travaux portent sur les plateformes de développement de services de téléphonie sur IP, sur la traversée des flux téléphoniques et les évolutions du multimédia de demain.

**Guy Pujolle**

**Guy Pujolle** est Professeur à l'Université Pierre et Marie Curie (Paris 6) et responsable de nombreux grands projets de recherche français et européens. Auteur de plus de cent articles et de nombreux ouvrages en langues française et anglaise, il est également membre du conseil scientifique du groupe Orange. Ses recherches portent actuellement sur la conception et le développement des réseaux post-IP. Il est cofondateur de la société GoSMOS, spécialisée dans la qualité de service, de la société Ucopia, qui commercialise un logiciel de gestion du nomadisme, de Ginkgo-Networks, qui développe un système de pilotage de réseau, d'EtherTrust, spécialiste de la très haute sécurité à l'aide de cartes à puce, et de VirtuOR, qui se focalise sur la virtualisation de réseau.

## Technologies et solutions de téléphonie sur IP

La téléphonie sur IP s'impose progressivement dans tous les secteurs : la convergence vers un réseau tout IP s'accélère dans les entreprises, les fournisseurs d'accès généralisent leurs offres triple-play et quadruple-play incluant le service de téléphonie sur IP, des logiciels comme Skype, WLM, Yahoo! Messenger ou Google Talk sont entrés dans les habitudes, sans parler de la téléphonie mobile qui devient hybride et s'adonne aux bienfaits du réseau IP.

Ce livre offre un vaste panorama des technologies et des solutions de téléphonie sur IP. Qu'apporte ce modèle par rapport au réseau téléphonique traditionnel? Quels sont les protocoles mis en œuvre? Comment garantir la qualité de service, la sécurité et le nomadisme? Quelles sont les difficultés rencontrées et comment les contourner? Quelles sont les architectures types à déployer en entreprise? Quels sont les logiciels grand public qui proposent ce type de service et comment les utiliser? Comment installer et maintenir gratuitement son propre PBX avec Asterisk? Que nous réserve la téléphonie du futur? Telles sont les questions auxquelles ce livre tente d'apporter une réponse.

Cette seconde édition s'enrichit de nombreux compléments et exemples sur les dernières évolutions des technologies et des solutions logicielles et inclut, en particulier, un nouveau chapitre dédié à l'architecture IMS (IP Multimedia Subsystem).

### Au sommaire

**La théorie.** Problématique de la ToIP • Les contraintes de la VoIP • La signalisation H.323 • Le protocole SIP • Le protocole MGCP • La qualité de service • Architectures et sécurité (ToIP sur Ethernet, ATM, relais de trames, Wi-Fi, WiMax...). **Les solutions pratiques.** La ToIP sur softphones • Skype • Windows Live Messenger et Yahoo! Messenger • Jabber et Google Talk • Asterisk : une solution logicielle de PBX, libre et ouverte • Offre des fournisseurs d'accès (ToIP sur xDSL, CATV...) • Filtrage des flux de ToIP (NAT et pare-feu). **Perspectives.** Les cinq problèmes clés de la ToIP • L'architecture IMS.

Code éditeur : G12359  
ISBN : 978-2-212-12359-3



9 782212 123593

Conception : Nord Compo  
© Fotolia

[www.editions-eyrolles.com](http://www.editions-eyrolles.com)

Groupe Eyrolles | Diffusion Geodif | Distribution Sodis

**EYROLLES**