

Etude des spécifications de protocoles de signalisation relatifs à la téléphonie sur Internet

CHAPITRE 1 PRÉSENTATION GÉNÉRALE	5
1.1 EVOLUTION DES TÉLÉCOMMUNICATIONS	5
1.2 TELECOMMUNICATIONS ET SIGNALISATION	6
1.3 OBJECTIF	7
CHAPITRE 2 INFORMATIONS SUR LES TÉLÉCOMMUNICATIONS.....	8
2.1 GÉNÉRALITÉS DES TÉLÉCOMMUNICATIONS.....	8
2.1.1 Définition	8
2.1.2 Signaux analogiques et numériques	8
2.1.3 Réseaux à commutation.....	9
2.1.4 Le terminal	9
2.1.5 Organismes de normalisation	10
2.2 TRANSMISSION DE LA VOIX	11
2.2.1 Capacité de transmission	11
2.2.2 Pulse Code Modulation.....	11
2.3 LA SIGNALISATION.....	12
2.3.1 Définition	12
2.3.2 Signalisation SS7.....	13
CHAPITRE 3 VOICE OVER IP	14
3.1 VOIP: UNE TECHNOLOGIE EFFICACE ?	14
3.2 LES DIVERSES CONFIGURATIONS DE VOIP	15
3.3 CARACTÉRISTIQUES DES APPLICATIONS	17
3.3.1 Outil de traitement de la voix.....	18
3.3.2 Outil de traitement des appels.....	18
3.4 PREMIÈRES QUESTIONS	19
3.5 AUTRES APPLICATIONS	19
CHAPITRE 4 SIP-H.323	20
4.1 LES STANDARDS SUPPORTANT VOIP	20
4.2 LE STANDARD H.323.....	20
4.2.1: Modules et définition de H.323.....	20
4.2.2 Le Terminal.....	23
4.2.3 Le Gateway	24
4.2.4 Le gatekeeper	25
4.2.5 Fonctions de signalisations RAS.....	26
4.2.6 Call signaling functions	26
4.2.7 Diagramme d'une connexion H.323.....	27
4.2.8 Principe de fonctionnement (sans gatekeeper)	28
4.2.9 Principe de fonctionnement avec gatekeeper.....	32
4.3 SESSION INITIATION PROTOCOL (SIP).....	35
4.3.1 Définition de SIP.....	35
4.3.2 Architecture SIP	36
4.3.3 Transaction SIP.....	37
4.3.4 Schéma d'établissement de connexion SIP.....	39
4.4 COMPARAISON SIP ET H.323	41
CHAPITRE 5 H.323 À LA TRACE...	43
5.1 VISUALISATION DE LA SITUATION	43
5.2 MESSAGES RAS: REQUÊTES RRQ ET RCF	44
5.3 MESSAGES RAS: REQUÊTES ARQ ET ARJ	47
CHAPITRE 6 TTT-SERVICES.....	48
6.1 DESCRIPTION	48
6.2 PORTABILITÉ NATIONALE ET INTERNATIONALE	48
6.3 CARACTÉRISTIQUES D'IMPLÉMENTATION	49
6.4 SCÉNARIOS D'ENREGISTREMENT.....	50
6.4.1 Concepts.....	50

6.4.2 Scénarios à partir d'un terminal SCN vers son GW.....	51
CHAPITRE 7 IMPLÉMENTATION.....	54
7.1 PRÉ-REQUIS D'IMPLÉMENTATION.....	54
7.1.1 Les objets ASN.1.....	54
7.1.2 Fonctionnement interne d'OpenGatekeeper.....	56
7.1.3 Structures du logiciel OpenGatekeeper	56
7.2 REQUÊTES D'ENREGISTREMENT H.225	58
7.2.1 Rappel du principe	58
7.2.2 Messages H.225	59
7.3 OBJECTIF ET MÉTHODE D'IMPLÉMENTATION	63
7.3.1 Objectif.....	63
7.3.2 Méthode d'implémentation	63
7.3.3 Serveur RAS et connexion à la DB TRS.....	64
7.3.4 ResolutionService API.....	67
7.4 VISUALISATION	68
CONCLUSION.....	69
GLOSSAIRE.....	70
ANNEXES.....	71
A.1 LES PROTOCOLES RTP ET RTCP	71
A.1.1 Généralités.....	71
A.1.2 Quelques définitions	72
A.1.3 Structure du paquet RTP	73
A.2 SYNTAXE ASN.1	75
A.3 SIGNALISATION SS7.....	78
A.3.1 Les points de signalisation.....	78
A.3.2 Pile du protocole SS7.....	79
A.4 CALCUL DE DIMENSIONNEMENT	80
A.5 DEPENDANCES	82
BIBLIOGRAPHIES.....	85

Serons-nous capables de choisir les éléments de la technologie qui améliorent la qualité de vie et d'éviter ceux qui la détériorent ?

*David Baltimore
Biologiste américain
Nobel de médecine 1975*

Chapitre 1

Présentation Générale

1.1 EVOLUTION DES TÉLÉCOMMUNICATIONS

Depuis l'invention du téléphone par Alexander Graham Bell en 1876, de nombreux progrès et révolutions se sont opérés dans le domaine des télécommunications. Aujourd'hui, d'ailleurs, nous vivons dans l'ère des télécommunications et il est devenu impensable de se séparer de cette merveilleuse technologie qu'est la téléphonie.

Avec le développement des nouvelles technologies informatiques ces dernières décennies, le monde des télécommunications est entré dans une phase d'effervescence qu'auparavant nous n'avions jamais connue. Il suffit de regarder autour de nous, le nombre de nouvelles entreprises qui ont pris naissance dans ce domaine pour nous rendre compte de la "success Story".

Malheureusement, les parts du gâteau deviennent relativement de plus en plus rares à se partager. On remarque que ce sont les entreprises les plus compétitives qui rachètent les plus petites entreprises téméraires. Ces grandes entreprises profitent alors d'un certain monopole mais malgré tout offrent un service de qualité et la plupart du temps à des prix compétitifs.

Quelques fois, et surtout ces derniers temps, il émerge des idées formidables d'un point de vue technologique. C'est le cas avec l'apparition des technologies GSM, UMTS, WAP... Ces technologies sont prometteuses d'un bel avenir, pour la plupart, et conduisent à la création de services de plus en plus personnalisés. On en arrive à dire que la société de communication trace un chemin pour une société de services à outrance.

Parmi les nouvelles communications, il en existe une qui s'est laissée progressivement découvrir au grand public durant la fin du siècle dernier. Il s'agit bien entendu d'Internet. Aujourd'hui le monde entier est relié au moyen de routes virtuelles qui constituent la toile mondiale. C'est par Internet que d'abord l'armée américaine et ensuite les universités s'échangeaient des données. Aujourd'hui Internet est bien présent parmi une grande majorité de foyers. On évalue à environ plusieurs centaines de millions le nombre de postes actuellement connectés à Internet. Nous sommes arrivés à une société de l'information.

Si autour de nous, nous entendons parler de télécommunication, il faut remarquer que ces télécommunications coûtent cher. Une idée géniale est apparue au milieu des années nonante, justement pour pallier les problèmes des coûts des communications téléphoniques nationales et internationales. Elle s'est basée sur l'observation qu'un grand nombre de personnes sont

connectées à Internet. Or Internet, on le sait, ne connaît pas les frontières, et qu'on aille butiner sur un site étranger ou national, notre tarification Internet reste inchangée; elle reste une tarification zonale.

Cette technologie consiste à faire basculer une partie du trafic issu des lignes téléphoniques conventionnelles sur le réseau Internet et d'éviter ainsi des tarifications parfois élevées. Dans le jargon elle est appelée Voice Over IP.

Commençons par nous éclairer sur ce en quoi consiste Voice over IP. Plusieurs définitions ont été établies et apportent chacune des précisions sur les modalités de la transmission de l'information. En voici une qui reprend l'ensemble des concepts à retenir:

Voice Over IP est une technologie qui permet des communications audio temps réel entre deux ou plusieurs points d'un réseau qui peut être hétérogène. Le réseau hétérogène est constitué de réseaux basés sur le principe de la transmission par paquet et supportant le protocole IP et de réseaux basés sur le principe de circuits commutés. Le principe de transfert de l'information sur le réseau IP est basé sur la méthode de transmission avec le meilleur effort qui consiste à envoyer l'information sur le support de transmission quand celui-ci est disponible et sans qu'il y ait de gestion de priorité de transmission.

1.2 TELECOMMUNICATIONS ET SIGNALISATION

La signalisation est une des plus importantes fonctions dans l'infrastructure des télécommunications puisqu'elle permet aux composants du réseau de communiquer entre eux pour établir et terminer des appels. Voice Over IP, dont le but est d'établir des canaux de communication vocaux entre utilisateurs, requiert l'utilisation de protocoles de signalisation pour établir et terminer les appels.

Au début des années nonante, il n'existait pas, à proprement parler, de protocoles standardisés permettant la signalisation entre logiciels intégrant Voice Over IP. Dès lors, il était impossible de communiquer entre participants si ceux-ci ne possédaient pas rigoureusement le même logiciel. En effet, chaque développeur construisait son logiciel d'après ses propres protocoles de signalisation. Pour pallier à ce problème, l'ITU (Union Internationale des Télécommunications) entreprit l'idée de standardiser la signalisation sur Voice Over IP. C'est en 1996 que le groupe de travail numéro 16 de l'ITU proposa le standard H.323 version 1. Il s'agissait du premier standard de signalisation concernant la transmission d'informations multimédia en temps réel sur des réseaux ne possédant pas de qualité de service.

Ensuite vint l'engouement pour l'interconnexion du monde IP et des réseaux à commutation de circuits (SCN). Il fallait développer des outils matériels qui puissent effectuer des conversions de médias et dont la tâche la plus importante consistait à comprendre les messages de signalisation envoyés de part et d'autre des réseaux hétérogènes. C'est la tâche des "gateways". Ces nouveaux développements rendirent le standard H.323 limité car celui-ci n'intégrait pas la signalisation SS7, propre aux réseaux SCN.

A nouveau, il fallut attendre 1998 pour que le standard MGCP (Media Gateway Control Protocol) ou "Megaco" soit établi par l'ITU groupe 16. Ce standard régit l'utilisation du Media Gateway et du Media Gateway Controller dont les buts sont de, respectivement, convertir les médias et contrôler les appels.

Dans d'autres perspectives d'évolution est également né SIP dans la fin des années nonante. Plus simple qu'H.323, il attire aujourd'hui les regards de développeurs.

1.3 OBJECTIF

L'objectif de ce travail de fin d'étude est d'une part de comprendre les différences entre les grands protocoles de signalisation existants, et qui traitent de la téléphonie sur IP, et d'autre part d'aborder un sujet assez vaste et qui concerne la mobilité universelle.

Ce travail est donc composé de deux grands volets. Le premier explique les caractéristiques fondamentales des protocoles de signalisation H.323 et de la recommandation SIP. Il tente également d'apporter une petite comparaison entre ces deux protocoles. Le deuxième volet explique les concepts de mobilité universelle et comment les appliquer à la téléphonie sur Internet. Dans ce volet on retrouvera une partie concernant l'intégration d'un service de mobilité universelle dans un logiciel Open Source. Ce logiciel est nécessaire dans certains cas de figure d'implémentation du protocole H.323.

Chapitre 2

Informations sur les télécommunications

2.1 GÉNÉRALITÉS DES TÉLÉCOMMUNICATIONS

2.1.1 Définition

La télécommunication implique la communication entre usagers d'un service (téléphonie, télécopie...) échangeant des informations via un terminal (téléphone à touche, fax). Cette communication est établie via une ressource de transmission qui les relie et supporte le transfert d'informations.

Un réseau de communication permet d'établir une liaison à la demande d'un usager connecté au réseau vers un autre. Deux types de réseau sont rencontrés: les réseaux publics et les réseaux privés. Les réseaux publics fournissent des services et des accès accessibles à tous. Quant aux réseaux privés, ils substituent des éléments du réseau privé à des éléments publics. L'usage de ce réseau est limité à des usagers internes à l'organisation qui possède le réseau privé. Malgré tout, un réseau privé peut utiliser une partie de l'infrastructure du réseau public.

2.1.2 Signaux analogiques et numériques

Une information à émettre est toujours représentée sous forme d'un signal qui peut être numérique ou analogique. Fondamentalement, un signal est toujours une grandeur physique analogique. La différence consiste dans le type d'information que celui-ci transporte. Un signal analogique prend des valeurs continues et varie en amplitude fréquence et phase. Un signal numérique prend une série de valeurs discrètes représentées par un signal élémentaire appelé moment. Les signaux analogiques peuvent être convertis en signaux numériques et inversement par des équipements appelés CODEC (codeur/décodeur) ou modems (modulateur/démodulateur).

2.1.3 Réseaux à commutation

Dans ces réseaux, l'information est transmise depuis le terminal de départ jusqu'au terminal d'arrivée au travers d'une série de nœuds connectés entre eux par des liaisons de transmission. Chaque nœud agit comme un commutateur et aiguille le chemin sur base de l'adresse de destination. Les réseaux de commutation sont subdivisés en deux catégories:

- Les réseaux à commutation de circuit
- Les réseaux à commutation de paquets

La commutation de circuits établit une connexion dédiée entre l'émetteur et le récepteur situés sur le réseau de données. Les données circulent de la source à la destination le long du circuit établi pour cette session particulière. Une fois le transfert de données terminé, la connexion entre l'émetteur et le récepteur prend fin et le circuit disparaît.

Dans la commutation par paquets, l'information à transmettre est envoyée sous forme de messages de taille limitée appelés paquet. Chaque paquet est transmis de nœud en nœud suivant la technique "store and forward" en utilisant la capacité de transmission disponible entre les deux nœuds. Lorsqu'un paquet est reçu erroné, celui-ci peut être retransmis. La capacité de transmission n'est donc pas assignée à une communication bien précise mais à plusieurs communications simultanées. La petite taille de paquet utilisée sur les réseaux étendus à commutation par paquets permet d'acheminer les données rapidement et efficacement. Chaque paquet possède ses propres informations de contrôle et il est commuté indépendamment des autres au sein du réseau. Chaque paquet peut donc suivre un chemin différent des autres même s'ils ont la même destination.

Les réseaux à commutation de paquets peuvent tirer profit de circuits virtuels pour transférer les données. Un circuit virtuel établit un itinéraire défini au sein du réseau afin que tous les paquets de données empruntent le même chemin pour arriver à destination. Ce chemin pourra éventuellement être emprunté par d'autres utilisateurs, puisque les réseaux commutés emploient des lignes partagées. L'utilisation de circuits virtuels sur un réseau à commutation de paquets peut accroître les performances générales des transferts de données.

2.1.4 Le terminal

Le terminal est un élément bidirectionnel du réseau qui reçoit et envoie un trafic multimédia sur le réseau auquel il est connecté. Si le terminal est branché à un réseau IP il s'agit alors soit d'un PC multimédia, soit d'un appareil multimédia autonome. Si le terminal est connecté à un SCN alors il s'agit d'un téléphone.

2.1.5 Organismes de normalisation

2.1.5.1 Organismes internationaux

IETF

L' "Internet Engineering Task Force (IETF)" est l'organe de développement et de recherche sur les protocoles Internet. Il s'agit d'un groupe constitué par une communauté internationale d'architectes réseau, de vendeurs, de chercheurs concernés par toute évolution dans le domaine Internet. La plupart des travaux de l'IETF sont organisés dans des groupes de travail et traitent de sujets comme la sécurité, le routage, le transport ...

ITU

ITU est l'abréviation de "International Telecommunication Union". Il s'agit d'une organisation internationale dans laquelle les gouvernements et le secteur privé coordinent des services et des réseaux de télécommunication globaux.

2.1.5.2 Organismes européens

ETSI

ETSI est l'abréviation de "the European Telecommunications Standards Institute". Il s'agit d'une organisation sans but lucratif dont la mission est de produire des standards de télécommunication, à grande longévité, qui seront utilisés en Europe et ailleurs.

Toute organisation éprouvant des intérêts à promouvoir des standards de télécommunications européens a le droit de représenter ces intérêts à l'ETSI et donc d'influencer directement le processus d'élaboration des standards.

Ce sont les membres de l'ETSI qui décident des travaux à apporter sur les standards en fonction des demandes du marché.

Tiphon

En se rendant compte du besoin de solutions communes pour la standardisation de la voix sur Internet, l'ETSI a créé TIPHON. TIPHON est l'abréviation de " Telecommunications and Internet Protocol Harmonization Over Networks".

L'objectif de TIPHON est de supporter le marché des communications vocales, et autres communications du même registre (Fax..) , entre les utilisateurs. Le projet doit s'assurer que les utilisateurs connectés à un réseau IP peuvent communiquer avec des utilisateurs situés sur un réseau de télécommunication SCN (PSTN, ISDN, GSM) et inversement.

Le problème de la communication entre usagers connectés à un réseau de circuits commutés mais transitant aussi par Internet, est également abordé.

TIPHON délivre des documents contenant des spécifications techniques et des rapports. Les solutions proposées sont évaluées et confirmées par le biais de démonstrations.

Puisque TIPHON appartient à l'ETSI, qui est un organisme européen, et que travailler sur le protocole IP est international, il est reconnu que des relations de coopération existent entre les organismes ITU et IETF.

2.1.5.3 Organisme américain (ANSI)

ANSI (American National Standards Institute) est une organisation privée, sans but lucratif, qui administre et coordonne les standards américains.

Le but de l'organisation est d'améliorer la compétitivité du commerce américain et la qualité de vie. Ceci en promouvant et facilitant les standards à consensus volontaires et les systèmes à conformité évaluée, en sauvegardant leur intégrité.

2.2 TRANSMISSION DE LA VOIX

2.2.1 Capacité de transmission

La voix est un signal analogique de 3,1 kHz de bande passante (300 à 3,4 kHz). Ce signal est souvent numérisé et donc converti en signal numérique. D'après Shannon, ce signal analogique peut être converti en un signal numérique sans perte d'information pour autant que sa fréquence d'échantillonnage soit le double de sa fréquence maximale. Dans notre cas notre signal vocal a une fréquence maximale de 4kHz, donc il nous faut une fréquence d'échantillonnage de 8kHz. Si chacun de ces échantillons est codé au moyen d'un code à 8 bits, on obtient une capacité de 64kbits/s. On dit donc que la capacité du réseau téléphonique public (PSTN) est de 3,4 kHz ou 64kbits/s.

2.2.2 Pulse Code Modulation

Il s'agit de la méthode d'encoder un signal analogue en un train de bits. Tout d'abord, l'amplitude du signal analogique est échantillonnée. Cette opération s'appelle PAM (Pulse Amplitude Modulation); cet échantillon PAM est ensuite encodé en nombres binaires. Ce signal peut donc être commuté et transmis de manière digitale.

Il y a trois avantages à moduler un signal analogique par la méthode PCM; Premièrement c'est moins cher de commuter et de transmettre un signal digital. Deuxièmement, en transformant un signal analogue en digital, on

peut l'intercaler entre d'autres signaux digitaux tels les signaux de données émis par les ordinateurs. Et enfin, troisièmement, un signal vocal qui est transmis et commuté entre deux "End-Points" de façon digitale sera généralement plus claire car possèdera moins de bruit qu'un signal transmis de façon analogue.

PCM est une technique de digitalisation, il ne s'agit pas d'un standard accepté universellement. Nous nous plaçons à présent dans le cas où le signal analogique à coder est un signal vocal. La méthode PCM la plus conventionnelle est celle où on échantillonne la voix à une fréquence de 8000Hz. Le théorème de Shannon rappelle qu'il faut échantillonner un signal à au moins deux fois sa fréquence maximale. Comme la fréquence la plus élevée pour un signal audio transmis par le téléphone conventionnel est de 4000 Hz il faut échantillonner à 8000 Hz.

Plusieurs conversations digitales PCM sont placées typiquement sur un canal de transmission. En Europe nous utilisons un système de multiplexage des conversations téléphoniques sur 32 canaux. Sur ces 32 canaux, 2 sont requis pour le contrôle de la transmission et les 30 autres pour les conversations vocales. Ces 30 canaux ont un débit de 2 Mbit/s. Le système européen est calculé en $8\text{bits} \times 32 \text{ canaux} \times 8000 \text{ frames par seconde}$. En Amérique du Nord le canal type est appelé T1. Il place 24 conversations vocales sur deux paires de cuivre (Transmission et réception). Il contient 8000 frames de chacune 8 bits de 24 canaux "voix" et un bit de synchronisation, ce qui donne un débit de 1,54 Mbits soit $8000 \times (8 \times 24 + 1)$. Les systèmes européen et américain ne peuvent être connectés directement. Aujourd'hui, les Américains et les Européens ont adopté un standard commun qui est l'ISDN.

2.3 LA SIGNALISATION

2.3.1 Définition

La signalisation concerne l'échange d'informations entre les nœuds d'un réseau. Ces informations servent à l'établissement et au contrôle des connexions à travers le réseau. Il existe trois types de signalisation:

- La signalisation entre l'utilisateur et le réseau (User Network Signaling)
- La signalisation entre les nœuds du réseau (Inter Office Signaling)
- La signalisation entre les utilisateurs du réseau (End to End)

L'utilité de la signalisation repose bien sûr sur le principe de l'établissement des connexions, mais elle permet le transfert d'informations concernant la gestion du réseau et de ses ressources, la taxation, ...

Il existe un réseau particulier qui peut être attribué entièrement pour effectuer la signalisation. Ce genre de réseau se retrouve notamment en combinaison avec les technologies de l'ISDN et du PSTN. Il s'agit du réseau SS7 (ou CCS7 ou CCITT n°7).

2.3.2 Signalisation SS7

SS7 ou "Common Channel Signaling System" est un standard global de télécommunication défini par l'ITU. Ce standard définit les procédures et les protocoles par lesquels les éléments du réseau, dans un réseau à commutations de circuits, s'échangent des informations de contrôle et de routage sur un réseau digital de signalisation.

SS7 est utilisé pour:

- L'établissement et la gestion des appels
- Les services mobiles comme le roaming
- La portabilité des numéros
- Les services intelligents (0800, 0900..)
- Les services de transfert (call forwarding)

La signalisation SS7 s'effectue par transmission de messages, entre les éléments du réseau, à une vitesse de 56 à 64 Kbps sur un canal bi-directionnel appelé "signaling link". La signalisation se transmet en dehors de la bande de transmission réservée au transport proprement dit des données. C'est de la signalisation "Out Of Band".

Cette signalisation offre, en comparaison avec la signalisation "In band", de meilleurs temps de réponse pour l'établissement des appels (cfr: Multi Frequency Signaling Tones), une meilleure utilisation des canaux voix et un support pour les réseaux intelligents.

Voir en annexe 3 pour des explications plus complètes concernant SS7.

Chapitre 3

Voice Over IP

3.1 VOIP: UNE TECHNOLOGIE EFFICACE ?

Certaines personnes peuvent vite être déçues du résultat d'une première expérience avec la téléphonie sur Internet: De nombreux problèmes de lenteur de connexion, de délai dans la transmission, d'effet d'écho peuvent être rencontrés. Mais alors quel est cet engouement pour cette technologie? Il y a plusieurs raisons principales:

La première provient des avancées des nouvelles technologies. On a constaté que la vue d'une image statique sur une page web fait moins d'effet qu'une image animée, de même qu'il est mieux d'ajouter un fond musical lors du chargement de celle-ci. On assure une continuité dans ces progrès, en permettant par exemple à l'utilisateur d'interagir directement avec sa voix sur ce qu'il perçoit lors du chargement d'une page web.

La deuxième est évidemment d'ordre économique. La téléphonie sur IP évite les tarifications onéreuses des communications longues distance puisque Internet est "gratuit".

La troisième est la présence universelle du protocole IP dans les installations réseau.

A fin de montrer ce que pensent les cabinets de consultance a propos de la technologie Voice over IP, voici un schéma, dont la source provient du "Eastern Management Group", qui est très représentatif de l'évolution qu'ils en espèrent. Bien entendu ce schéma n'est le résultat que d'une série de suppositions et ne constitue en rien une vérité.

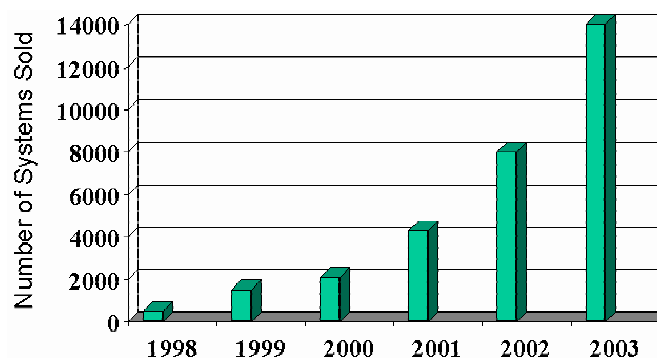


Figure 3.1
Perspective d'évolution des ventes de Voice Over IP

J'ai pour ma part testé gratuitement la téléphonie sur IP grâce au service offert par le site web <http://www.hottelephone.com>. Malheureusement les meilleurs plaisirs ayant une fin, par manque de moyens, la gratuité du service s'est transformée en service payant. On constatait un écho certain sur les communications transmises, mais le service étant gratuit, on ne disposait probablement pas des meilleures configurations pour établir les appels. Malgré tout j'ai pu constater que la technologie offerte permettait d'établir des communications téléphoniques sans trop de problème depuis un poste doté d'une connexion Internet vers un poste du réseau PSTN (qu'il fût belge ou étranger).

3.2 LES DIVERSES CONFIGURATIONS DE VOIP

VoIP permet la transmission de la voix entre deux correspondants de trois manières différentes:

- Entre deux terminaux, chacun d'eux étant raccordé à un réseau IP
- Entre un terminal raccordé à un réseau IP et un terminal raccordé à un réseau de commutation par circuits
- Entre un terminal raccordé à un réseau IP et un terminal raccordé à un réseau de commutation par circuits
- Entre deux terminaux raccordés chacun à un réseau de commutation par circuits

La dernière configuration peut, à première vue, ne pas différer de la configuration que chacun d'entre nous connaît. Malgré le fait que les deux terminaux soient raccordés chacun à un SCN, il se peut que le trafic généré soit dévié vers un réseau IP intermédiaire aux deux SCN.

Ces trois cas de figure sont représentés sur les schémas suivants.

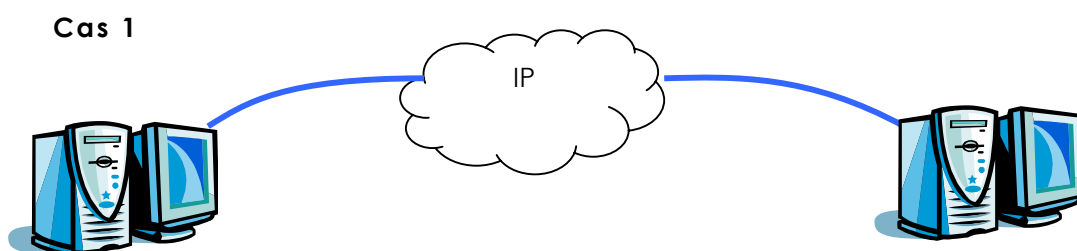


Figure 3.2
cas 1

Le cas représenté par la figure 3.2 est le plus simple car il n'y a pas de connexion avec un SCN. Il suffit, si le réseau est du type Internet, de connaître les adresses IP de chacun des terminaux pour établir une communication.

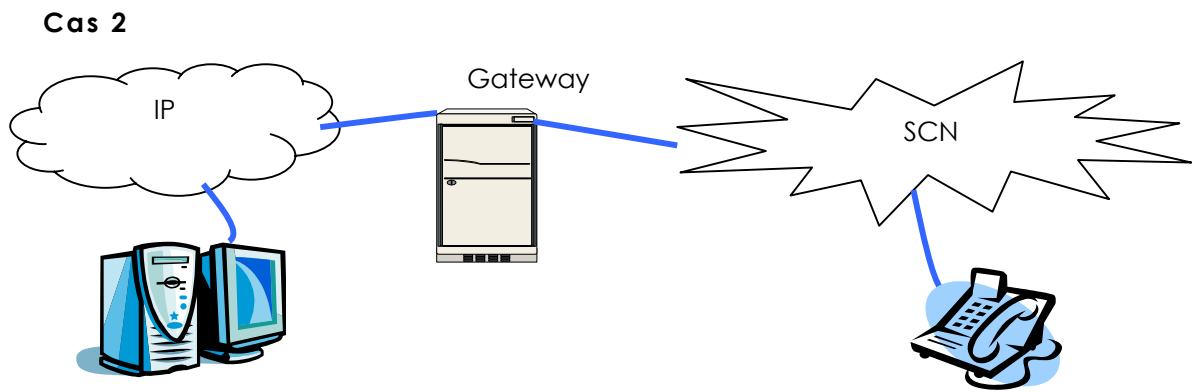


Figure 3.3
Cas 2

Le cas de la figure 3.3 est intermédiaire au cas 1 car il nécessite une conversion des signaux entre le réseau IP et le SCN. Le convertisseur utilisé est, on le verra, un Gateway.

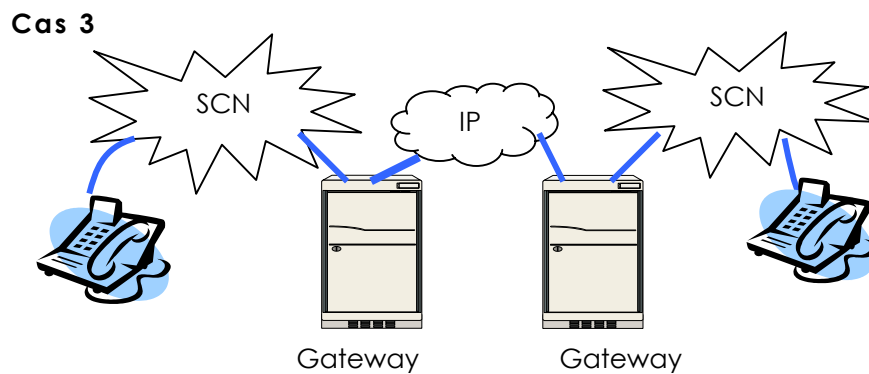


Figure 3.4
Cas 3

Le cas de la figure 3.4 est le plus compliqué car nécessite deux conversions de signaux.

3.3 CARACTÉRISTIQUES DES APPLICATIONS

Nous pouvons résumer, à présent, quelques caractéristiques qu'une application "VoIP" doit intégrer. On a effectivement besoin, lors de la programmation de ce genre d'application, d'être doté de compétences à divers niveaux. C'est la raison pour laquelle il faut disposer de beaucoup d'outils logiciels pour établir une communication entre un abonné du réseau PSTN et un abonné du réseau IP. Ces outils permettent notamment l'interaction entre l'application et l'interface réseau.

On peut classer ces outils en trois types:

- Les outils de traitement de la voix
- Les outils de traitement des appels
- Les outils de traitement des paquets
- Les outils de gestion du réseau

Les outils de traitement de la voix sont ceux qui préparent des échantillonnages de son à transmettre sur le réseau de paquets. Ces logiciels sont typiquement des DSP (Digital Signal Processing).

Les outils de traitement des appels sont ceux qui assurent les connexions avec les gateways qui permettent à l'échantillonnage de son de passer sur le réseau de paquets.

Les outils de traitement des paquets sont ceux qui traitent les paquets de voix et de signalisation en leur ajoutant un en-tête correct avant de les transmettre sur le réseau de paquets IP.

Les outils de gestion du réseau permettent la gestion des fautes, de tenir une comptabilité d'utilisation ou de configurer d'autres services. Ils sont parfois dotés de modules assurant une sécurité ou fournissant des statistiques d'utilisation...

Nous pouvons suggérer un ensemble de caractéristiques nécessaires au bon fonctionnement de ces outils.

3.3.1 Outil de traitement de la voix

Pour l'outil de traitement de la voix, il faut que celui-ci possède:

- Une interface PCM (Pulse Code Modulation) qui reçoit les signaux vocaux du PSTN et les renvoie vers le module approprié pour être traité en ré-échantillonnant la voix vers l'interface analogue.
- Une unité d'élimination d'écho. Cette opération est nécessaire sur une structure VoIP car sinon sa durée est intolérable (min 50 ms).
- Un détecteur d'(in)-activité, qui supprime la transmission de paquets s'il n'y a pas de signal audio à transmettre durant un certain temps programmable.
- Un détecteur de tonalité "clavier" qui les différencie des signaux vocaux.
- Un protocole de transmission de la voix sur le réseau de paquets. Cette unité encapsule les paquets à transmettre et leur assigne un numéro de séquence pour qu'ils puissent être réceptionnés dans le bon ordre.
- Un module de "playback audio" qui après avoir reçu suffisamment d'informations de l'émetteur dans son buffer les émet sur un haut-parleur.

3.3.2 Outil de traitement des appels

En ce qui concerne l'outil de traitement des appels, il faut que celui-ci permette la détection de la présence d'un nouvel appel et réceptionne les informations d'adressage de ces appels. Plusieurs standards de signalisation téléphonique doivent également être supportés. S'il s'agit de communications avec un téléphone du réseau téléphonique conventionnel, il faut que ce module réponde à certaines spécifications:

- L'interface du réseau de téléphone doit être sous monitorat (réception des commandes et des réponses)
- Les protocoles de signalisation doivent être terminés et l'information doit être extraite
- L'information de signalisation doit être convertie dans un format qui puisse être utilisé pour établir une session au travers du réseau de paquets
- Les numéros de téléphone conventionnels (E.164) doivent être convertis en adresse IP éventuellement à l'aide d'un service d'annuaire.

Il n'est pas besoin de rappeler que le logiciel utilisé dans les périphériques VoIP doivent supporter les protocoles "temps réel".

3.4 PREMIÈRES QUESTIONS

D'après ce qu'il vient d'être écrit, il peut nous venir à l'esprit quelques questions auxquelles je donnerai ici les réponses. Tout d'abord pourquoi la nécessité d'emprunter un chemin différent de celui des SCN ? La réponse est d'ordre économique et financier: L'Internet gratuit est la réalité d'aujourd'hui, et la plupart des entreprises y sont reliées de même que de nombreux particuliers. Alors que pour joindre, depuis chez soi, un site Web sur un autre continent nous ne payons pas plus cher qu'une communication téléphonique zonale, pour téléphoner à l'étranger un tarif beaucoup plus élevé nous est demandé. La solution adéquate serait donc d'inventer un système qui puisse se servir de l'Internet pour téléphoner à son correspondant. Cette solution réside dans l'emploi de VoIP. Dans ce cas, seule la facture de la communication zonale nous est prélevée. Nous parvenons ainsi à engranger de sérieuses économies. De plus parfois il est plus évident de relier des régions retirées à Internet qu'à un réseau commuté. (une ligne rapide à placer plutôt que de nombreuses paires de cuivre)

Mais la valeur ajoutée de Voice Over IP n'est pas qu'économique. L'unification de données et de la voix reste un atout. A ces avantages, il faut malheureusement ajouter des inconvénients. Le passage d'une utilisation standard de la téléphonie à une téléphonie sur IP oblige une modification de nombreux équipements déjà présents dans une entreprise, comme les routeurs. On n'est d'ailleurs pas près de voir le PSTN remplacé.

Ensuite, quel sera l'avenir des sociétés de téléphonie qui fournissent une partie du support de VoIP ? Dans une récente étude il est montré que ces sociétés planifient une chute de leur chiffre d'affaires d'environ 10%. Malgré tout, elles ne sont pas en reste, car si le nombre total de communications téléphoniques "réelles" chute, au contraire, l'emploi de l'Internet ne fera, au contraire, que croître. Et ainsi, une partie de la boucle est bouclée. Ce que ces sociétés perdront en communications inter zonales ou internationales, elles le gagneront en communications zonales. Ces sociétés n'ont donc pas grand chose à craindre de cette nouvelle technologie. On estime également que Voice over IP représente un marché de 120 millions de dollars pour l'Europe (Source: Cabinet d'étude IDC).

3.5 AUTRES APPLICATIONS

De nouvelles applications naîtront certainement dans les prochains mois autour de VoIP, mais retenons déjà qu'une technologie similaire est utilisée pour transmettre des fax sur Internet (Fax Over IP), et que l'on n'est pas limité au simple transport de la voix.

Pour l'entreprise, intégrer les données et la voix sur un support de transmission commun offre une ouverture sur un potentiel élevé d'applications. Par exemple on peut imaginer des systèmes de téléformation de visioconférence...

Chapitre 4

SIP-H.323

4.1 LES STANDARDS SUPPORTANT VOIP

Jusqu'à présent, il existe trois standards ou protocoles qui permettent la mise en place d'un "service" VoIP. Le plus connu est le standard H.323 qui sera détaillé dans la section suivante. Ensuite, plus ancien, le MGCP (Media Gateway Control Protocol). Le plus récent est le SIP qui, également, sera détaillé dans une des sections suivantes. J'attire l'attention sur le fait que H.323 est plus difficile à mettre en œuvre par rapport à SIP et que sa description est plus complexe. H323 retient malgré tout toute mon attention car il s'agit du protocole qui est utilisé dans le logiciel qui est à ma disposition pour la suite de ce travail de fin d'étude.

4.2 LE STANDARD H.323

4.2.1: Modules et définition de H.323

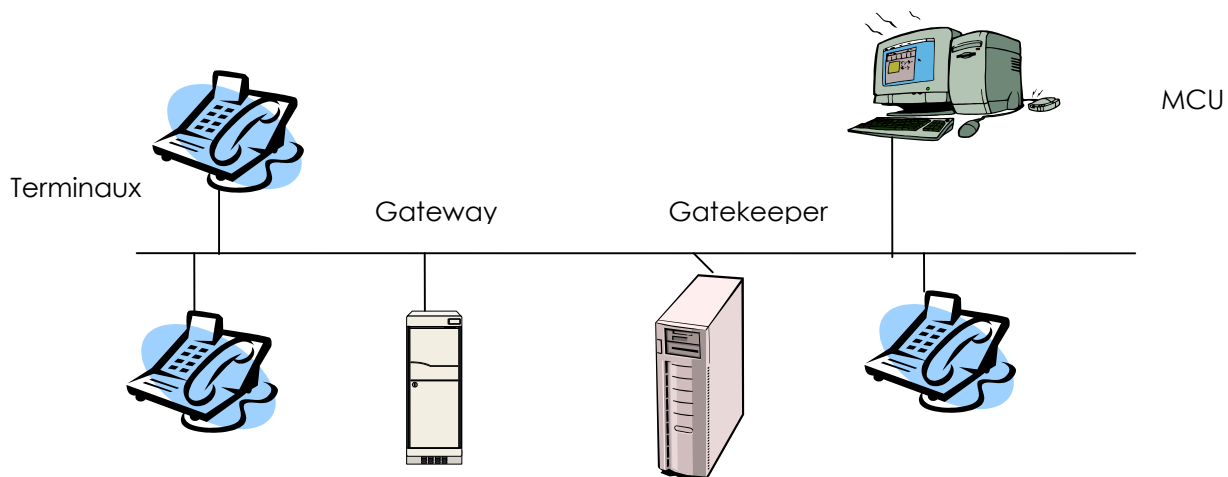


Figure 4.1
Réseau H.323

Le standard H.323 a été conçu par l'ITU-T. Il spécifie les composants, protocoles et procédures permettant la mise en place d'un service multimédia sur un réseau à transmission par paquets (LAN, MAN...). H.323 fait partie d'une série de recommandations qui toutes décrivent des transmissions multimédia mais sur des réseaux différents. H.323 transmet des informations multimédia sur des réseaux à paquets commutés sans garantie de bande passante. Ce standard est valable pour VoIP car il permet de transmettre uniquement la voix ou un mélange de voix et de données. Il est constitué par un ensemble de protocoles permettant des communications entre plusieurs objets. Ces objets sont les Gateways, Gatekeeper et les terminaux. La figure 4.1 montre un réseau doté d'équipements basés sur le modèle H.323. Nous allons décrire le rôle de chacun de ces objets, mais avant tout voici dans la Table 4.1 la liste des protocoles que ce standard regroupe.

CODECS Audio
CODECS Video
RAS H.225 (Registration Admission Status)
H.225 (Call Signaling)
H.245 Control Signaling
RTP (Real Time Transfer Protocol)
RTCP (Real Time Control Protocol)

Table 4.1

Ensembles des protocoles contenus dans le modèle H.323

Le Terminal H.323

Le terminal H.323 est soit un téléphone, soit un personal computer muni d'une carte son et d'un micro, soit un appareil (Stand Alone) tournant sous le modèle du standard H.323 et exécutant des applications audio. Il s'agit d'un appareil "client" pour l'utilisateur. Eventuellement, le terminal peut être doté d'un système de transmission d'images et de données, mais ce n'est pas obligatoire. Cet appareil joue un rôle clef dans VoIP car c'est à partir de lui que seront émises et reçues les conversations des utilisateurs. Le rôle premier du standard H.323 est de permettre les échanges entre les terminaux.

Le Gateway

Le Gateway est l'appareil qui permet d'interconnecter deux réseaux dissemblables. Il s'agit d'un nœud sur le LAN. Il traduit et transmet le trafic d'un réseau H.323 vers un réseau non-H.323 et inversement. Par exemple il peut être connecté à un LAN et à un SCN du type PSTN (Public Switched Telephony Network). Cette traduction s'accomplit par les conversions de protocoles et de médias entre les deux réseaux nécessaires. Un gateway n'est pas nécessaire s'il s'agit de connecter uniquement des terminaux H.323. Sous certaines conditions, Le Gateway peut éviter le passage par un routeur connecté au réseau local LAN.

Le Gatekeeper

Le Gatekeeper est considéré comme le cerveau du réseau H.323. Il s'agit du point de focalisation pour tous les appels d'un réseau H.323. Bien qu'il ne soit pas nécessaire, le Gatekeeper est un objet commode du réseau H.323. C'est lui qui se charge d'autoriser les appels, d'authentifier les utilisateurs, d'établir une comptabilité, de contrôler la bande passante... Le Gatekeeper peut également fournir des services de routage. Un Gatekeeper administre un ensemble de terminaux, Gateways d'une certaine zone.

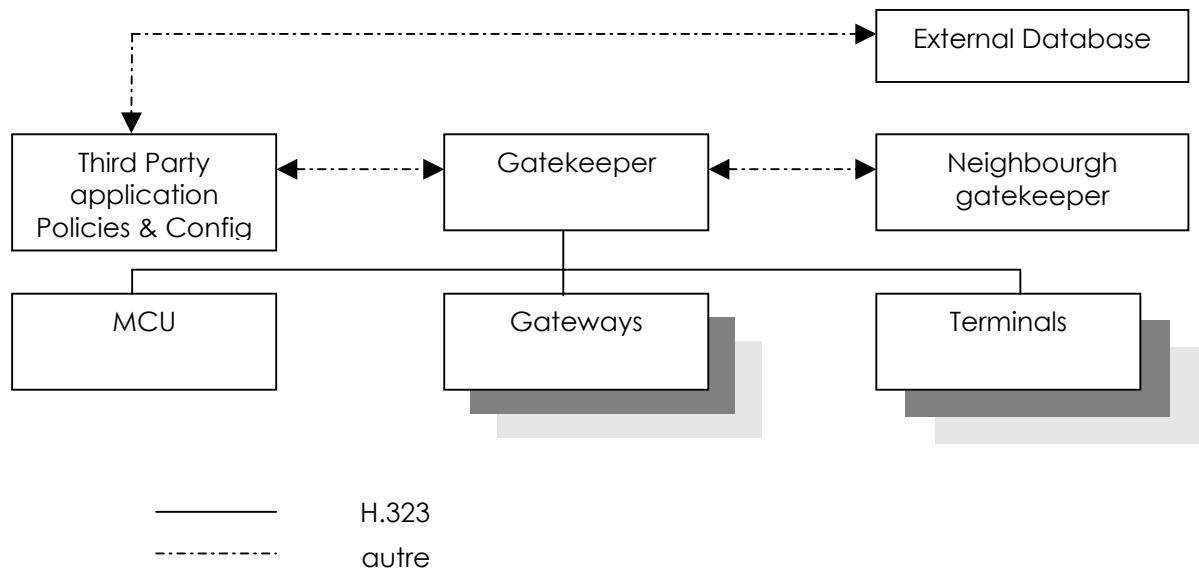


Figure 4.2

Aréancement logique des composants d'un réseau H.323

Le MCU (Multipoint Control Unit)

Le MCU fournit du support pour une conférence entre trois ou plusieurs terminaux. Chacun des terminaux désirant participer à la conversation doit s'enregistrer auprès du MCU. C'est le MCU qui négocie, entre les terminaux, les CODECS à employer durant la conférence. Il se charge également de signaler à chacun des terminaux s'il s'agit d'une audio-conférence ou d'une vidéo-conférence.

La Zone

La zone est un ensemble de terminaux, Gateways et MCU administrés par un Gatekeeper. Une zone inclut au minimum un terminal et peut inclure un gateway. Elle ne possède qu'un Gatekeeper. La zone peut englober plusieurs segments interconnectés par des routeurs.

La figure 4.2 représente l'agencement des composants d'un réseau H.323 et montre comment ils agissent l'un avec l'autre. Dans les points suivants nous allons décrire les caractéristiques de chacun de ces composants

4.2.2 Le Terminal

Le fonctionnement du terminal réside en ce qu'il peut envoyer et recevoir des messages multimédias. Il est ainsi doté d'une couche protocolaire d'application audio et vidéo. Cette couche représente l'interface de l'application vue par l'utilisateur sur le terminal. Elle repose sur un ensemble de CODECS audio et vidéo qui sont des standards de compression/décompression et d'encodage/décodage audio/vidéo. Un terminal doit obligatoirement avoir un CODEC audio codant à 64kbps.

Le transport des informations multimédia issues du terminal est effectué par l'intermédiaire du protocole RTP (Real Time Transport Protocol) et ensuite par la couche transport et l'interface réseau, (La couche de transport et l'interface réseau ne font pas partie du standard H.323).

Aux cotés des fonctionnalités évidentes du terminal, on retrouve un ensemble de protocoles qui serviront à initialiser et contrôler une session. Il s'agit des protocoles RTCP, H.225 RAS, H.225 Call Signaling, H.245 Control Signaling. Voici le schéma de la figure 4.3 qui représente les diverses couches protocolaires du terminal.

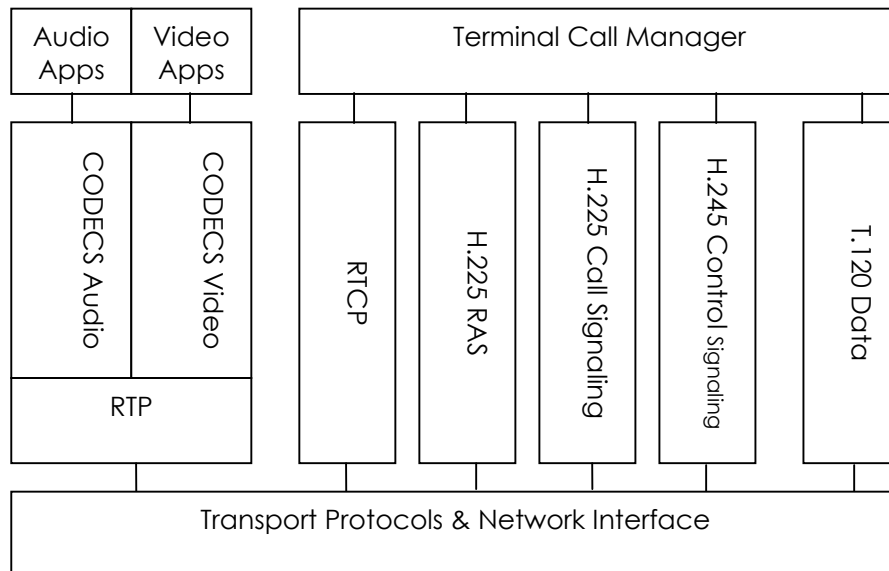


Figure 4.3

Structure en couche du terminal

4.2.3 Le Gateway

La structure du Gateway se compose de deux parties. La première est attachée au réseau de paquets, et la seconde au réseau public de commutation (téléphonique).

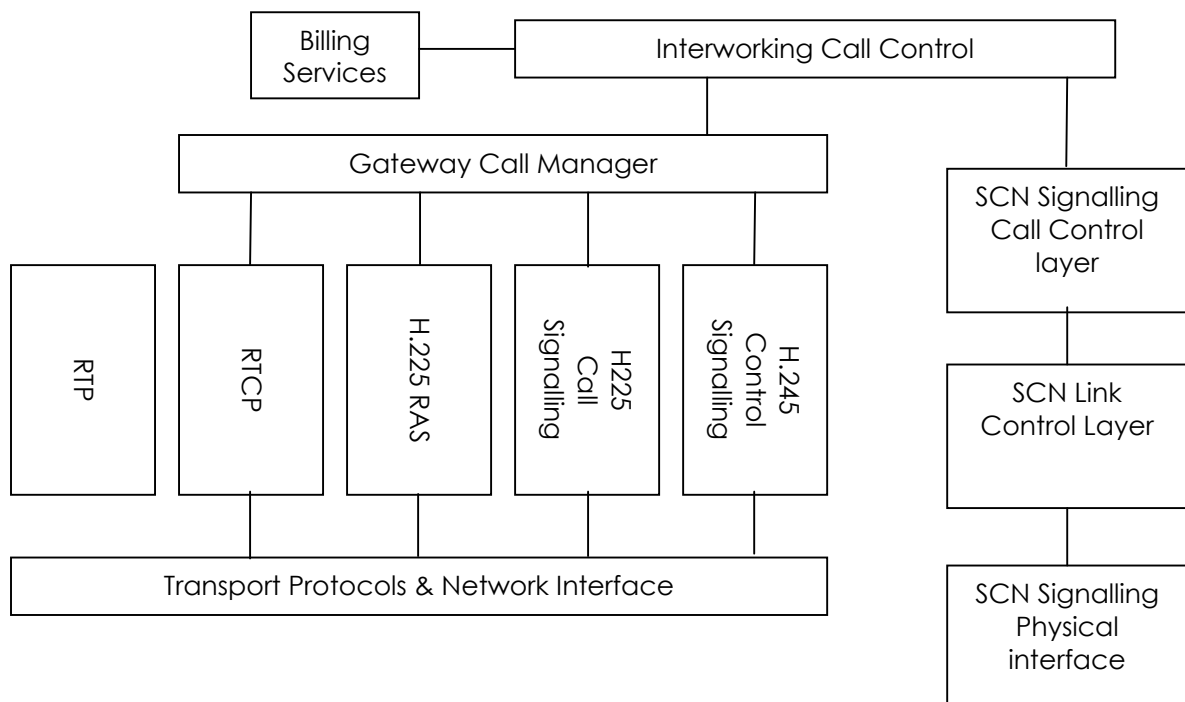


Figure 4.4

Structure en couche du gateway

Dans la partie "réseaux par paquets", on retrouve le contrôle de signalisation H.245 et l'H.225 dont une partie s'occupe du "call setup & release" et l'autre de RAS vers le gatekeeper. Les terminaux du côté "Réseau par Paquet" contactent le Gateway par l'intermédiaire de H.245 (Control Signaling) et H.225 (Call signaling).

Du côté SCN, Le Gateway fonctionne à l'aide des protocoles propres aux SCN tels SS7 et ISDN. Les terminaux du côté SCN le contactent également au moyen de la pile de protocoles spécifiques au SCN. Le gateway peut supporter plusieurs communications simultanées. La figure 4.4 montre les couches protocolaires par lesquelles est lié le gateway.

4.2.4 Le gatekeeper

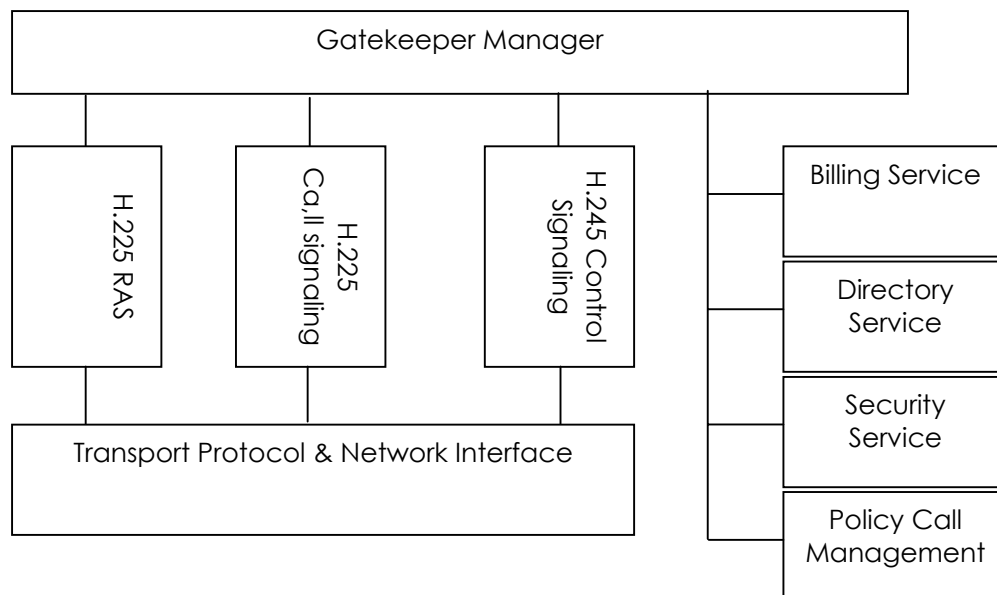


Figure 4.5
Structure en couches du gatekeeper

Le gatekeeper doit obligatoirement s'occuper d'effectuer des conversions d'adresse: Les appels originaires d'un réseau H.323 peuvent utiliser un alias pour adresser un autre terminal et de même, les appels originaires d'un réseau différent du H.323 et reçus par le gateway peuvent utiliser une adresse de type téléphonique (E.164) pour adresser un terminal. Le gatekeeper doit convertir cette adresse en une adresse IP.

En plus des conversions, une caractéristique importante du gatekeeper réside dans ce qu'il gère la fonctionnalité RAS en envoyant des messages de confirmation de requête aux clients qui le contactent.

Quand une entité du réseau H.323 entre en fonctionnement, elle envoie une requête sur le réseau pour s'informer si un gatekeeper est présent. Ce message est envoyé par broadcast.

La figure 4.5 montre les diverses couches protocolaires auxquelles est lié le gatekeeper.

4.2.5 Fonctions de signalisations RAS

La signalisation RAS utilise des messages H.225 pour établir l'inscription, l'admission, les changements de bande passante, le statut et pour les procédures de désactivation entre les points finaux "end-points" et les Gatekeepers. Le Canal de signalisation RAS est indépendant du canal de signalisation d'appel et du canal de contrôle H.245. Dans les environnements qui ne possèdent pas de Gatekeeper, le canal de signalisation RAS n'est pas utilisé. Si par contre le réseau contient un Gatekeeper (zone), un canal de signalisation RAS est ouvert entre le "enpoint" et le gatekeeper. Ce canal est ouvert par priorité à tous les autres canaux des "endpoints" H.323.

4.2.6 Call signaling functions

Il utilise les signalisations H.225 pour établir une connexion entre deux "endpoints" H.323. le canal de signalisation d'appel est ouvert par priorité à l'établissement d'un canal H.245 et de tout autre canal entre deux "endpoints" H.323. Si le système ne possède pas de gatekeeper, le canal de signalisation d'appel est ouvert entre les deux points qui ont engendré l'appel. Si le système possède un Gatekeeper, le canal de signalisation d'appel est ouvert entre l'endpoint et le gatekeeper ou entre les "endpoints" eux-mêmes ".

4.2.7 Diagramme d'une connexion H.323

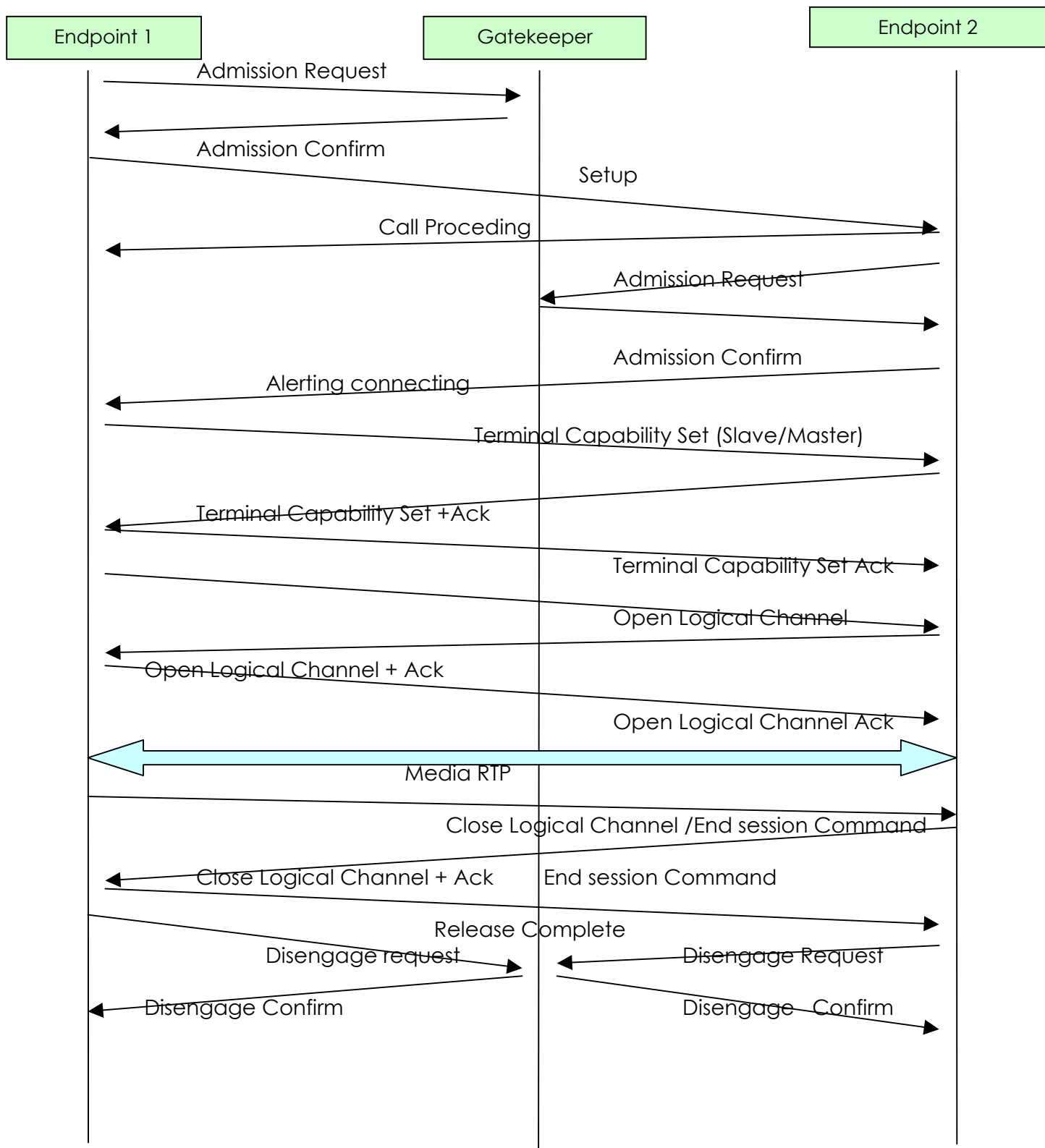


Figure 4.6

Diagramme d'établissement d'appel suivant le modèle H.323

4.2.8 Principe de fonctionnement (sans gatekeeper)

Soit deux utilisateurs attachés à deux terminaux distincts (possédant donc deux adresses IP différentes). L'un d'eux désire établir avec l'autre une communication vocale sous l'aspect du protocole H. 323. Cette situation est la même si on remplace les terminaux IP par des terminaux analogiques connectés à un réseau public de télécommunication.

Ce procédé requiert l'ouverture de deux connexions de type TCP. L'une pour le canal "Call Setup", l'autre pour le canal "Call Control".

Phase 1: l'appelant déclenche l'ouverture du canal "Setup"

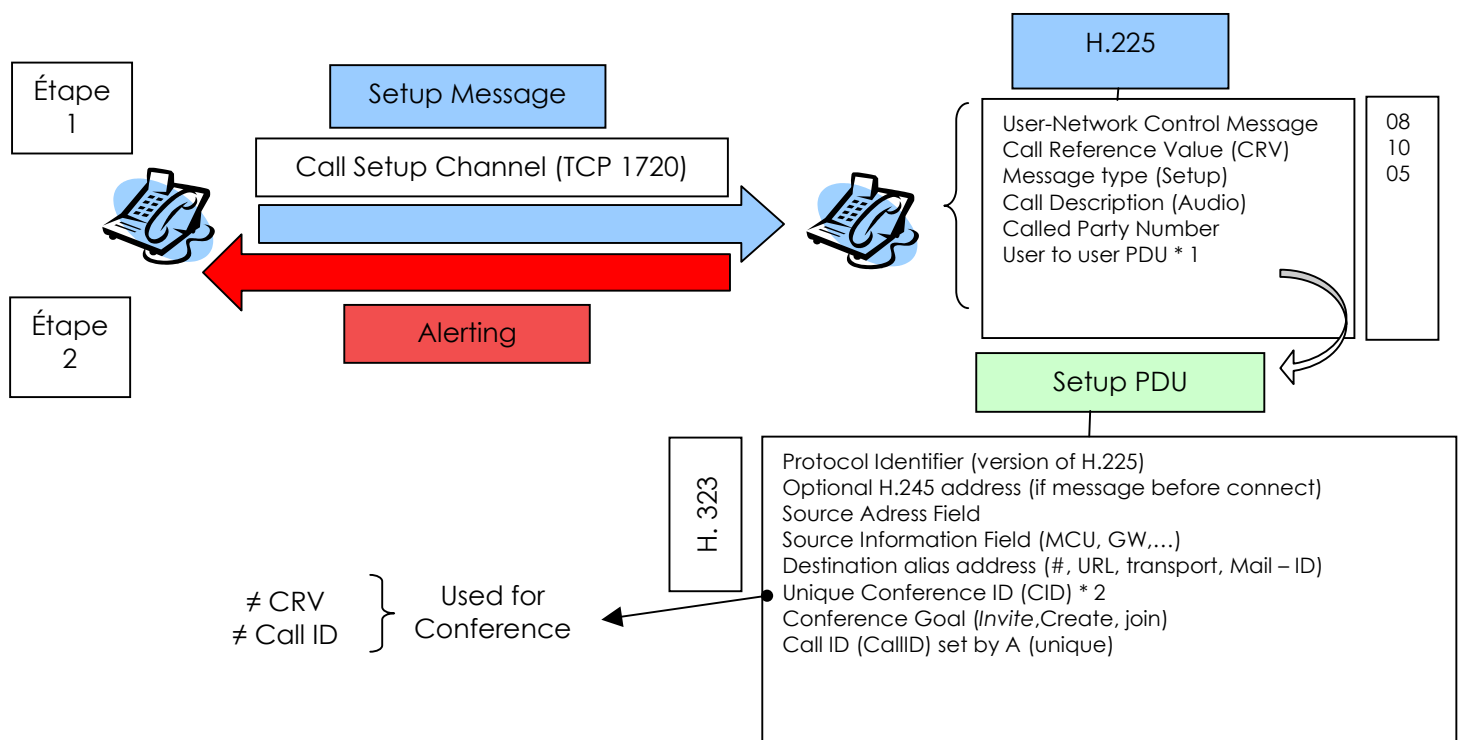


Figure 4.7
Phase 1 étapes 1 & 2

Un message "Alerting" doit être envoyé par l'appelé lorsqu'il accuse bonne réception du message "Setup". Lorsque le message "Alerting" est envoyé, l'utilisateur a trois minutes pour accepter ou refuser l'appel. Si l'appel est accepté par l'action de "décrocher", un message "Connect" est envoyé (voir figures 4.7 & 4.8).

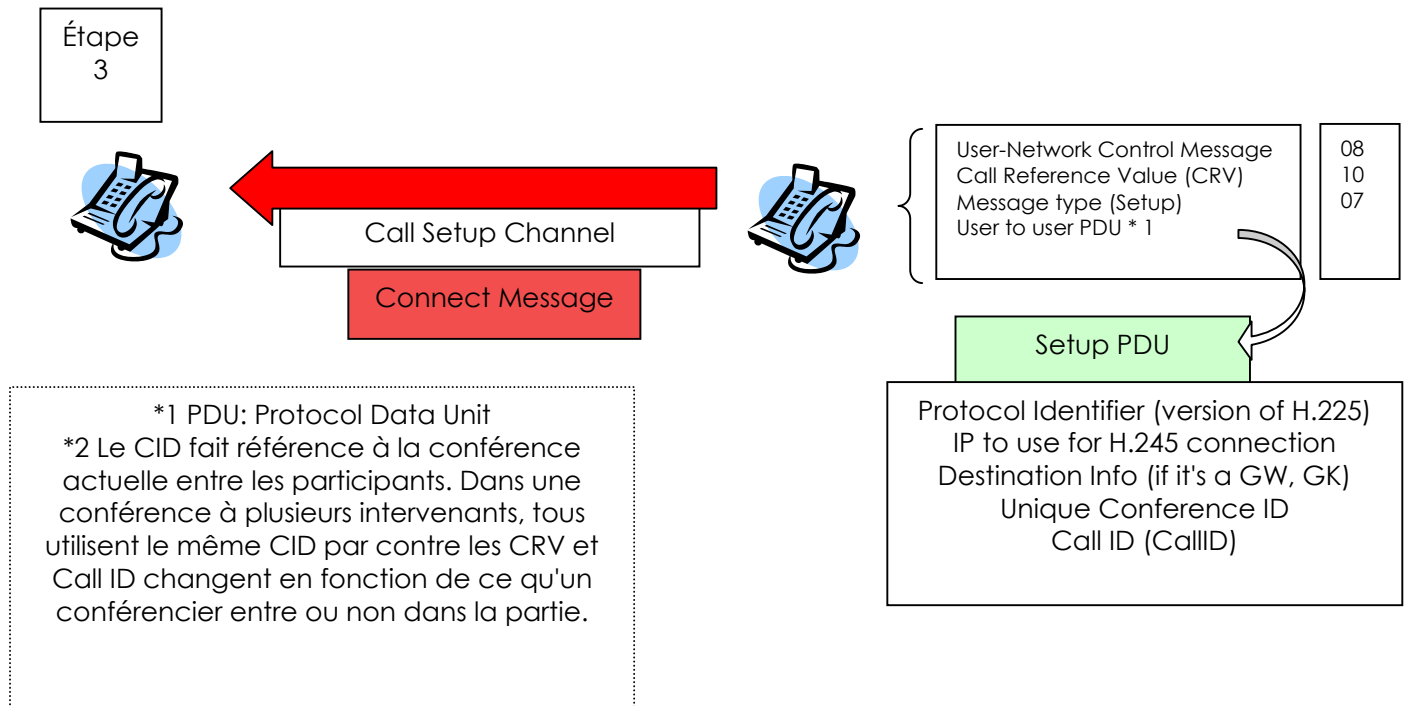


Figure 4.8
Phase 1 étape 3

Phase 2: Ouverture du canal de contrôle

Les messages de contrôle d'appel et les messages d'échange de capacité sont envoyés sur le second canal: le canal de contrôle. Ce canal de contrôle est ouvert par le terminal appelant sur un port défini du terminal appelé. Ce canal est ouvert dès réception d'un signal "Alert, Call Processing ou Connect" (Voir figure 4.9).

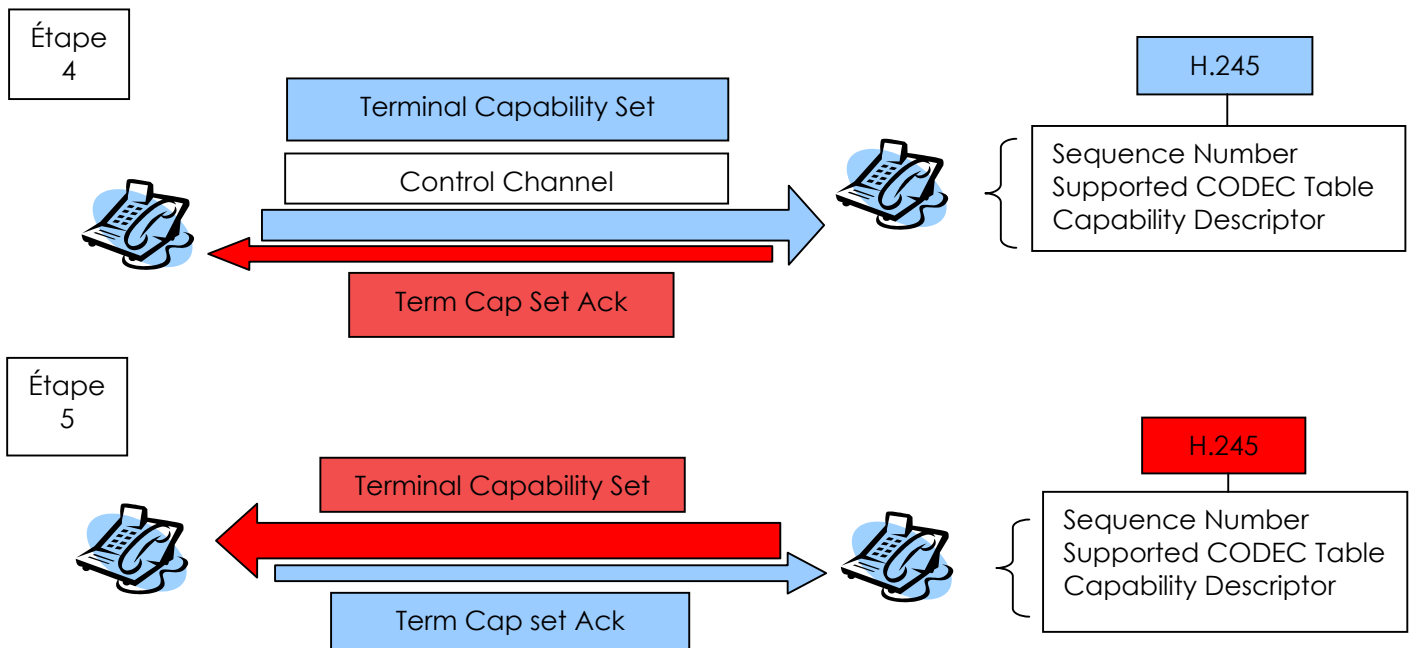


Figure 4.9
Phase 2 étapes 4 & 5

Phase 3: Préparation à la conversation

Cette phase débute par l'ouverture des canaux de communications vocales. Les données seront transportées dans plusieurs canaux logiques unidirectionnels sauf dans le cas des données T.120.

Le terminal appelant, qui débute cette phase, envoie un message "Open Logical Channel" vers le terminal appelé (voir figure 4.10).

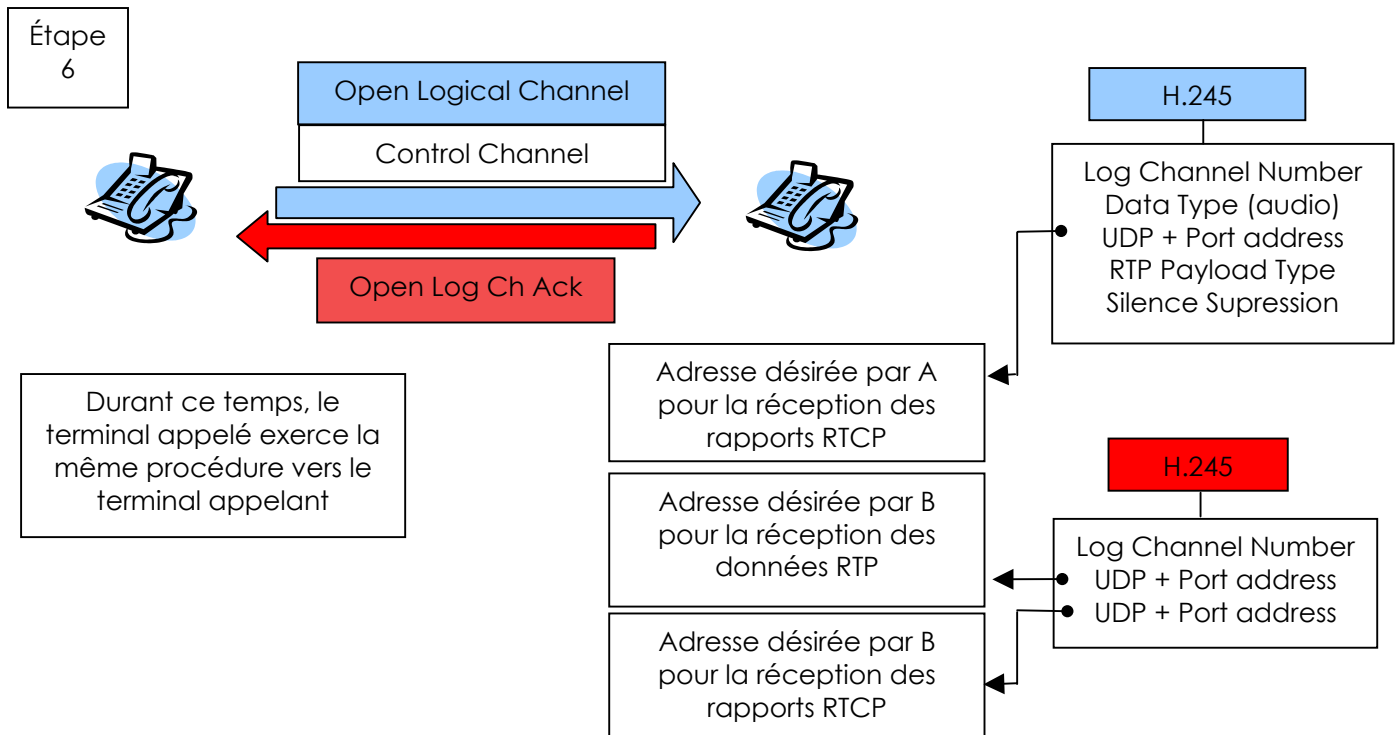


Figure 4.10
Phase 3 étape 6

Phase 4: Conversation

Les deux interlocuteurs peuvent à présent communiquer oralement. Les informations vocales sont transmises sous forme de paquets RTP. Les paquets RTCP envoyés par A permettent à B de synchroniser plusieurs flux RTP et également d'évaluer le taux de données RTP. La phase de conversation est représentée par la figure 4.11.

Les paquets RTCP envoyés par B permettent quant à eux de donner une idée à A de la qualité du service entre A et B. Les messages transmis par RTCP contiennent:

- La fraction de données perdues durant la transmission depuis le dernier paquet réceptionné
- Le taux cumulé de paquets perdus
- Des informations sur les jitters
- Le taux le plus élevé de séquences reçues

H.323 doit s'occuper de réguler le trafic de données envoyées de manière à rendre la communication la meilleure possible en réduisant les pertes de paquets. Ceci se réalise en diminuant la vitesse de transmission.

H.323 recommande de n'ouvrir que trois sessions maximum entre deux terminaux. La première pour le trafic vocal, la seconde pour le trafic de données et la troisième pour le trafic vidéo.

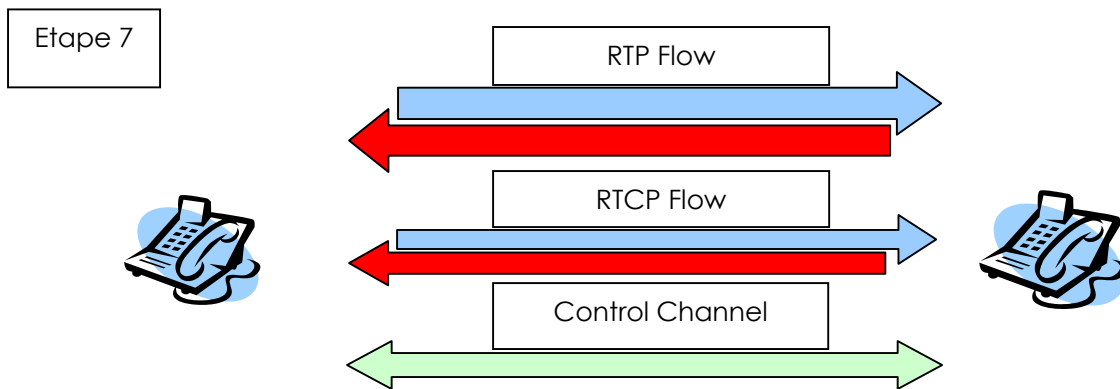


Figure 4.11
Phase 4 étape 7

Phase 5: Terminaison

Si le terminal A désire arrêter la communication, il envoie un message "Close Logical Channel" sur le canal de contrôle, pour chacun des canaux logiques qu'il a ouverts. Le terminal B doit accepter ces commandes et en accuse bonne réception en envoyant un message "Close Logical Channel Ack".

Lorsque tous les canaux logiques sont refermés, le terminal A envoie un message de fermeture de session: "End Session". Il attend le même message de la part du terminal B et referme le canal de contrôle. Vient ensuite la fermeture du canal de signalisation par l'envoi réciproque d'un message "Release Complete" sur ce canal. Le canal est alors refermé et la communication est terminée.

4.2.9 Principe de fonctionnement avec gatekeeper

Il s'agit ici de la représentation du cas le plus couramment rencontré dans la téléphonie sur IP. Nous décrivons les étapes du processus qui permettent à un utilisateur de contacter son correspondant à l'aide d'un Gatekeeper. Ce cas-ci en particulier considère les deux utilisateurs situés chacun sur un réseau IP

Pour rappel, le Gatekeeper est considéré comme une sorte de bottin électronique contenant un ensemble d'informations sur des utilisateurs qui s'y sont enregistrés. Ces informations sont des associations entre des noms d'adresse appelés "Alias" et les adresses physiques du réseau qui y correspondent. Le Gatekeeper est également l'élément le plus complexe en téléphonie H.323. Il a été défini dans la première version H.323 v1 et ses fonctionnalités furent clarifiées dans H.323 v2.

Le déroulement du processus s'effectue selon les étapes suivantes:

1. Découverte du GK et processus d'enregistrement
2. Demande, par le terminal, d'une permission d'appeler un client
3. Signalisation
4. Terminaison

Etape 1

Lorsque le client est démarré, commence la première étape en laquelle consiste la recherche du gatekeeper le "plus" proche. Cette recherche est amorcée par l'émission d'une requête broadcast appelée "GRQ" (GK Request). Le terminal peut aussi très bien spécifier dans cette requête s'il désire se connecter à un GK en particulier. Dans cette éventualité,

il précise l'adresse du GK qu'il désire contacter. On pourrait également effectuer une requête GRK unicast en y précisant l'adresse d'un GK particulier.

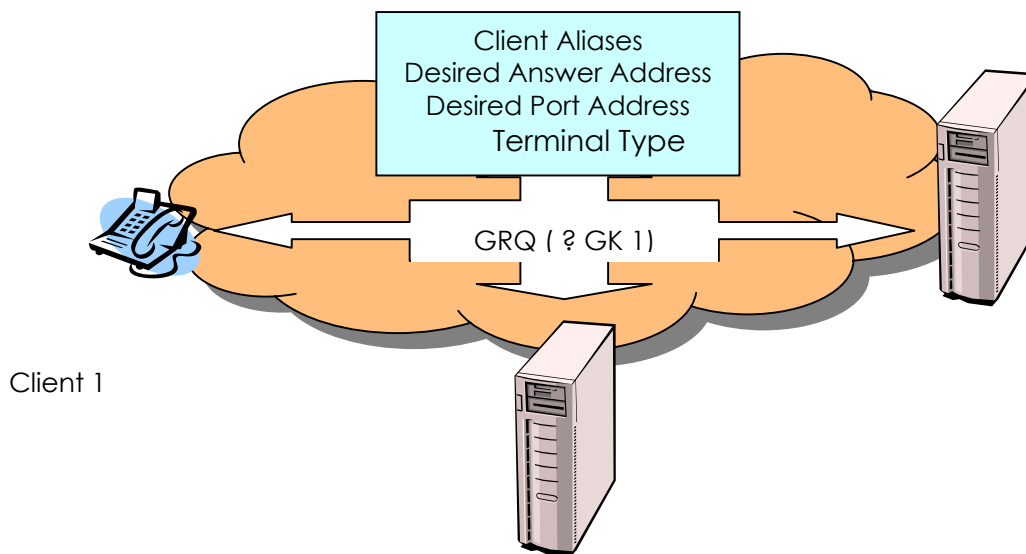


Figure 4.12
Recherche du gatekeeper

Le client insère également la liste des alias qui permettent de l'identifier dans la requête GRQ ainsi que l'adresse et le port sur lequel il désire que le GK lui renvoie une réponse.

Il est une règle de bonne pratique de limiter dans un premier temps la durée de vie de la requête pour que seuls les GK du domaine local répondent les premiers. Si, pour cette requête, aucun GK ne répond, on prolonge la durée de vie progressivement pour atteindre les GK des domaines avoisinants. Le client reçoit ensuite une réponse d'un GK suite à la requête GRK

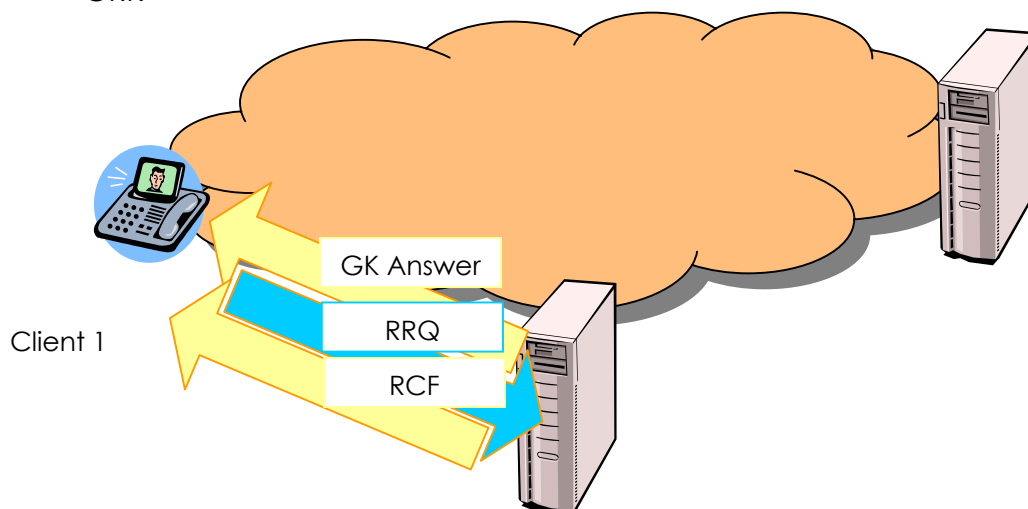


Figure 4.13
Réponse du gatekeeper

Après avoir obtenu la réponse du GK, le client lui émet une requête de demande d'enregistrement (RRQ) "Registration Request". Cette requête RRQ est généralement émise sur un port particulier du GK et est accompagnée par l'adresse qui permettra la signalisation des appels. Le GK émet une réponse RCF "Registration Confirm" vers le terminal. En même temps, le GK assigne un numéro d'identification unique au terminal qu'il utilisera dans chacune des transactions RAS entre lui même et le terminal. Le terminal est maintenant enregistré auprès du GK désiré.

Etape 2

Le terminal va passer une requête d'appel vers un correspondant dont il connaît une adresse Alias mais pas l'adresse physique. Cette requête sera émise vers le GK car nous nous trouvons dans le modèle à appel routé par un GK. Mais avant tout il y a d'abord une requête RAS du terminal vers le GK. Cette requête est une requête ARQ pour "Admission Request". La requête possède un ensemble d'informations émises par le terminal, comme son identifieur unique (qui lui a été assigné par le GK), l'alias du correspondant, le type d'appel (point à point), un numéro unique qui permettra d'identifier la connexion établie entre le client1 et le client2 et des informations sur les CODECS qui sont susceptibles d'être utilisées.

Deux cas sont envisageables pour la signalisation:

- GK Routed
- Direct

Dans le premier cas, la signalisation passera par l'intermédiaire du GK, dans l'autre la signalisation sera établie directement entre les deux clients.

Si le GK accepte l'appel, il émet vers le client un message ACF pour "Admission Confirm".

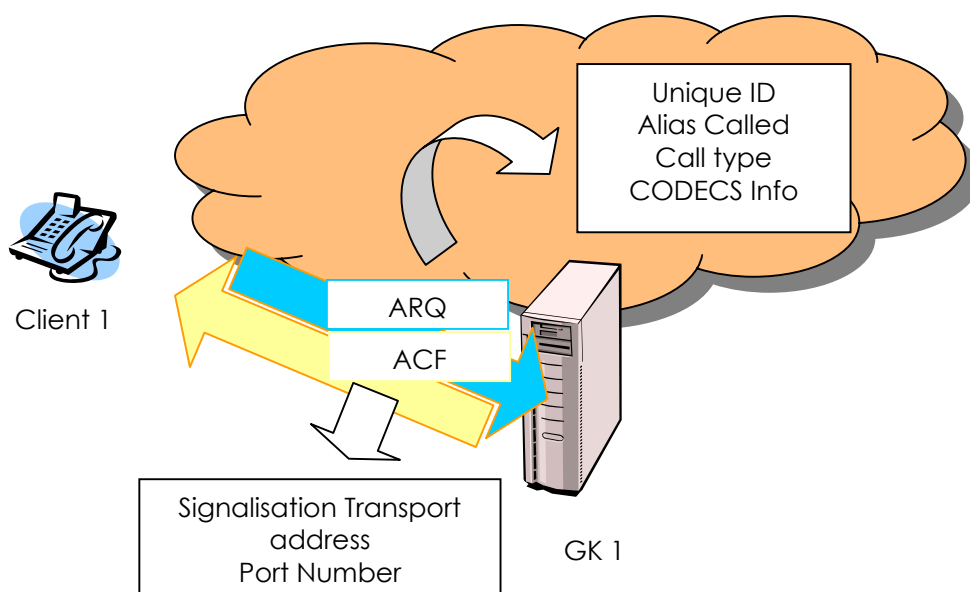


Figure 4.14
Requête ARQ

Etape 3

Le terminal peut émettre un message "SETUP" vers l'adresse et le port fournis par le GK. Il s'agira de l'adresse du client2 que le client1 désire contacter.

La suite des étapes se déroule comme dans le modèle décrit dans 4.2.8.

4.3 SESSION INITIATION PROTOCOL (SIP)

4.3.1 Définition de SIP

SIP est un protocole développé par le groupe de travail MMUSIC (Multiparty Multimedia Session Control) de l'IETF (Internet Engineering Task Force). Il est défini dans le RFC 2543 de mars 1999. Il s'agit d'un protocole complémentaire aux protocoles déjà développés par l'IETF comme RTP. Il est aujourd'hui le protocole qui attire le plus l'attention des développeurs de logiciel VoIP car il est sensiblement plus simple à exploiter que H.323. On peut également rencontrer SIP comme protocole pour envoyer des messages instantanés ou renseigner sur des événements.

D'un point de vue architectural, SIP est plus "léger" qu'H.323 car il n'a pas adopté le standard Q.931 ou H.245 qui est le protocole de signalisation entre les terminaux H.323 et différent à quelques points de vue de Q.931, connue du monde PSTN. H.323 a gardé comme base de signalisation Q.931 dans l'idée justement de ne pas trop se distinguer du monde PSTN. Dans un futur proche, les protocoles SIP et H.323 coexisteront, c'est pourquoi on parle d'interconnexion SIP/H.323.

Comme avec H.323, les données multimédia transitent par le protocole RTP. La différence réside dans le contrôle de signalisation. Ces fonctions sont exécutées par le "Session Initiation Protocol" (SIP).

SIP est décrit comme un protocole de contrôle de la couche application. Il établit, modifie, et termine des conversations multimédias. Il ressemble un peu, en syntaxe, à HTTP et SMTP, car il permet d'établir une session entre des interlocuteurs identifiés par des adresses similaires à des adresses e-mail.

La mobilité personnelle est une des fonctionnalités de SIP. Un utilisateur peut garder le même numéro malgré qu'il soit connecté à des terminaux d'adresses physiques différentes. Egalement, comme avec le principe des e-mails, plusieurs adresses d'identificateurs peuvent référencer un même terminal. Inversement, une adresse SIP peut référencer plusieurs terminaux différents.

4.3.2 Architecture SIP

Plusieurs entités contribuent au fonctionnement de SIP: Les *Users Agents*, *Registrars*, *proxy* et *Redirect Servers*.

User Agent (UA)

C'est une application qui réside sur une "end station". Le UA est composé de deux parties: le client (UAC: User Agent Client) et le serveur (UAS: User Agent Server). Le client envoie les requêtes SIP lorsqu'on initialise un appel. Le UAS est une application qui contacte l'utilisateur si un appel lui est destiné.

Redirect Server

Un utilisateur peut envoyer une requête d'invitation à une autre personne par l'intermédiaire d'un serveur de re-direction. Ce serveur se chargera de retrouver cette personne et de renvoyer les informations nécessaires au client appelant, pour qu'il puisse établir une connexion directe avec l'interlocuteur désiré.

Proxy Server

Un utilisateur peut envoyer une requête d'invitation à une autre personne par l'intermédiaire d'un serveur proxy. Le proxy tentera également de localiser le destinataire de la communication, mais à la différence du "redirect server", il tentera d'établir une connexion entre les intéressés. Le serveur Proxy agit comme serveur et client à la fois, c'est-à-dire qu'il peut recevoir et envoyer des requêtes.

Registrar Server

Le registrar server est un serveur qui accepte les requêtes d'enregistrement (Register Request). Il peut également implémenter d'autres fonctionnalités SIP comme servir de serveur de proxy.

Un registrar Server permet de garder traces des localisations des utilisateurs. Cette fonctionnalité provient du fait que si un utilisateur se connecte à Internet par le biais d'un Internet Service Provider (ISP), on lui associe une adresse IP dynamique.

Cette même remarque est valable pour les personnes connectées au réseau LAN par un serveur DHCP. Il est donc nécessaire que ces utilisateurs puissent être joignables indépendamment de leur adresse IP. Une table d'association des adresses IP et SIP est maintenue et mise à jour sur ce serveur.

Exemple d'adresse SIP : "mailto:test@164.15.81.2"

Un utilisateur envoie des informations personnelles à propos de sa localisation et s'auto enregistre. C'est à partir de cette information qu'il pourra être contacté par un autre interlocuteur. Grâce à cette méthode, un utilisateur peut être joignable depuis n'importe quelle localisation. On peut donc parler de téléphonie mobile sur le réseau de commutation par paquets. Les informations envoyées au registrar server, permettront de savoir quel système utiliser pour le contacter.

4.2.3 Transaction SIP

Une transaction SIP typique est représentée par la figure 4.15 où un terminal invite un second terminal à participer à une nouvelle session. Cette requête peut être modélisée par la syntaxe: `INVITE sip:joe@company.com` où `joe` est l'identificateur du correspondant sur le terminal dont le domaine est `company.com`.

Cette requête est émise vers le serveur proxy local qui recherchera, auprès d'un DNS, des informations concernant le domaine `company.com`. Ces informations reprennent l'adresse IP du serveur du domaine `company.com` traitant les requêtes SIP.

Le proxy local, une fois en possession de ces informations, émet vers le proxy associé au second domaine la requête d'invitation émise par le client 1.

Le second proxy dispose d'informations concernant la localisation du client `joe`. Seulement `joe` est pour l'instant connecté sous `j.user@university.edu`. Le second proxy renvoie cette information au Proxy 1 qui recherche auprès de son DNS l'adresse IP du serveur SIP associé au domaine `university.edu`. Il établit une nouvelle requête d'invitation vers ce troisième proxy.

Le troisième proxy consulte sa base de localisation à la recherche d'informations concernant `j.user@university.edu`. Il obtient que `j.user@university.edu` est associé à `j.smith@cs.university.edu` du même domaine `university.edu`. Il transmet ensuite sa requête d'invitation vers le serveur SIP local au sous-domaine `cs`.

Le proxy `cs`, connaît l'adresse IP à laquelle `joe` est finalement connecté et transmet à `joe` la requête d'invitation.

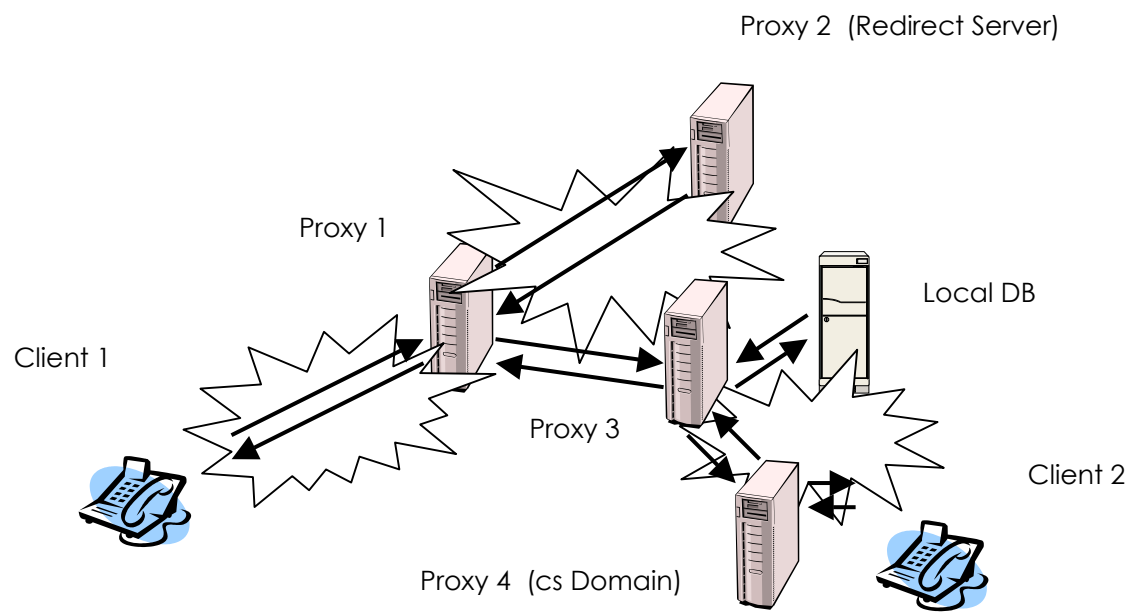


Figure 4.15
Transaction SIP

4.3.4 Schéma d'établissement de connexion SIP

4.3.4.1 Opérations SIP en mode "redirection"

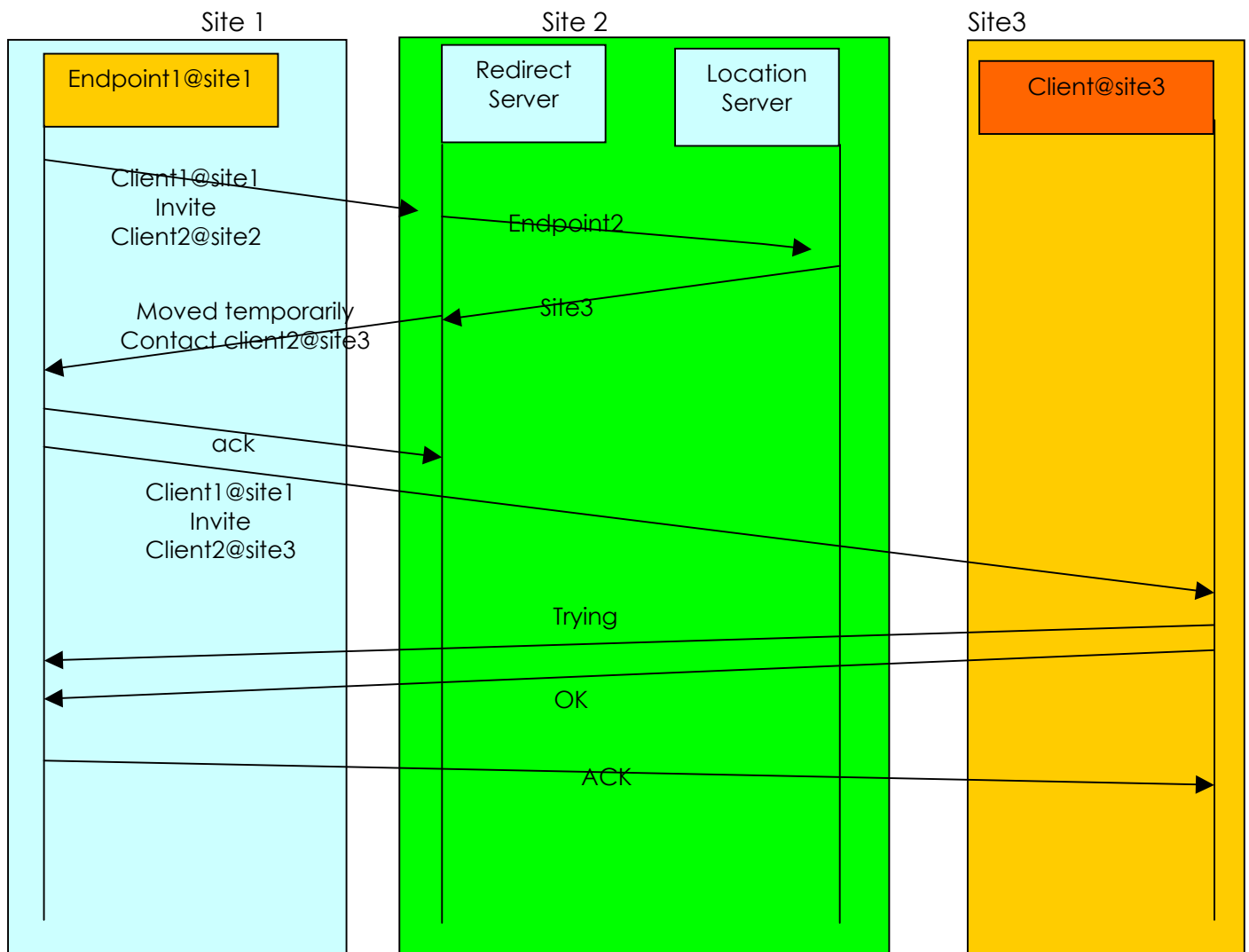


Figure 4.16
SIP en mode "redirection"

4.3.4.2 Opérations SIP en mode Proxy

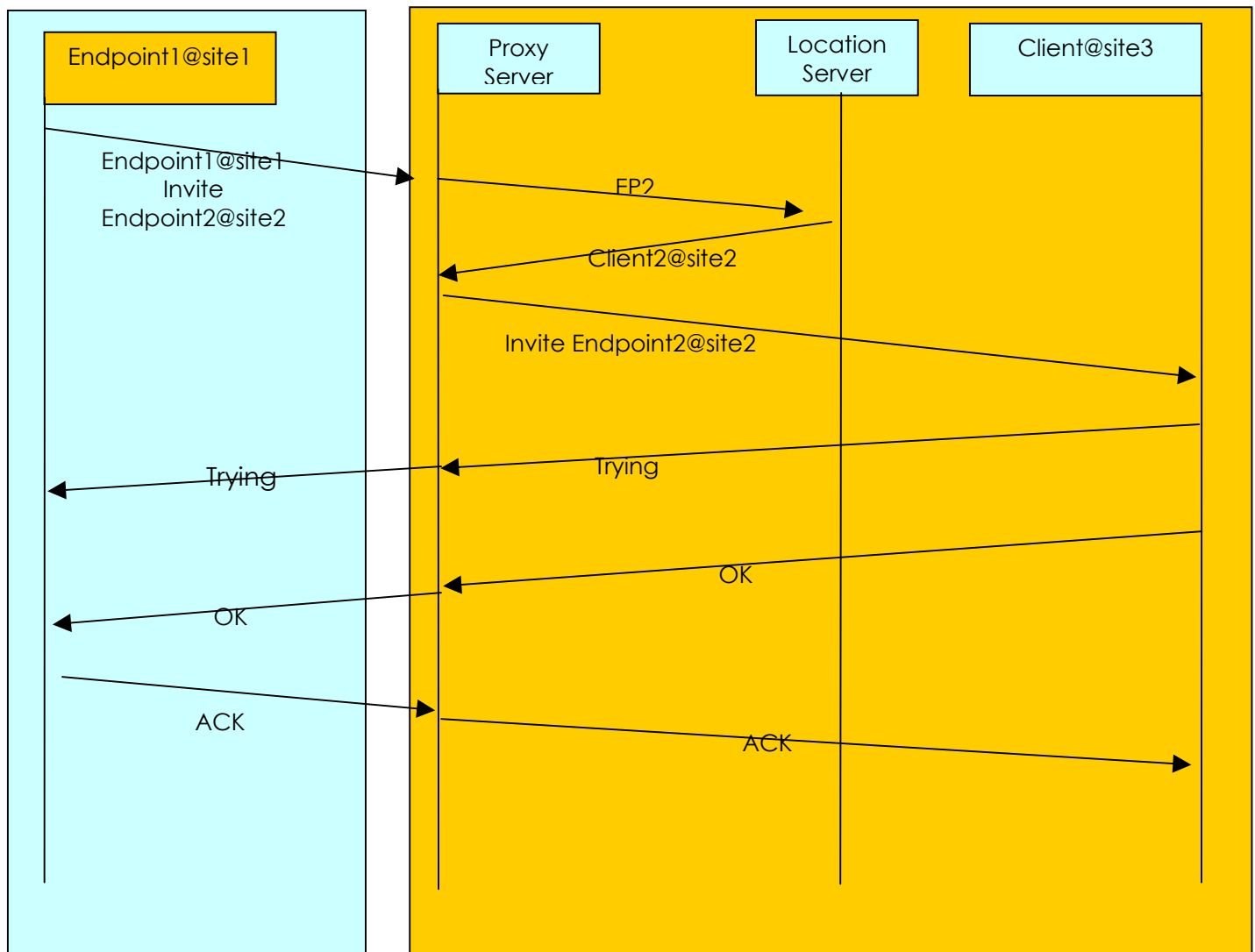


Figure 4.17
SIP en mode "proxy"

4.4 COMPARAISON SIP ET H.323

Les deux protocoles SIP et H.323 représentent les standards définis jusqu'à présent pour la signalisation à propos de la téléphonie sur Internet. Ils présentent tous deux des approches différentes pour résoudre un même problème.

H.323 englobe une partie de la signalisation Q.931 définie pour les réseaux ISDN. Par ce lien, on considère H.323 basé sur une approche traditionnelle de réseaux à commutation de circuits. Quant à SIP, il est plus léger car basé sur une approche similaire au protocole HTTP. Il ré-utilise d'ailleurs les structures, grammaires, codes d'erreurs et les mécanismes d'authentification d'HTTP. Tous deux utilisent le protocole RTP comme protocole de transfert des données multimédia. La table 4.2 reprend les protocoles contenus par H.323.

H.245 Control
H.225 Set-UP
H.332 large Conferences
H.450.1 Other Services
H.450.2 Other Services
H.450.3 Other Services
H.235 Security
H.246 Interoperability
Protocoles contenus par H.323

Table 4.2

Ensemble des protocoles de H.323

Au départ H.323 fût conçu pour la téléphonie sur les réseaux sans QoS, mais on l'adapta pour qu'il prenne en considération l'évolution complexe de la téléphonie sur Internet.

Pour donner une idée de la complexité du protocole H.323 par rapport à SIP, H.323 est défini en un peu plus de 700 pages et SIP quant à lui en moins de 200. H.323, on le verra plus tard, est conçu sur base d'une syntaxe de notation abstraite ASN.1 et SIP encode ses messages en simple texte. Cette manière de coder par SIP est propice à une mise au point simple au contraire de H.323.

La complexité de H.323 provient encore du fait de la nécessité de faire appel à plusieurs protocoles simultanément pour établir un service. SIP n'a pas ce problème.

L'évolution que connaissent les technologies liées à Internet force souvent les applications existantes à elles-même évoluer. De ce point de vue, SIP a gardé les leçons liées aux protocoles HTTP et SMTP en permettant des fonctionnalités extensibles et compatibles avec les versions antérieures. Il est

doté de requêtes par lesquelles il demande aux serveurs avec lesquels il entre en contact, de fournir certaines informations obligatoires. Les champs qui lui sont fournis mais dont il ignore le rôle (par exemple à cause d'une nouvelle implémentation) sont ignorés, mais il peut répondre par un code d'erreur en y spécifiant les champs incompris.

Pour H.323, on parle également d'évolution, mais la manière dont on peut le faire évoluer est moins simple. Il s'agit de compléter des syntaxes ASN par des champs qui sont propres à chacun des développeurs. De plus les terminaux H.323 ne sont pas dotés de la fonctionnalité de SIP par laquelle ils peuvent faire-part des champs incompris des requêtes qui leur sont destinées.

SIP ne requiert pas de compatibilité descendante. SIP est un protocole horizontal au contraire d'H.323: les nouvelles versions d'H.323 doivent tenir compte des fonctionnalités des anciennes pour continuer à fonctionner. Ceci entraîne pour H.323 de "traîner" un peu plus de code à chaque version.

H.323 ne reconnaît que les CODECS standardisés pour la transmission des données multimédia proprement dit alors que SIP, au contraire, peut très bien en reconnaître d'autres.

On pourrait continuer à comparer H.323 avec SIP plus longuement, mais il est mieux de se référer à la littérature abondante sur ce sujet. En conclusion on peut dire que H.323 et SIP remplissent heureusement les mêmes rôles mais suivant des complexités nettement différentes. SIP, quant à lui, est plus évolutif qu'H.323. Pour le choix entre une solution SIP ou une solution H.323 il sera bien de se renseigner sur des calculs de performance de chacun d'eux et d'alors adopter la meilleure solution suivant ses besoins.

	H.323	SIP
Similarité protocole	Q.931	HTTP
Volume	700 pages	200 pages
Syntaxe	ASN	Texte
Inter-action protocoles	✓	
Une tâche = plusieurs méthodes	✓	
redondance des fonctionnalités	✓	
expansibilité	✓ (mais complexe)	✓
Volume de code	De + en + volumineux avec les versions	Volume constant
Protocole	Vertical	Horizontal
CODECS	Standardisés	Nouveaux + Standardisés

Table 4.3

Comparaison SIP / H.323

Chapitre 5

H.323 à la trace...

Dans ce point nous avons une vue des paquets qui transitent sur le réseau IP lors de l'établissement de requêtes de connexion auprès d'un GK à partir d'un client NetMeeting. Nous ne ferons pas l'analyse des paquets porteurs d'informations multimédias, mais nous porterons l'attention plus particulièrement sur les requêtes RAS. Ce sont ces requêtes RAS qui sont importantes pour la suite du travail. Il est donc utile d'avoir une vue de leur représentation. Ce point permet également de mieux illustrer quelques-uns des états représentés dans le chapitre 4.2.7.

Les images issues dans ce point sont extraites du logiciel "Network Monitor" de Microsoft. Ce logiciel n'est pas disponible dans la version standard de Windows NT server car il est doté d'une librairie permettant de comprendre le protocole H.323. Des logiciels similaires peuvent être retrouvés sur Internet et sont peut être plus complets dans le traitement de l'information perçue ou offrent certainement plus de fonctionnalités concernant le traitement des paquets RTP.

5.1 VISUALISATION DE LA SITUATION

Nous représentons dans ce point la situation telle que celle utilisée pour modéliser l'essai. Il faut rappeler que cet essai ne requiert que l'utilisation de deux PC disposant l'un d'un client NetMeeting, l'autre du serveur OpenGatekeeper. Les deux ordinateurs tournent sous un environnement Windows. Un des ordinateurs devra contenir le logiciel "Network Monitor".

Les deux ordinateurs sont connectés sur un réseau LAN de type coaxial. Le réseau local n'est pas doté d'un switch.

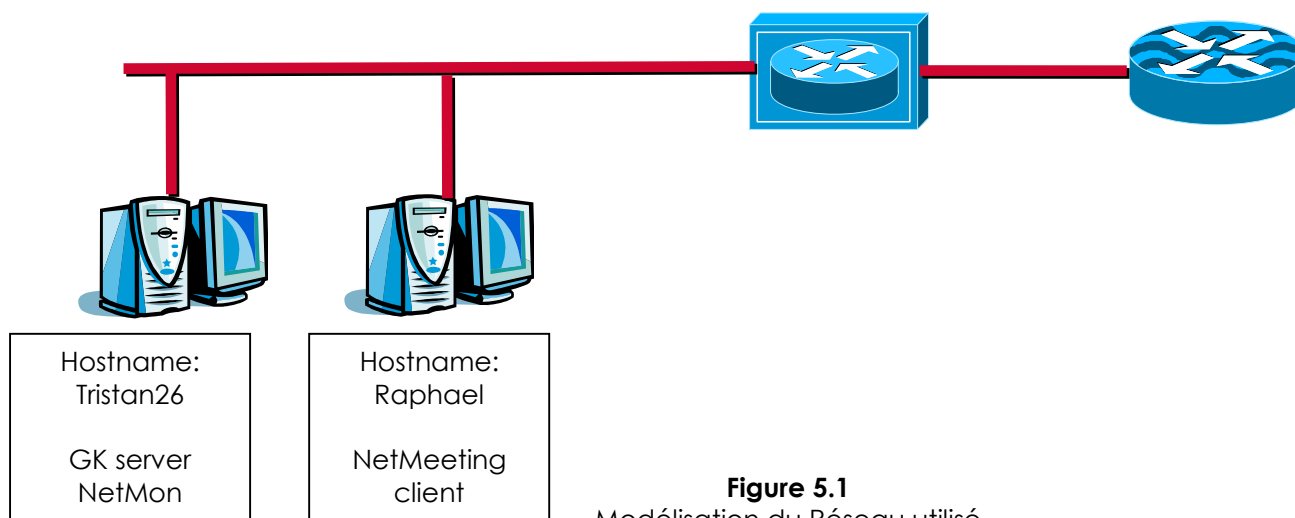


Figure 5.1
Modélisation du Réseau utilisé

Le client NetMeeting est configuré de manière à obtenir tous les appels routés par le Gatekeeper. De plus on fournit dans NetMeeting l'adresse du Gatekeeper de manière à ne pas utiliser la fonctionnalité de "gatekeeper discovery". Il faut ajouter que NetMeeting est également configuré pour se faire enregistrer auprès du Gatekeeper sous l'alias "111". Il n'y a aucune configuration du client qui limite la bande passante à utiliser.

Les ordinateurs fonctionnent, comme on l'a dit, en réseau et sont dotés d'une configuration réseau qui admet les échanges TCP/IP. Les deux ordinateurs sont situés sur le même segment et plus particulièrement dans le cas présent sur un segment du gate 78. Ils possèdent donc une adresse IP de la forme 164.15.78.XXX. On retrouve sur la figure 5.1 l'ensemble de ces informations réunies.

5.2 MESSAGES RAS: REQUÊTES RRQ ET RCF

Avant d'aller plus loin dans la discussion, il faut s'arrêter pour expliquer les différents champs qui apparaissent dans les figures, qui seront présentées ci-après, et qui représentent les vues obtenues à partir du logiciel Network Monitor. Les figures représentent l'ensemble des paquets qui ont été capturés lors du fonctionnement du logiciel Network Monitor. Elles comportent principalement trois zones distinctes:

- Une zone explicitant des informations sur les trames capturées
- Une zone qui, pour une trame, en explicite l'encapsulation des protocoles et qui pour chacun des protocoles en décrit le contenu
- Une zone qui explicite de manière hexadécimale et ASCII le contenu de la trame sélectionnée.

La première figure que nous représentons relate la requête RRQ (Registration Request) issue du client Netmeeting vers le GK. On remarque que dans la première zone on a deux trames qui ont été capturées. Nous nous intéresserons d'abord à la première qui correspond à la requête RRQ. La seconde, nous le verrons, est la réponse du GK à cette requête et est une réponse RCF qui signifie Registration Confirm. Nous le voyons, cette première trame est un message RAS comme indiqué.

Un simple coup d'œil sur la zone 2 nous renseigne sur l'encapsulation des protocoles. Nous voyons qu'il s'agit d'éthernet comme protocole de la couche physique et qu'en remontant, on obtient IP qui encapsule par UDP un message H.225 de requête RRQ. Cette requête RRQ contient l'adresse du terminal, qui effectue la requête, sous forme hexadécimale ainsi que l'alias du terminal. Ces informations sont utiles au gatekeeper pour qu'il puisse mettre à jour ses tables de renseignement.

Les "  " renseignent sur les champs importants des figures.

La seconde figure relate la réponse du gatekeeper. Cette réponse est une RCF (registration confirm). Cela signifie que notre gatekeeper a accepté la requête RRQ et a enregistré les informations concernant le client.

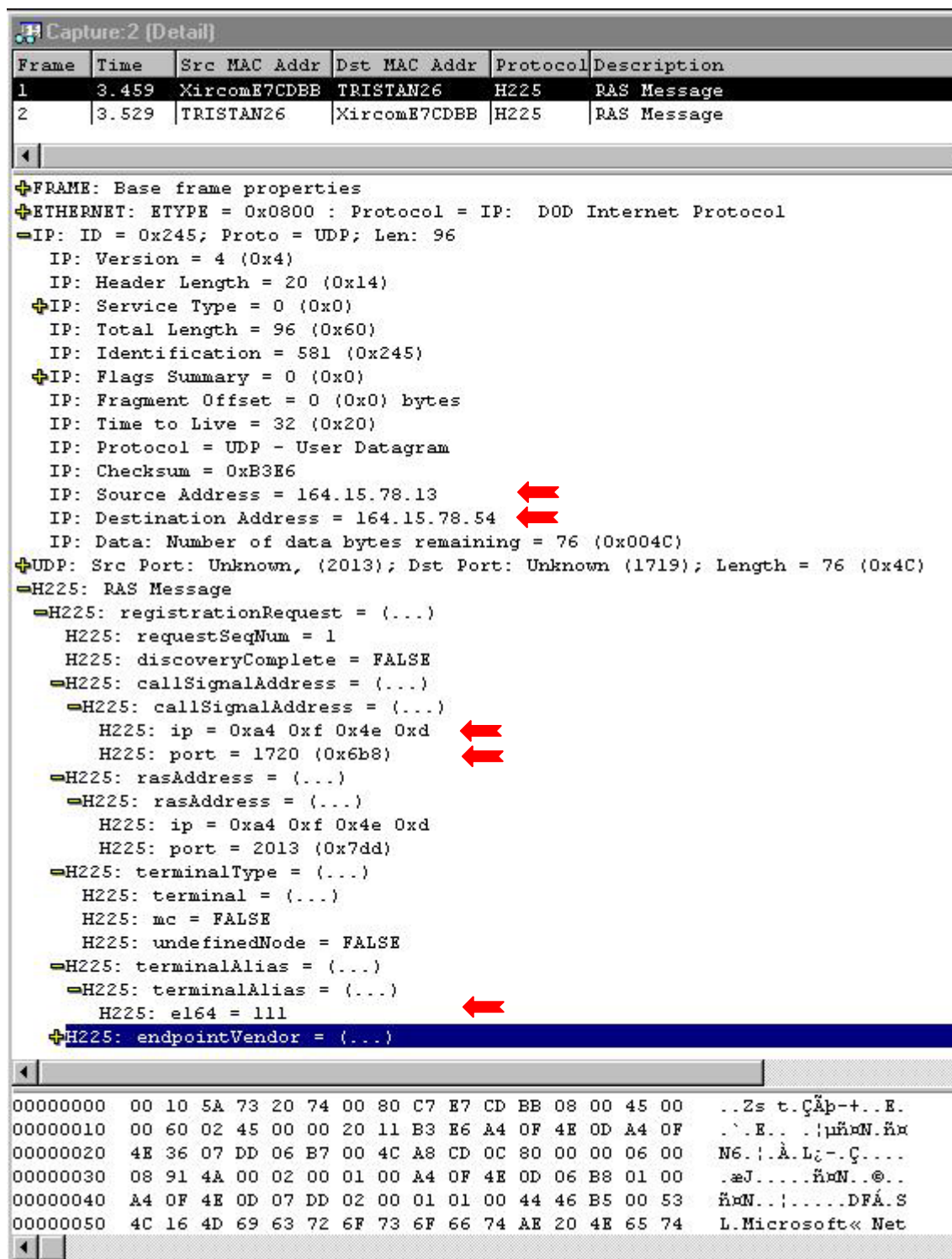


Figure 5.2
Message RRQ

Capture:2 (Detail)					
Frame	Time	Src MAC Addr	Dst MAC Addr	Protocol	Description
1	3.459	XircomE7CDBB	TRISTAN26	H225	RAS Message
2	3.529	TRISTAN26	XircomE7CDBB	H225	RAS Message

+

FRAME: Base frame properties

+

ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol

=

IP: ID = 0x5D2D; Proto = UDP; Len: 139

IP: Version = 4 (0x4)

IP: Header Length = 20 (0x14)

+

IP: Service Type = 0 (0x0)

IP: Total Length = 139 (0x8B)

IP: Identification = 23853 (0x5D2D)

+

IP: Flags Summary = 0 (0x0)

IP: Fragment Offset = 0 (0x0) bytes

IP: Time to Live = 128 (0x80)

IP: Protocol = UDP - User Datagram

IP: Checksum = 0xF8D2

IP: Source Address = 164.15.78.54

IP: Destination Address = 164.15.78.13

IP: Data: Number of data bytes remaining = 119 (0x0077)

+

UDP: Src Port: Unknown, (1719); Dst Port: Unknown (2013); Length = 119 (0x77)

=

H225: RAS Message

=

H225: registrationConfirm = (...)

H225: requestSeqNum = 1

=

H225: callSignalAddress = (...)

=

H225: callSignalAddress = (...)

H225: ip = 0xa4 0xf 0x4e 0xd

H225: port = 1720 (0x6b8)

H225: RegistrationConfirm_gatekeeperIdentifier = Opengate: tristan26

H225: endpointIdentifier = 0: Opengate: tristan26

H225: RegistrationConfirm_timeToLive = 600

H225: RegistrationConfirm_willRespondToIRR = FALSE

H225: makeCall = FALSE

H225: useCKCallSignalAddressToMakeCall = FALSE

H225: answerCall = FALSE

H225: useCKCallSignalAddressToAnswer = FALSE

00000000	00 80 C7 E7 CD BB 00 10 5A 73 20 74 08 00 45 00	.ÇÃp-+...Zs t..E.
00000010	00 8B 5D 2D 00 00 80 11 F8 D2 A4 0F 4E 36 A4 0F	.i)-...Ç."ÊÑN6ÑÑ
00000020	4E 0D 06 B7 07 DD 00 77 F6 07 12 40 00 00 06 00	N..À.!.w+...@....
00000030	08 91 4A 00 02 01 00 A4 0F 4E 0D 06 B8 24 00 4F	.æJ....ÑÑN...@\$.0
00000040	00 70 00 65 00 6E 00 67 00 61 00 74 00 65 00 3A	.p.e.n.g.a.t.e.:.
00000050	00 20 00 74 00 72 00 69 00 73 00 74 00 61 00 6E	. .t.r.i.s.t.a.n

Figure 5.3
Message RCF

Page 46

5.3 MESSAGES RAS: REQUÊTES ARQ ET ARJ

Dans cette partie nous montrons les paquets qui transitent lors d'une requête d'admission par le client NetMeeting et d'une réponse de rejet donnée par le gatekeeper. La situation est obtenue en demandant d'établir une connexion par le client NetMeeting vers un autre client NetMeeting non enregistré auprès du gatekeeper. Le gatekeeper constatant qu'aucun abonné ne répond à l'alias présenté émet une réponse ARJ.

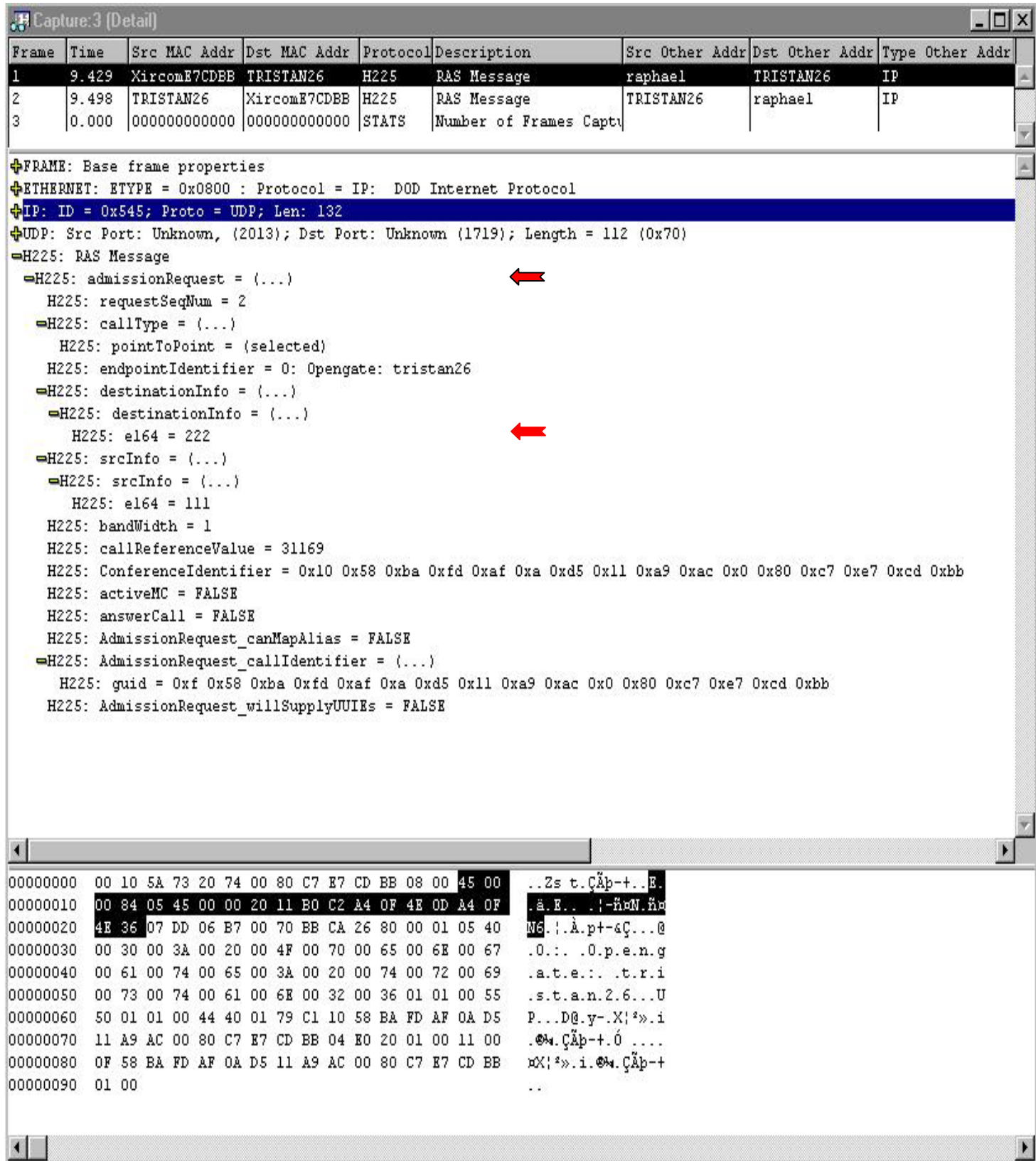





Figure 5.4
- Message ARQ

Frame	Time	Src MAC Addr	Dst MAC Addr	Protocol	Description	Src Other Addr	Dst Other Addr	Type Other Addr
1	9.429	XircomE7CDBB	TRISTAN26	H225	RAS Message	raphael	TRISTAN26	IP
2	9.498	TRISTAN26	XircomE7CDBB	H225	RAS Message	TRISTAN26	raphael	IP
3	0.000	000000000000	000000000000	STATS	Number of Frames Capt			

+FRAME: Base frame properties
 +ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol
 +IP: ID = 0x752E; Proto = UDP; Len: 32
 +UDP: Src Port: Unknown, (1719); Dst Port: Unknown (2013); Length = 12 (0xC)
 =H225: RAS Message
 =H225: admissionReject = (...) 
 H225: requestSeqNum = 2
 =H225: rejectReason = (...) 
 H225: AdmissionRejectReason_calledPartyNotRegistered = (selected) 

00000000 00 80 C7 E7 CD BB 00 10 5A 73 20 74 08 00 45 00 .Ç&p-+..Zs t..E.
 00000010 00 20 75 2E 00 00 80 11 E1 3C A4 0F 4E 36 A4 0F . u...Ç.B<R&N6R&
 00000020 4E OD 06 B7 07 DD 00 0C DF DF 2C 00 01 00 N..À.!--,...

La figure 5.5 représente la réponse donnée par le gatekeeper suite à l'ARQ du client NetMeeting. Il s'agit d'une réponse ARJ pour rejeter la requête. Le gatekeeper indique également la raison du refus. Il s'agit dans notre cas d'un refus par manque d'informations, dans la base de données, sur l'utilisateur appelé. En fait l'utilisateur demandé ne s'est pas fait enregistrer auprès du gatekeeper.

Chapitre 6

TTT-Services

6.1 DESCRIPTION

Le projet "Ten Telecom Tiphon¹-services" a pour but d'établir des services Voice Over IP standardisés et harmonisés sur les spécifications de TIPHON (Telecommunication and Internet Protocol Harmonization over Networks). Ces spécifications sont:

- la portabilité nationale et internationale
- La mobilité personnelle

Ce projet est la suite du projet "TTT-net" qui se préoccupait de l'interopérabilité entre les modules Gateways, Gatekeepers et terminaux. L'interopérabilité concerne les connexions entre les modules appartenant aussi bien à des domaines différents qu'à des équipements de fournisseurs différents, ce qui revient à dire que l'on travaille en milieu hétérogène.

6.2 PORTABILITÉ NATIONALE ET INTERNATIONALE

La portabilité globale s'acquiert en fournissant aux utilisateurs du service un numéro d'identification unique. Il est important de remarquer que ce numéro est associé à l'utilisateur et non pas à l'équipement. Ce sont donc des numéros personnels. Ainsi, même si l'utilisateur décide de changer de fournisseur de service, il peut garder son numéro sans le changer. Ces numéros sont délivrés par une autorité compétente (IA: Issuing Authority) et peuvent être conservés ad-vitam.

A cette fin on utilise des numéros de télécommunication personnels universels UPT (Universal Personal Telecommunications) ayant la structure :

878	878	9 chiffres (numéro du souscrivant)
-----	-----	------------------------------------

Puisque ces numéros ne présentent aucune caractéristique propre aux fournisseurs, une portabilité totale est assurée. Il est à remarquer que le

¹ Ten Telecom encourage le déploiement de services et d'applications de télécommunications trans-Européennes dans les domaines d'intérêts généraux, en fournissant une aide sur la préparation de business plan ou sur des phases initiales d'investissements.

second groupement "878" a été attribué par l'ITU au projet TTT service, mais qu'à priori des autres nombres sont acceptés.

6.3 CARACTÉRISTIQUES D'IMPLEMENTATION

Comme on vient de le voir, il y a un organisme qui possède la faculté d'attribuer à chaque utilisateur un numéro unique. Celui-ci met à jour une base de données, accessible globalement, dans laquelle il y a une association entre l'UPT de l'utilisateur et son fournisseur de service local. Cette base de données est une base de données "Tiphon Resolution Service (TRS)". Elle est conçue pour répondre à des requêtes provenant de services divers.

La figure 6.1 représente la structure de la base de données nécessaire à la résolution des requêtes de numéros internationaux UPT. Lorsqu'il veut se faire enregistrer dans la base de données, l'utilisateur entre en contact soit avec une "Number Usage Authority (NUA)" ou soit avec un fournisseur de services UPT (UPTSP). La NUA s'assure que le numéro qu'acquerra l'utilisateur est bien unique.

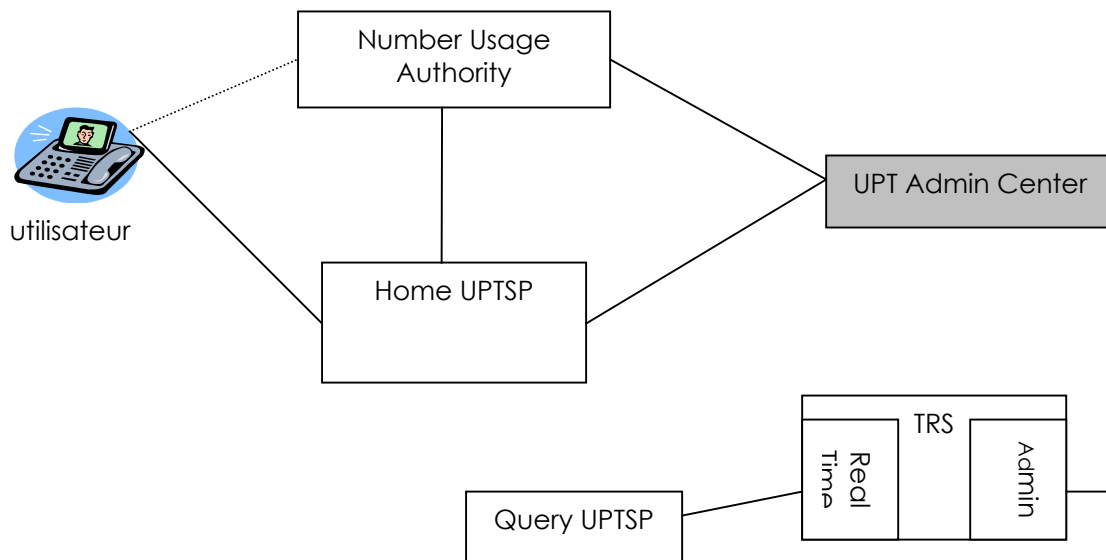


Figure 6.1
Structure de la base de données

A chaque numéro délivré est associé, dans la base de données TRS UPT, l'identificateur de l'UPTSP auquel souscrit l'utilisateur. Ceci permettra de localiser le terminal physique auquel est connecté actuellement l'utilisateur lorsqu'un appel lui est destiné. En effet, l'utilisateur souscrivant un numéro UPT possède un profil local à l'UPTSP dans lequel il précise l'adresse physique du terminal sur lequel il désire recevoir les appels destinés à son numéro UPT.

Schématiquement, lorsqu'un appel est destiné à un utilisateur UPT, le système de signalisation consulte la base de données UPT RTS qui le renseigne sur l'UPTSP du destinataire, et l'UTPSP contient les informations nécessaires au routage pour acheminer la communication vers le terminal physique auquel l'utilisateur est connecté.

La partie Temps Réel de TRS se charge de la récupération des données de routage tandis que la partie administrative se charge de l'enregistrement des Home UPTSP de tous les utilisateurs et de ce que l'attribution des numéros UPT soit unique.

6.4 SCÉNARIOS D'ENREGISTREMENT

6.4.1 Concepts

Tout utilisateur désirant être joignable par le biais des numéros UPT doit d'abord se faire enregistrer auprès de la base de donnée UPT, c'est à dire renseigner l'UPTSP². La façon d'enregistrement diffèrera sur base du type de terminal utilisé. Deux cas généraux sont donc possibles: Si le terminal est raccordé à un réseau public commuté ou si le terminal est connecté à un réseau IP. Mais fondamentalement on retrouve un nombre plus élevé de scénarios qui apparaissent en fonction de ce que le terminal s'enregistre auprès de son SP alors qu'il est localisé dans le domaine du SP ou qu'il se trouve dans un domaine qui est étranger à celui du SP. Les cas suivants sont donc possibles:

A partir d'un terminal de type SCN

- 1.1 attaché au réseau national de son UPTSP, via le GW de son UPTSP
- 1.2 attaché à un réseau étranger à celui de son UPTSP, via le GW de son UPTSP
- 1.3 attaché à un réseau étranger à celui de son UPTSP, via le GW de ce domaine étranger mais transféré vers son UPTSP
- 1.4 attaché à un réseau étranger à celui de son UPTSP, via le GW de ce domaine étranger

A partir d'un terminal de type IP

- 2.1 attaché au domaine de son UPTSP
- 2.2 attaché à un domaine étranger à son UPTSP

La phase d'enregistrement est précédée d'une phase d'identification. Celle-ci est basée sur la possession par l'utilisateur d'un code PIN.

Les points suivants traduisent de manière explicite quelques-uns des différents scénarios cités.

² SP:Service Provider

6.4.2 Scénarios à partir d'un terminal SCN vers son GW

Le but étant pour le terminal d'accéder au GW de son domaine qui lui fournit un service IVR (Interactive Voice Response). L'accès à ce GW se fait par l'intermédiaire d'un UPTAN (UPT Accès Number). L'UPTAN est un numéro de téléphone du réseau public associé au GW et devrait toujours être sans taxation pour l'appelant.

L'IVR est un système de communication tel qu'à partir d'un téléphone conventionnel à touches, on peut dialoguer avec un GW qui réagit en fonction des tonalités qui lui sont envoyées. Par exemple en phase d'identification, le GW demande à l'utilisateur, par transmission vocale, de s'identifier. Ce dernier, en introduisant son code d'identification sur le clavier envoie des impulsions vers le GW qui les interprète et réagit par transition dans d'autres états.

Voici quelques schémas de connexion du terminal vers le GW:

- A partir d'un numéro 0800 propre au domaine SCN du terminal qui est aussi celui du GW.

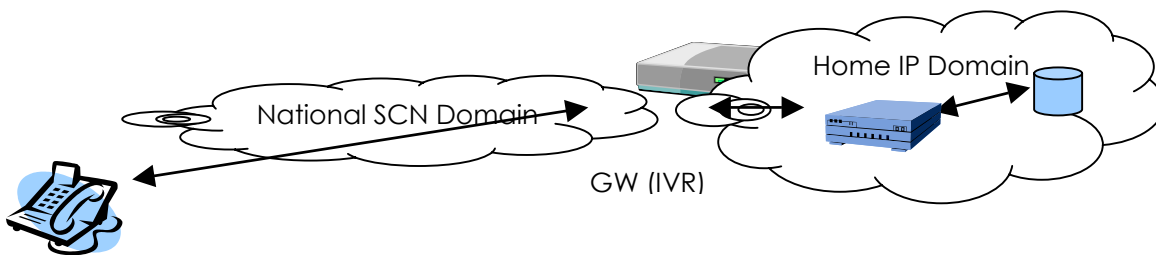


Figure 6.2
Cas 1

- A partir d'un numéro 0800 propre au domaine SCN du terminal mais formé à l'aide du préfixe international. Cette alternative engendre des taxations à charge du terminal appelant.

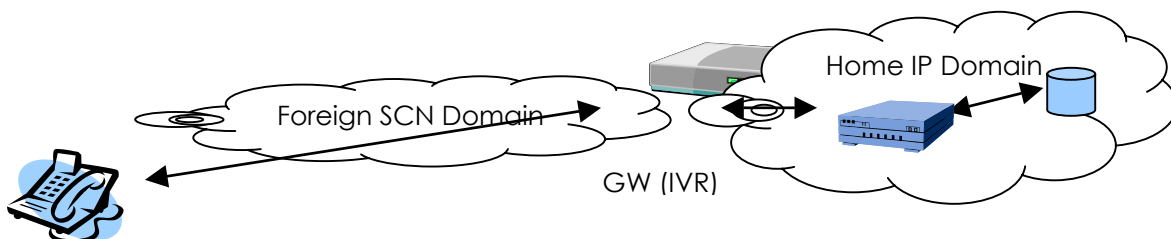


Figure 6.3
Cas 2

- A partir d'un numéro 0800 international propre au GW si le terminal ne se trouve pas dans le SCN de son GW. Cette alternative est meilleure que la précédente car ne demande pas de taxation

- A partir de l'UPTAC³ du GW, si le terminal est situé dans le domaine de son GW.

6.4.2.1 Terminal attaché au réseau public identique à son GW

L'utilisateur appelle son UPTSP par son UPTAN.

L'appel est routé vers le GW connecté au SCN et au domaine IP de l'UPTSP

Le GW répond par IVR et commence la phase d'authentification

Le numéro UPT et le code PIN sont envoyés vers le GK

Le GK envoie ces codes vers l'AD-BES⁴ pour vérifier l'identification

Si l'utilisateur est correctement identifié, un message est envoyé au GW spécifiant que l'identification est correcte et que l'utilisateur peut continuer son enregistrement. Dans ce cas, l'utilisateur précise s'il désire recevoir les appels UPT qui lui sont destinés et s'il désire pouvoir appeler d'autres abonnés par le biais des numéros UPT.

Les informations fournies par l'utilisateur sont ensuite envoyées vers l'AD-BES par le GK pour qu'elles puissent être conservées dans le profil de l'utilisateur.

6.4.2.2 Terminal attaché à un SCN différent de celui auquel est rattaché son UPTSP (version 1)

L'utilisateur appelle son UPTSP par son UPTAN international (sans taxation) ou par son numéro UPTAN précédé d'un indicatif international. Dans ce dernier cas, une taxation est opérée envers l'appelant.

Les démarches suivantes sont identiques au cas 6.4.2.1

6.4.2.3 Terminal attaché à un SCN différent de celui auquel est rattaché son UPTSP (version 2)

Cet enregistrement diffère des précédents dans le sens où l'utilisateur n'appelle pas directement son UPTSP par le biais de son GW mais appelle le GW du réseau public auquel il est connecté (ici différent du SCN auquel est lié son UPTSP).

La première étape à suivre par le GW contacté est donc de d'abord se renseigner si l'appelant fait partie de son domaine ou y est étranger; opération réalisée en se connectant à la base de données AD-BES. Si l'appelant n'en fait pas partie, une requête vers la base de données TRS est exécutée pour connaître l'adresse de l'UPTSP exacte de l'utilisateur. Celle-ci connue, la communication établie par l'utilisateur est redirigée sur le réseau IP vers le GW du domaine de l'utilisateur qui prend en charge la suite des opérations en IVR.

³ AC: Acces Code

⁴ AD-BES: Administrative Domain Back-End Service

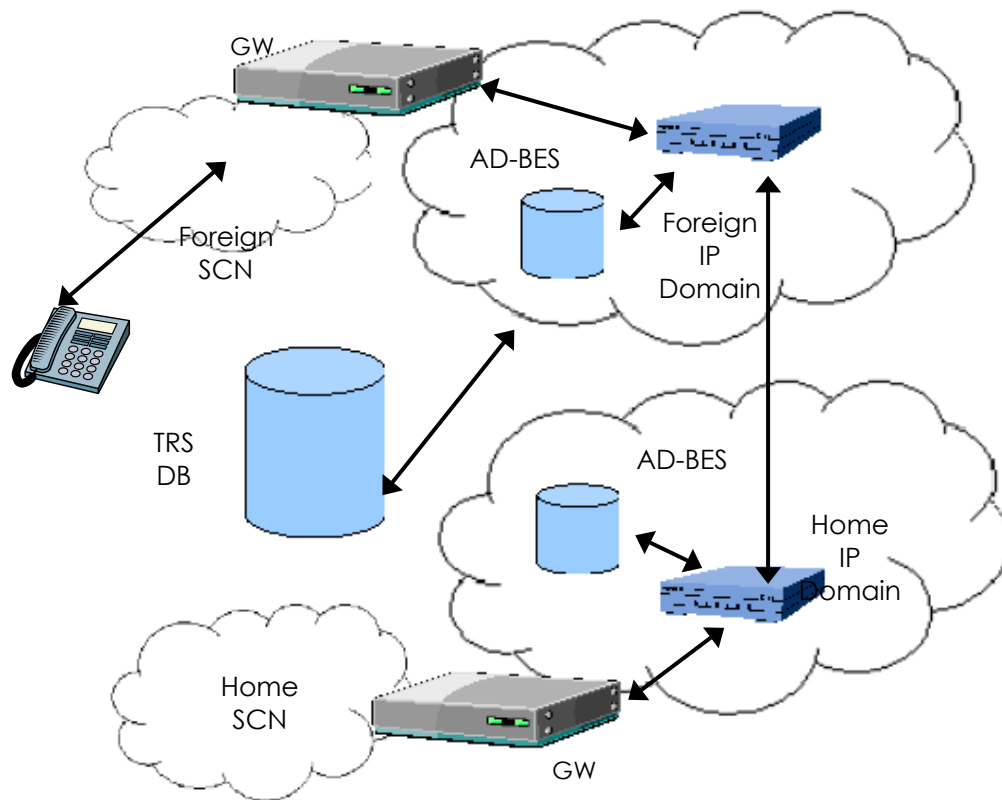


Figure 6.4

Terminal attaché à un SCN différent de celui auquel est rattaché son UPTSP

Les étapes suivantes sont suivies dans ce cas particulier

- Appel par l'utilisateur du numéro UPTAN du GW du domaine dans lequel il réside
- Première identification de l'utilisateur en IVR par le GW du domaine "étranger"
- Ces informations sont envoyées vers le GK du domaine IP "étranger"
- Le GK s'informe auprès de l'AD-BES pour savoir si l'appelant fait partie de ses propres utilisateurs
- Si l'utilisateur n'appartient pas au groupe géré par l'UPTSP, l'AD-BES émet une requête vers la partie temps réel de TRS pour obtenir l'adresse de l'UPTSP de cet utilisateur
- L'AD-BES renvoie ces informations au GK
- Le GK du domaine "étranger" route l'appel vers l'UPTSP auquel appartient l'utilisateur
- Le GK du domaine auquel appartient l'utilisateur route l'appel vers le GW de son domaine
- Le GW du domaine auquel appartient l'utilisateur prend en charge la seconde partie de l'enregistrement permettant ainsi à l'utilisateur d'accéder à son profile local.

Chapitre 7

Implémentation

7.1 PRÉ-REQUIS D'IMPLÉMENTATION

7.1.1 Les objets ASN.1

Il très fréquent de constater que les nombres sont représentés de bien des manières différentes entre les ordinateurs. Un entier peut par exemple être codé sur 16 bits ou 20 bits. Cette constatation n'est pas de goût à faciliter la programmation sur les diverses architectures. Pour gagner en compatibilité, lorsque l'on construit un logiciel pour plusieurs plates-formes distinctes, on passe par les méthodes d'abstraction de données.

Dans le domaine des télécommunications on établit également des règles d'ordre de passage des paramètres. Il est en effet essentiel de connaître le format des messages envoyés pour les décoder à la réception. Le protocole IP est d'ailleurs un exemple qui illustre bien ce problème. Il a été choisi de transmettre d'abord les bytes les plus significatifs. IP est surnommé un "Network Byte Ordering". De cette manière on établit un langage compréhensible entre les parties.

Dans d'autres cas de transmission de messages, il s'agit de transmettre des structures composées de plusieurs types complexes. Il est essentiel de se faciliter la tâche en inventant des règles de notation qui permettront de décrire ces structures complexes. Ces règles peuvent spécifier l'ordre dans lequel sont écrits les paramètres dans la structure, définir l'ensemble d'attributs obligatoires ou optionnels à intégrer dans la structure.

C'est ce qui est tenté d'être obtenu par les ASN.1. ASN.1 est l'acronyme de "Abstract Syntax Notation One". Il s'agit d'une spécification établie dans le but d'établir ces correspondances entre les formats des divers "types" disponibles sur les plates-formes informatiques. Cette abstraction débarrasse le programmeur de se renseigner sur la manière dont une structure est implémentée ou représentée.

Dans le cas d'OSI (Open System Interconnection), qui est une architecture standardisée internationale qui gouverne les interconnexions des ordinateurs entre les couches basses (physiques) et hautes (d'applications), des objets sont définis de manière abstraite dans les couches hautes pour être implémentées dans les couches basses. Par exemple un service d'une couche peut vouloir transférer certains objets abstraits entre des ordinateurs. Une couche inférieure peut fournir le service de transfert de zéros et de uns. Elle devra donc convertir l'objet à transférer en zéros et uns. OSI est

donc un protocole ouvert car il supporte plusieurs implémentations différentes de services à chaque couche.

OSI utilise les spécifications ASN.1 de déclaration des objets abstraits. Pour effectuer une petite comparaison, les notations ASN.1 sont comme le langage d'une grammaire BNF. Il existe des symboles terminaux et non-terminaux et un symbole de départ.

ASN.1, utilisé dans le logiciel OpenGatekeeper, décrit un ensemble de types de données des plus simples comme les entiers, aux plus compliquées comme les structures. Le tableau 7.1 reprend la syntaxe utilisée pour décrire la syntaxe ASN.1.

::=	"est définit comme"
SEQUENCE	Séquence d'éléments ASN.
Xxxxxxxxxx	Ce mot décrit un champ dans un paquet et est définit ailleurs dans le code par une autre structure ASN
xxxxxxxxxxxx	Description "User Friendly" du mot avec majuscule associé

Table 7.1
Syntaxe ASN.1

Dans ASN.1 un type est un ensemble de valeurs. Pour certains types il y a une infinité de valeurs, pour d'autres il y en a un nombre fini. Une valeur d'un type ASN.1 est un élément de l'ensemble du type.

ASN.1 possède quatre sortes de types:

- les simples (comme les entiers)
- les structures (qui ont des composants)
- les Tags (qui sont dérivés d'autres types)
- les autres types comme "CHOICE" et "ANY".

On assigne un type à un nom à l'aide de " ::= ". Ces noms peuvent eux-mêmes servir à la définition d'autres types.

Tous les types (sauf CHOICE et ANY) possèdent un TAG qui consiste en une classe et un nombre tag positif. Deux types sont identiques si leur numéro de tag est identique. En d'autres mots le nom du type importe peu pour le différencier d'un autre. Seul le "tag number" intervient.

Il y a quatre types de tags:

- Universel (pour les types dont le sens est le même dans toutes applications)
- Application (pour les types dont le sens est propre à l'application courante)
- Privés (pour les types dont le sens est propre à une entreprise)
- Context-Specific (dont le sens varie en fonction du contexte)

Je n'entrerai pas plus dans les détails du formalisme de l'ASN.1. Je renvoie aux références pour de plus amples explications. Il est bien de s'y référer pour comprendre le programme OpenGatekeeper.

7.1.2 Fonctionnement interne d'OpenGatekeeper

Le logiciel OpenGatekeeper est basé sur l'utilisation d'objets au format ASN.1 "Abstract Syntax Notation". Il est bien de d'abord se familiariser avec ces structures avant de commencer l'analyse complexe du code. Ces structures sont mises en place en parallèle avec des méthodes abstraites pour faciliter l'automatisation de tâches qui seraient, sans quoi, inévitablement plus compliquées à traiter. Opengatekeeper est un logiciel qui est écrit en C++ basé sur les travaux de OpenH323 et qui est soumis à licence Open Source.

On rencontre également, dans le logiciel, de nombreux appels vers des fonctions de traitement de signalisation H.225, H245,... contenues dans la librairie openH323. De même il y a de nombreux appels vers la librairie PWLib.

OpenGatekeeper est constitué en lui-même par une trentaine de classes C++ mais au total sa compilation et son fonctionnement demandent un millier de classes provenant des librairies précitées. Pour donner une estimation, environ 350 classes sont nécessaires au protocole H.245 et 160 classes au protocole de signalisation H.225.

OpenGatekeeper ne pourra pas être complètement analysé dans ce travail. Je porterai l'accent sur les points qui m'ont été nécessaires dans les perspectives d'aboutissement du projet de connexion à la base de données TRS. Nous aborderons d'abord une discussion à propos des structures importantes contenues dans le logiciel, ensuite nous poursuivrons par une description plus complète des signaux H.225 et des structures auxquels ils font référence.

7.1.3 Structures du logiciel OpenGatekeeper

On l'aura compris, OpenGatekeeper est construit à partir de nombreuses structures faisant appel à l'abstraction de données. Une grande partie du logiciel est basée sur l'emploi de tableaux au format ASN. Il est donc utile pour la bonne compréhension du programme de s'attarder sur ceux-ci. Après avoir vu le fonctionnement des tableaux, nous aborderons une discussion à propos des objets importants que ces tableaux contiendront.

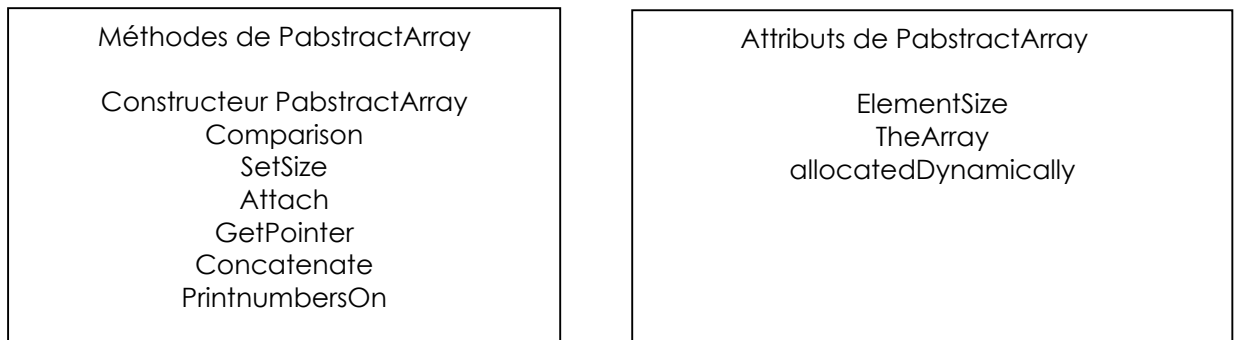
Le fichier *array.h* définit les classes des tableaux dynamiques utilisés dans le programme.

7.1.3.1. Classe PabstractArray

Cette classe contient la définition d'un tableau de longueur variable d'éléments de taille mémoire quelconque. Ces éléments peuvent être des bytes ou mêmes des structures complexes. La seule condition est qu'ils ne peuvent contenir d'objets qui requièrent une construction ou une destruction.

La classe est construite de manière à obtenir une abstraction telle que l'on ne doive pas se soucier du format des objets introduits dans le tableau.

De plus la classe gère l'insertion et la suppression d'éléments en libérant ou allouant suffisamment de mémoire.

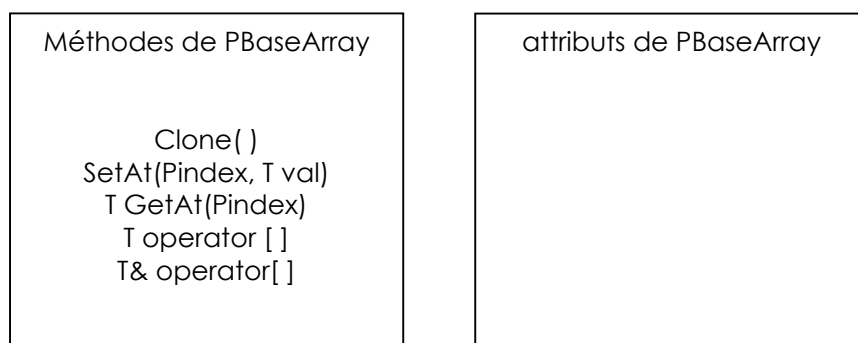


7.1.3.2 Classe Template PBaseArray

Cette classe Template est dérivée de PabstractArray. Elle associe à un tableau PabstractArray des objets d'un type spécifique <T>.

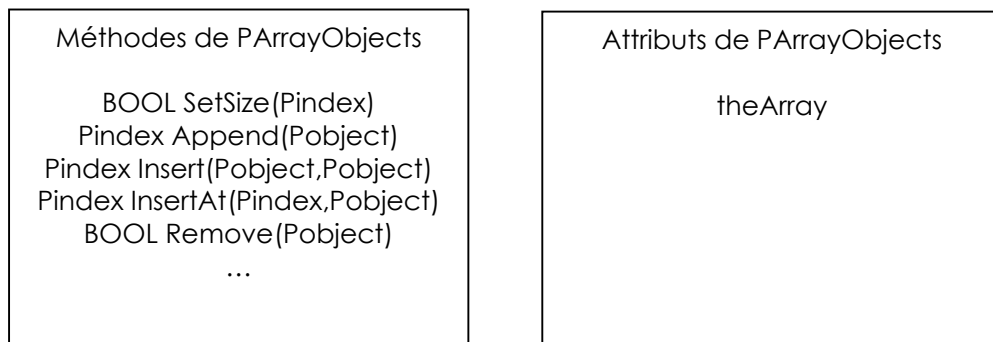
Dans ce template de classe est défini le surchargement de l'opérateur [] qui renvoie l'élément du tableau situé à la place [i] par la méthode GetAt(). Il est utile de constater cette observation sans quoi on peut se poser des questions quant à la compréhension du logiciel.

On retrouve également dans array.h la définition de PBASEARRAY qui est un simple: `#define PBASEARRAY (cls,T) typedef PbaseArray<T> cls`



7.1.3.3 Classe PArrayObjects

Cette classe est dérivée de PCollection. Elle est utilisée pour définir les objets PASN_ObjectArray grâce au template de classe PArray.



7.2 REQUÊTES D'ENREGISTREMENT H.225

7.2.1 Rappel du principe

Le protocole H.225 définit les procédures d'enregistrement et d'autorisation de connexion de points d'accès à des objets du protocole H.323. Un client désirant établir une connexion avec un interlocuteur peut par exemple devoir d'abord se faire enregistrer auprès de son gatekeeper. Cette démarche est constituée par un ensemble de procédures administratives appelées requêtes. Dans la figure suivante est représenté l'organigramme du processus d'enregistrement d'un client auprès d'un gatekeeper. Sur le même diagramme on retrouve également le processus de déconnexion du client.

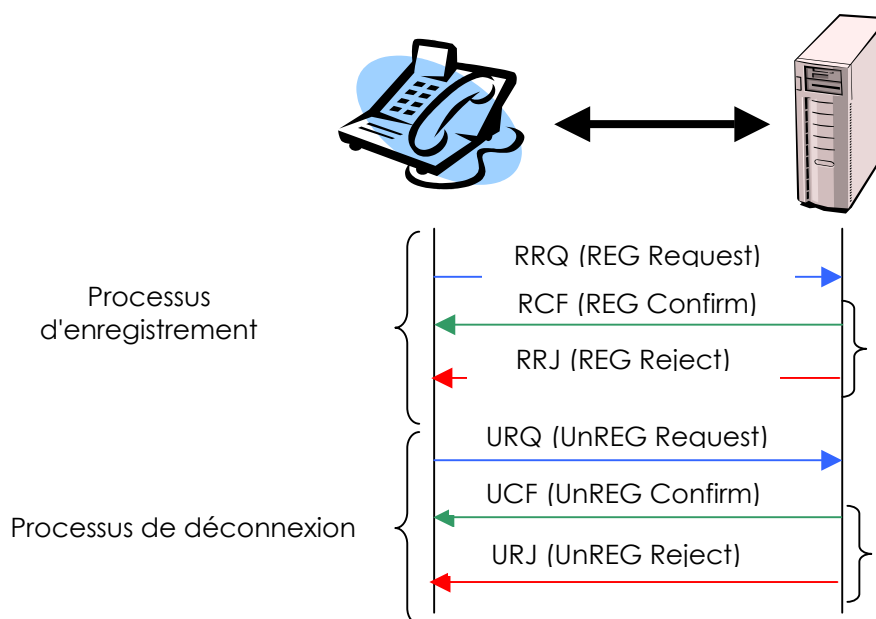


Figure 7.1
Requêtes H.225

Le client, désirant se faire enregistrer auprès du gatekeeper, envoie une première requête RRQ contenant des informations précises sur son

identification. Elle transmet une table d'alias correspondant au "surnom" du client ainsi que l'adresse de transport associée au client.

Le GK s'assure qu'aucun alias portant le même nom ne se trouve déjà enregistré dans sa mémoire. S'il existe déjà une entrée correspondant à cet alias, le GK vérifie si cette entrée est associée à la même adresse de transport que le client désirant se faire enregistrer. Si ce n'est pas le cas, le GK renvoie dans un message H.225 un "Registration Reject" et y spécifie la raison du refus. Si par contre l'adresse de transport est la même que celle du client désirant se faire enregistrer, ou si l'alias n'existe pas encore dans la mémoire du GK, celui-ci renvoie un "Registration Confirm" et enregistre les données propres au client.

7.2.2 Messages H.225

La signalisation H.225 permet, on vient de le voir, le transfert de messages RAS. Les messages H.225 sont réceptionnés dans le gatekeeper par un serveur de messages RAS. Ce serveur effectue des actions différentes en fonction du type de message H.225 reçus. Nous analyserons dans ce serveur quelques requêtes intéressantes comme les requêtes RRQ et ARQ, cela nous permettra de bien comprendre le mécanisme d'abstraction de données. De plus ces requêtes sont utiles pour donner suffisamment d'informations au moment où nous programmerons le gatekeeper.

7.2.2.1 Requête RRQ

La fonction du serveur RAS traitant les messages RRQ reçoit les paramètres suivants:

```
const H225_RegistrationRequest &      RegReq
      H225_RasMessage &                Reply
      EndpointTable *                  ETable
Const H225_GatekeeperIdentifier &      Gkid
      H225_TransportAddress &         ReplyTo
```

Nous analyserons uniquement le paramètre RegReq. Nous verrons que c'est lui qui détient les informations concernant le client à enregistrer auprès de la base de données du gatekeeper. Les autres paramètres contiennent les informations du gatekeeper, comme par exemple ETable qui détient les informations sur les clients déjà enregistrés auprès du gatekeeper.

RegReq est un objet du type H225_RegistrationRequest qui possède un attribut m_callSignalAddress. Cet attribut est un tableau contenant des informations sur l'adresse du client se faisant enregistrer. Le premier élément de ce tableau nous fournira l'adresse H225 de transport assignée au client. Par un surchargement d'opérateur nous retirons l'adresse de ce client sous forme d'une adresse IP.

```
((H225_TransportAddress_ipAddress&)RegReq.m_callSignalAddress[0]).m_ip
```

Ce résultat sera utilisé pour afficher, par le gatekeeper, l'adresse IP du client désirant se faire enregistrer.

La figure 7.2 explicite la hiérarchie de la classe H225_RegistrationRequest et la démarche à suivre pour extraire à partir d'une telle structure l'adresse IP qu'elle contient. On remarquera qu'on se retrouve vite imbriqué dans l'ensemble des autres structures du programme et que cette opération qui peut paraître banale requiert une connaissance approfondie de l'ensemble des classes qu'elle fait intervenir. La figure 7.4 donne une légende pour comprendre les figures qui suivront.

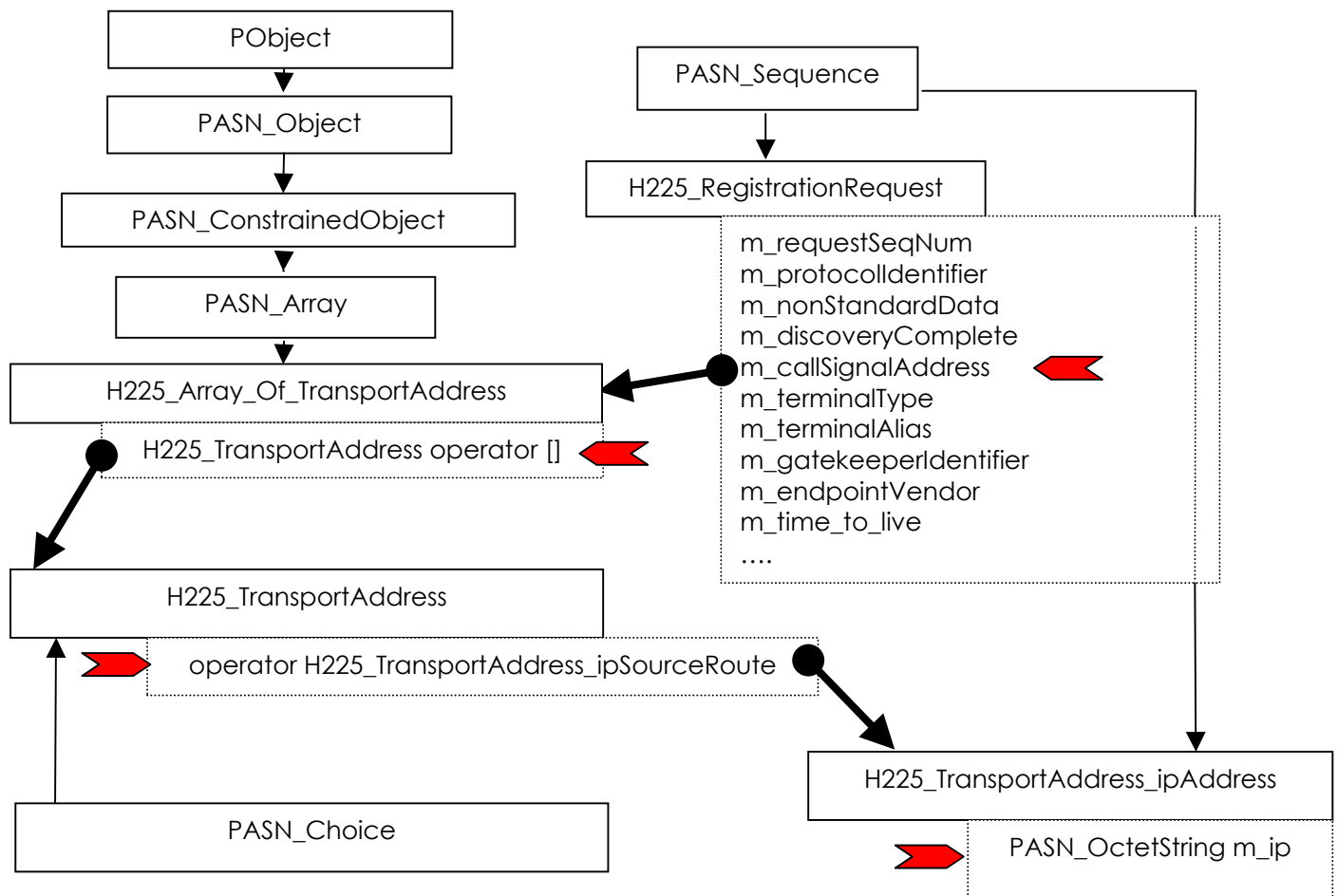


Figure 7.2

Récupération de l'adresse IP du client

7.2.2.2 Requête ARQ

La fonction du serveur RAS traitant les messages ARQ reçoit les paramètres suivants:

```

const H225_RegistrationRequest &    AdmissReq
      H225_RasMessage &             Reply
      EndpointTable *                Eptable
      CallTable *                    CTable
const H225_GatekeeperIdentifier &    Gkid
      H225_TransportAddress &        ReplyTo
const H225_TransportAddress &        MyCallAddr
      bool                           GKRouter
const Environ &                      Environ
  
```

Nous analyserons uniquement le paramètre AdmissReq. Nous verrons que c'est lui qui détient les informations concernant le client à contacter. Les autres paramètres contiennent, comme pour la requête RRQ, des informations concernant le gatekeeper, comme par exemple Eptable qui détient les informations sur les clients déjà enregistrés auprès du GK... La figure 7.3 montre le chemin à suivre pour extraire l'alias du client appelé par le client appelant.

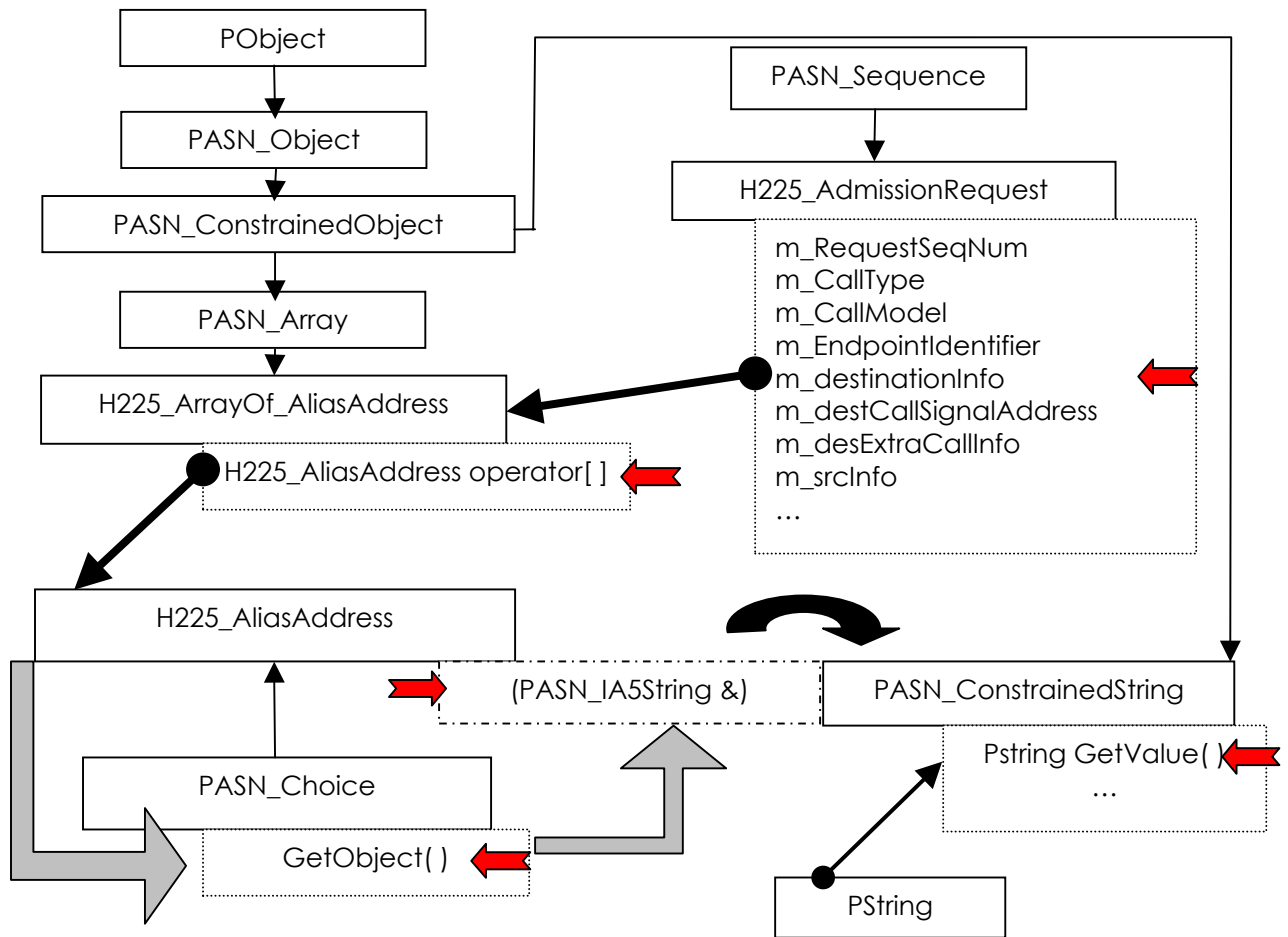


Figure 7.3
Récupération de l'adresse à contacter

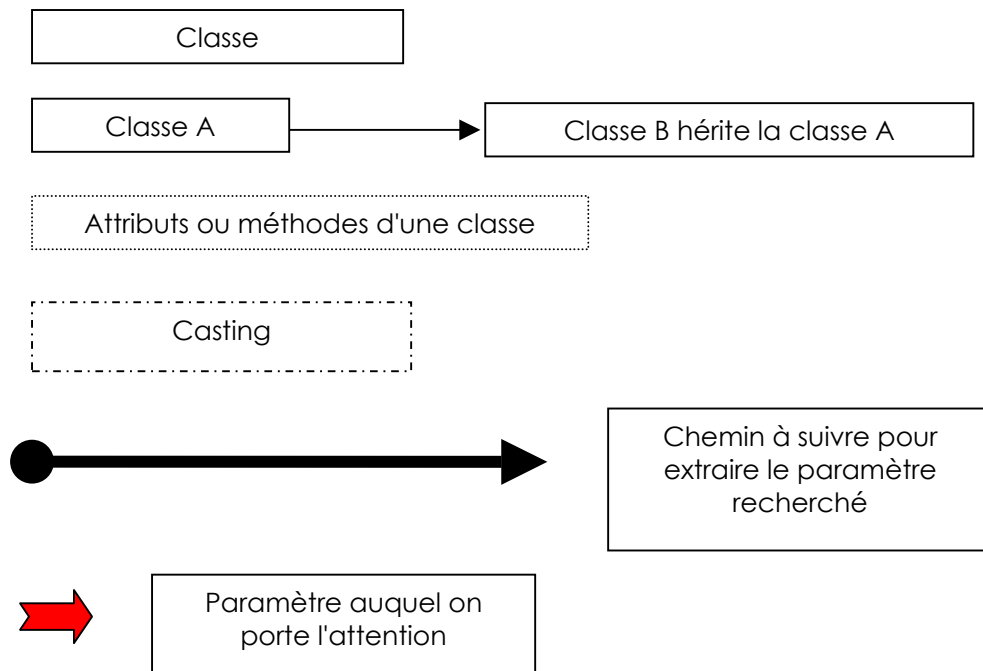


Figure 7.4
Légende

7.3 OBJECTIF ET MÉTHODE D'IMPLEMENTATION

7.3.1 Objectif

L'objectif désiré dans cette implémentation est de permettre la prise en considération des services de mobilité, définis par TRS, par le logiciel OpenGatekeeper. Pour atteindre cet objectif, nous avons besoin d'une infrastructure client-serveur, permettant de tester le gatekeeper, ainsi que d'un environnement de développement pour le programmer.

Dans ce travail nous abordons uniquement l'aspect Voice Over IP entre deux clients se trouvant sur un réseau LAN. Aucune interconnexion vers le monde PSTN ne sera établie.

Le client est modélisé par le logiciel NetMeeting de Microsoft. Il ne sera apporté aucune modification à ce logiciel. Le gatekeeper est modélisé par le logiciel OpenGatekeeper.

On configure également le client Netmeeting pour se trouver dans le cas du modèle à "routage par gatekeeper". Il s'agit du modèle où le client accède aux services du gatekeeper comme par exemple la résolution d'adresse et le routage des appels. L'utilisation de RAS est nécessaire dans cette perspective.

Des modifications sont apportées au gatekeeper de manière à ce que lors d'une requête de connexion auprès d'un numéro UPT, le gatekeeper se connecte à la base de donnée TRS pour récupérer des informations à propos du numéro UPT concerné. Ces informations sont ensuite mises en forme pour être stockées dans les tables du gatekeeper.

Pour l'instant le GK ne fait aucune différence entre un numéro UPT et un numéro quelconque, un client Netmeeting pouvant fort bien se faire enregistrer auprès d'un GK sous un numéro 878-878-XXXXX sans pour autant être qualifié de vrai numéro UPT.

Dans le cadre de ce travail, on considère qu'aucun client Netmeeting ne peut se faire enregistrer auprès d'un GK sous forme d'un numéro UPT.

7.3.2 Méthode d'implémentation

L'implémentation traitée ici est en rapport avec les modifications apportées au gatekeeper. Vu qu'il n'y a aucune implémentation à apporter au code du client NetMeeting, nous nous contentons de le configurer correctement suivant ce qui a été annoncé au point précédent.

L'implémentation réalisée dans ce travail s'est poursuivie sur un rythme lent, car avant d'arriver à situer les lieux où il y a des modifications à apporter, il m'a fallu comprendre la structure générale du programme. De plus, le code du gatekeeper est, je l'ai déjà écrit, assez complexe et fortement optimisé. Il y

a donc eu un pré-travail de découverte du code et de documentation à son sujet.

Le code du gatekeeper a d'abord été compilé pour qu'il puisse démarrer à partir d'une station fonctionnant sous un système d'exploitation Solaris. Malheureusement, et à contre cœur, j'ai dû abandonner l'idée de continuer sous ce système d'exploitation. Cette raison est motivée par le fait que jamais je ne suis arrivé à compiler le code sous ce système d'exploitation. En effet, les bibliothèques fournies avec le code ne le permettaient pas. Je n'ai pas recherché à savoir si actuellement ce problème a été résolu.

Par après j'ai réussi à compiler le code du gatekeeper pour qu'il fonctionne sous un système d'exploitation Linux. Mais là également je n'ai pas pu continuer à progresser. En cause le problème de l'inexistence d'un environnement de développement suffisamment capable de traiter un code C++ élaboré.

Au bout du compte, j'ai poursuivi le traitement du code sous l'environnement Visual C++ tournant sous le système d'exploitation Windows NT. Il est bien de faire remarquer que la compilation du code nécessite des ressources CPU et de mémoire assez importantes sur la machine de travail. Dans cette optique, je signale que disposer de 128 Mb de RAM n'est pas un luxe mais une nécessité et à proprement parler se révèle parfois insuffisant si l'on travaille simultanément avec d'autres programmes.

Avant d'implémenter, il faut rechercher dans le code existant l'endroit où insérer la partie qui se connectera à la base de données TRS et savoir dans quelles conditions s'y connecter. On aura compris que c'est dans le serveur RAS du gatekeeper que l'on devra rechercher un endroit adéquat. Nous verrons exactement où placer l'appel vers la base de données TRS dans le point suivant.

7.3.3 Serveur RAS et connexion à la DB TRS

Les fonctionnalités du Serveur RAS sont séparées en deux parties distinctes dans le logiciel OpenGatekeeper: La gestion des requêtes arrivant vers le serveur RAS et leur répartition vers leur méthode de traitement. Ces deux fonctionnalités sont traitées par des classes différentes. Il s'agit pour la première de la classe RasThread et pour la seconde de la classe RasServer comme le présente la figure suivante.

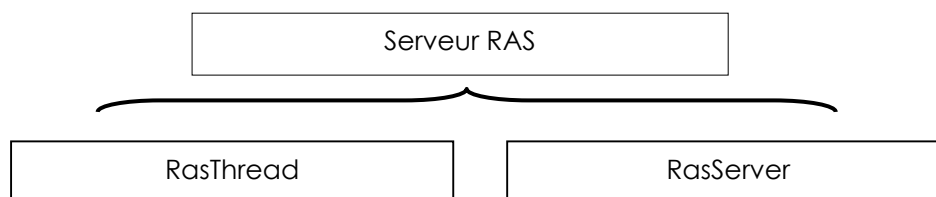


Figure 7.5
Schéma du serveur RAS

La classe RasThread est la première des classes du serveur RAS à être instanciée pour permettre à plusieurs requêtes émises par des clients différents d'être traitées simultanément. Il n'est en effet pas conseillé de prolonger les délais d'attente entre la requête et sa réponse en plaçant les requêtes dans des queues. D'où la nécessité du multi-threading.

Elle se compose de deux méthodes principales: ReadRasReq et WriteRasReq qui collectent et émettent respectivement les requêtes et les messages issus et destinés aux clients.

Lorsqu'un thread reçoit une requête RAS il la remet au RasServer qui s'occupera de son traitement. RasServer traite la requête RAS en fonction de son type. Dans notre cas nous ne nous occuperons pas de l'ensemble des requêtes RAS possibles, mais uniquement de deux en particulier. Le tableau suivant représente néanmoins l'ensemble des types de requêtes RAS possibles. Dans ce tableau on retrouvera en caractères gras les requêtes RAS qui sont approchées dans le cadre du travail. Les requêtes RAS sont traitées dans le RasServer par la méthode HandleRasRequest.

registrationRequest	unregistrationRequest	admissionRequest
bandwidthRequest	disengageRequest	locationRequest
gatekeeperRequest	infoRequestResponse	unregistrationConfirm
unregistrationReject	gatekeeperConfirm	gatekeeperReject
registrationConfirm	registrationReject	admissionConfirm
admissionReject	bandwidthConfirm	bandwidthReject
disengageConfirm	disengageReject	locationConfirm
locationReject	infoRequest	nonStandardMessage
unknownMessageResponse	requestInProgress	resourcesAvailableIndicate
resourcesAvailableConfirm	infoRequestAck	infoRequestNak

Les requêtes à considérer figurent déjà dans les points 5.2 et 5.3 où on les retrouve sur les figures capturées à partir du moniteur réseau.

La requête RRQ est analysée dans l'optique de savoir avec quel client on traite pour les requêtes ARQ ultérieures. A partir de cette requête on affiche dans l'environnement graphique du gatekeeper l'adresse IP du client effectuant la requête RRQ par la méthode indiquée au point 7.2.2.1.

Mais c'est la requête ARQ qui est pour nous la plus importante à traiter puisque c'est elle qui fournit en paramètre l'alias ou l'adresse du correspondant que le client effectuant la requête ARQ désire contacter.

Par la requête ARQ on récupère donc l'alias du client à contacter. C'est à cet endroit qu'il faudra placer notre appel vers la base de données TRS. Il faut d'abord vérifier si le client émet une requête de connexion auprès d'un numéro au format UPT. Si c'est le cas, on se connectera à la base de données TRS et on y récoltera les informations concernant l'abonné demandé. Si ce n'est pas le cas, on continue dans le programme comme si la requête était une requête standard.

Concernant la connexion éventuelle vers la base de donnée TRS, il faut introduire dans le code une fonction qui, sur base du numéro UPT, aille rechercher les informations voulues. Le site WEB <http://www.resolution-service.org> propose le téléchargement d'une API C qui doit en principe, pour un numéro donné, récupérer les informations de localisation universelle.

Nous aurions pu placer l'appel vers la base de donnée TRS à un autre endroit du logiciel OpenGatekeeper. Cet endroit est situé au moment de rechercher des informations concernant l'appelé dans la liste des clients déjà enregistrés auprès du gatekeeper. Cet endroit est repéré par la méthode EPTable->FindByAlias dans la méthode OnARQ située dans le fichier RasServer.cxx (EPTable est le tableau dynamique qui contient l'ensemble des clients déjà enregistrés dans le gatekeeper). Nous aurions dû alors retravailler la méthode FindByAlias en envisageant le cas où l'alias introduit est un numéro UPT.

Cette idée serait certainement moins adéquate car on lancerait la méthode FindByAlias. Or cette méthode se base sur la recherche d'alias à proprement parler dans la EPTable et non pas d'identificateurs UPT. Donc j'ai gardé l'idée d'effectuer la vérification au début de la requête ARQ.

En finale, une fois que les requêtes sont finalisées d'être traitées par RasServer, la main est rendue au RasThread qui rend la réponse de la requête au client concerné. On peut d'ailleurs schématiser par un petit diagramme l'ordre du déroulement des requêtes:

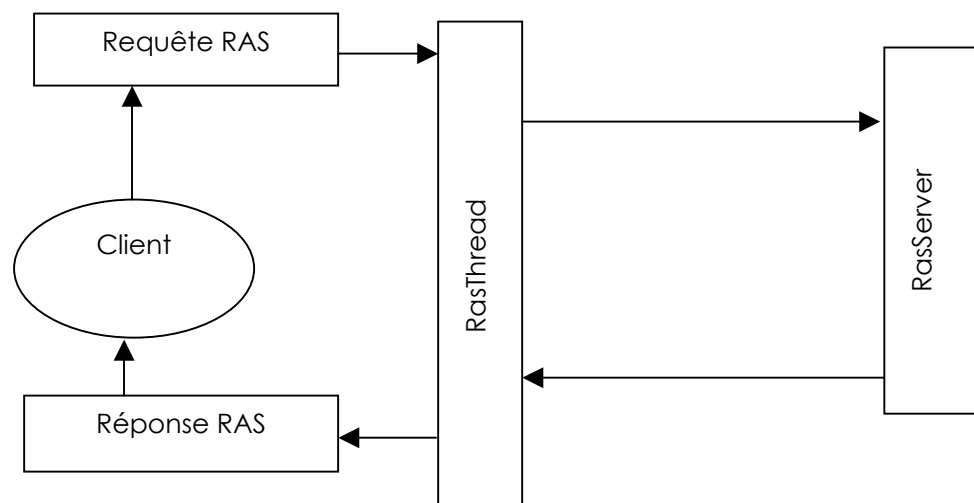


figure 7.6
Diagramme d'établissement des requêtes

7.3.4 ResolutionService API

L'API disponible sur Internet est écrite dans le langage C++. Comme le code du gatekeeper est également en C++, on n'a pas de problème de conversion de langage. L'API de ResolutionService permet d'effectuer différentes actions en rapport avec la base de données TRS. Malheureusement, ces actions ne sont pas documentées et seul leur nom peut apporter quelques éclaircissements sur leur objectif. Je ne ferai que citer dans le tableau 7.2 les actions. J'accorderai par après plus d'importance à l'action "Get Translation". Il s'agit de l'action qui donne des informations à propos des numéros UPT.

Provider Registration	Provider View
Provider Update	New Subscriber Registration
New Subscriber Registration with UPT	Subscriber view by Name
Subscriber view by UPT	Subscriber Update
Subscriber's Provider Change	Get Translation

Table 7.2
Actions envisagées par l'API

"Get Translation" retourne un identificateur de contact après avoir introduit un numéro UPT. Malheureusement je n'ai pu obtenir (pour l'instant) de plus amples renseignements concernant l'identificateur fourni en réponse à la requête. Par exemple pour le numéro UPT de test⁵ introduit dans la requête on obtient l'identificateur de contact "00004". Or cet identificateur ne correspond à aucune adresse IP. Voici l'entête de la méthode GetTranslation:

```
int    GetTranslation
(
    const char *UPT,
    char *ContactID,
    Cause_e& cause,
    char *errMsg=NULL
);
```

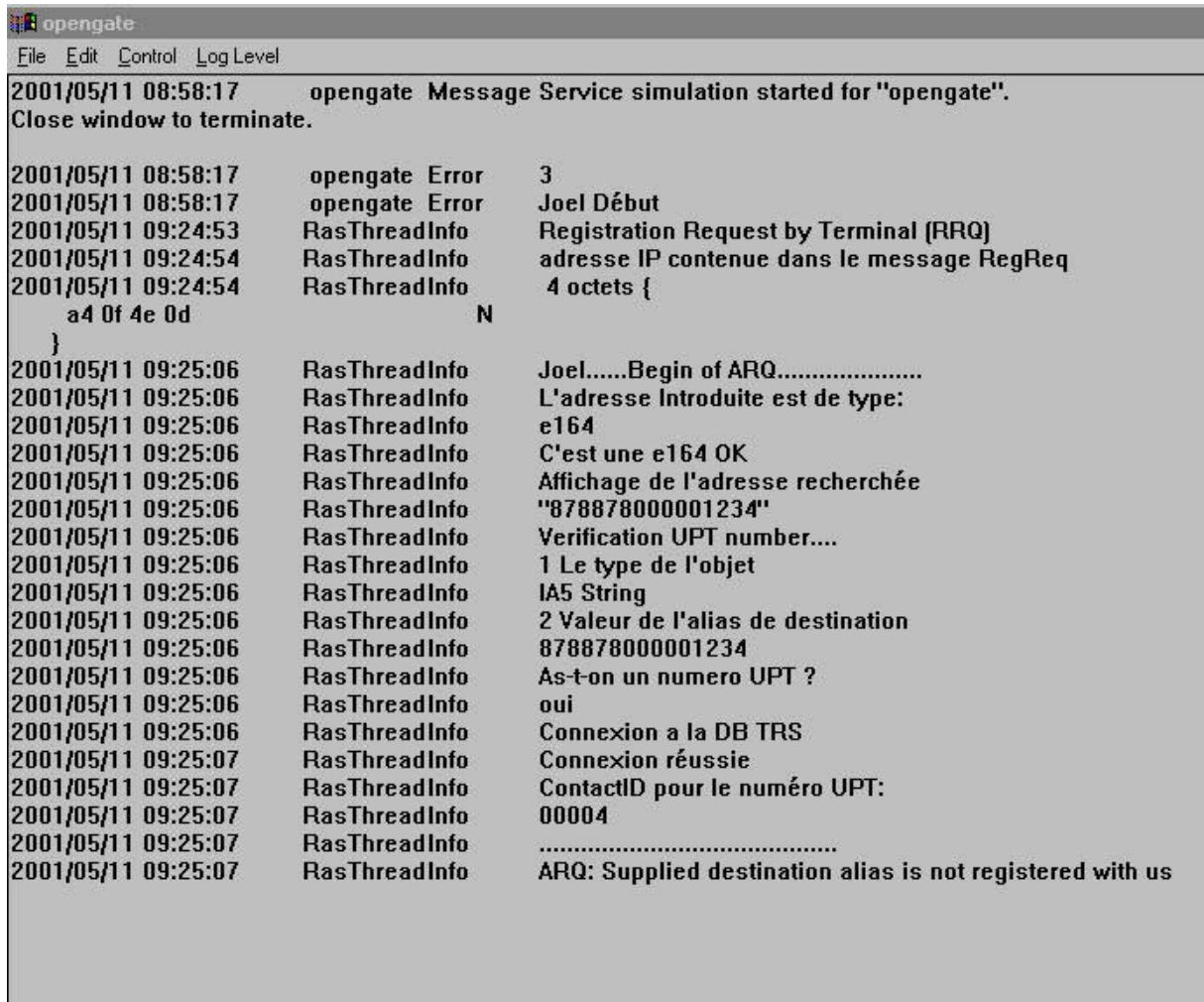
Pour l'instant j'ai modifié le code du gatekeeper pour qu'il récupère cette information. Il faudra traiter cette donnée peut-être dans le cadre d'un autre travail.

Le numéro d'identification fourni par cette méthode peut également être accessible par le site web <http://www.resolution-service.org/>. Sur celui-ci on peut introduire les informations concernant un numéro UPT et en récupérer l'identificateur.

⁵ Le numéro UPT de test est 878878000001234

7.4 VISUALISATION

Cette petite partie montre par une figure l'aspect de l'environnement graphique qu'offre le serveur OpenGatekeeper lors des requêtes RRQ et ARQ. Les messages affichés sont suffisamment explicites pour comprendre. Il s'agit des réponses produites par le gatekeeper lors d'une requête de connexion auprès d'un numéro UPT par un client Netmeeting. Dans un premier temps, on affiche l'adresse IP du client, ensuite on affiche l'alias du destinataire à contacter et s'il s'agit d'un numéro UPT on effectue la requête vers la base de données TRS.



```
opengate
File Edit Control Log Level
2001/05/11 08:58:17 opengate Message Service simulation started for "opengate".
Close window to terminate.

2001/05/11 08:58:17 opengate Error 3
2001/05/11 08:58:17 opengate Error Joel Début
2001/05/11 09:24:53 RasThreadInfo Registration Request by Terminal (RRQ)
2001/05/11 09:24:54 RasThreadInfo adresse IP contenue dans le message RegReq
2001/05/11 09:24:54 RasThreadInfo 4 octets {
a4 0f 4e 0d N
}
2001/05/11 09:25:06 RasThreadInfo Joel.....Begin of ARQ.....
2001/05/11 09:25:06 RasThreadInfo L'adresse Introduite est de type:
2001/05/11 09:25:06 RasThreadInfo e164
2001/05/11 09:25:06 RasThreadInfo C'est une e164 OK
2001/05/11 09:25:06 RasThreadInfo Affichage de l'adresse recherchée
2001/05/11 09:25:06 RasThreadInfo "878878000001234"
2001/05/11 09:25:06 RasThreadInfo Verification UPT number....
2001/05/11 09:25:06 RasThreadInfo 1 Le type de l'objet
2001/05/11 09:25:06 RasThreadInfo IA5 String
2001/05/11 09:25:06 RasThreadInfo 2 Valeur de l'alias de destination
2001/05/11 09:25:06 RasThreadInfo 878878000001234
2001/05/11 09:25:06 RasThreadInfo As-t-on un numero UPT ?
2001/05/11 09:25:06 RasThreadInfo oui
2001/05/11 09:25:06 RasThreadInfo Connexion a la DB TRS
2001/05/11 09:25:07 RasThreadInfo Connexion réussie
2001/05/11 09:25:07 RasThreadInfo ContactID pour le numéro UPT:
2001/05/11 09:25:07 RasThreadInfo 00004
2001/05/11 09:25:07 RasThreadInfo .....
2001/05/11 09:25:07 RasThreadInfo ARQ: Supplied destination alias is not registered with us
```

Figure 7.7
Visualisation de l'interface de OpenGatekeeper

Conclusion

Les projets de normalisation des protocoles de signalisation relatifs à la téléphonie sur Internet suivent leur chemin. Il faut leur laisser du temps d'aboutir pour que toute la logique interne à cette technologie se mette en place et ainsi de pouvoir convenir d'un standard commun et efficace.

On constate qu'il ne faut pas nécessairement inventer un protocole de signalisation en se basant sur les techniques de signalisation existantes et que d'ailleurs une approche verticale peut compliquer étonnamment la complexité de cette signalisation, comme on le constate avec H.323. Il faut malgré tout remarquer que H.323 est le protocole le plus complet à ce moment mais également le plus fastidieux à mettre en place en comparaison avec le protocole SIP. SIP ouvre la voie à de nouvelles extensions et autorise des mises à jour plus simples que pour les autres protocoles.

De nombreux services téléphoniques comparables à ceux qu'on retrouve en téléphonie standard sont également disponibles en téléphonie sur Internet: déviations d'appels ou encore messageries vocales... Certains d'entre eux sont encore en phase de développement mais leur avenir est assuré.

En ajoutant à ces services les perspectives de mobilité personnelle, on atteint des objectifs pratiques pour les personnes à grande mobilité, désireuses d'être joignables facilement.

Ce travail a montré quelques-unes des logiques propres aux protocoles de signalisation téléphonique pour ensuite aboutir sur l'intégration, dans un logiciel, d'une partie de service de mobilité personnelle.

Le service de mobilité personnelle n'est pas encore disponible pour le grand public, il faudra encore attendre un certain temps avant que celui-ci ne voie le jour, car il est nécessaire de créer les organismes qui s'occuperont de sa gestion et de plus car les services de mobilité personnelle ne sont pas encore complètement normalisés. En effet, des autres services de mobilité personnelle que ceux proposés par TTT-Services, existent. Il faudra donc peut-être opérer un choix pour choisir le plus efficace.

La téléphonie sur Internet, malgré qu'elle ne soit plus si récente, laisse donc encore de nombreuses portes ouvertes pour des développements ultérieurs. Nous nous sommes concentrés dans ce travail sur les aspects logiciels de ces développements mais il faut bien évidemment également tenir compte des nouvelles techniques à inventer concernant son aspect matériel. Je pense ici aux optimisations des appareils de routage.

En ce qui concerne les méthodes d'adressage, nous arriverons peut être à court d'adresses IP disponibles si nous conservons le protocole IPV4. Il y aura encore certainement du travail à consacrer concernant la téléphonie sur Internet disposant cette fois du protocole IPV6.

Glossaire

PABX (Private Automated Branch Exchange)

Système de commutation automatique utilisé pour desservir un système téléphonique privé tel une entreprise. Il en existe de deux types : analogues et digitaux. Les modèles digitaux peuvent commuter des données et des circuits "voix".

PBX (Private Automatic Telex Exchange)

Commutateur téléphonique fournissant des connexions vocales à l'intérieur d'une entreprise et permettant également aux utilisateurs d'accéder à des commutateurs publics ou des réseaux privés à l'extérieur de l'entreprise.

Protocole

Ensemble de règles acceptées volontairement par des vendeurs commerciaux et des utilisateurs pour s'assurer que les équipements de réception et de transmission se comprennent. En général un protocole couvre trois grands domaines: La méthode en laquelle les données sont représentées, la détection des erreurs et des échecs et l'action de correction active. Les terminaux exécutant les mêmes fonctionnalités, mais sous des protocoles différents, ne peuvent être utilisés sur le même réseau sans des convertisseurs ou des émulateurs de protocole. Il existe trois catégories de protocoles: Les bits, byte et les caractères.

Standard

Un document qui recommande un protocole, une interface, un type de câblage ou un autre aspect du réseau. Il peut même recommander quelque chose comme un modèle conceptuel (par exemple une architecture de communication) Ils sont développés par des organismes internationaux ou nationaux reconnus comme standards.

Switch

Périphérique qui établit, coupe ou change les connexions dans un circuit pour déplacer (shift) vers un autre circuit au moyen d'un commutateur. Dans le monde des télécommunications, il s'agit d'un synonyme d'un PABX ou d'un CO (Central Office) En réseau de commutation par paquet, il s'agit du périphérique qui redirige les paquets.

PSTN (Public Switched Telephone Network)

Il s'agit de l'ensemble complet d'un système téléphonique incluant les téléphones, les lignes locales et les "trunk lines" et les "exchanges".

Annexes

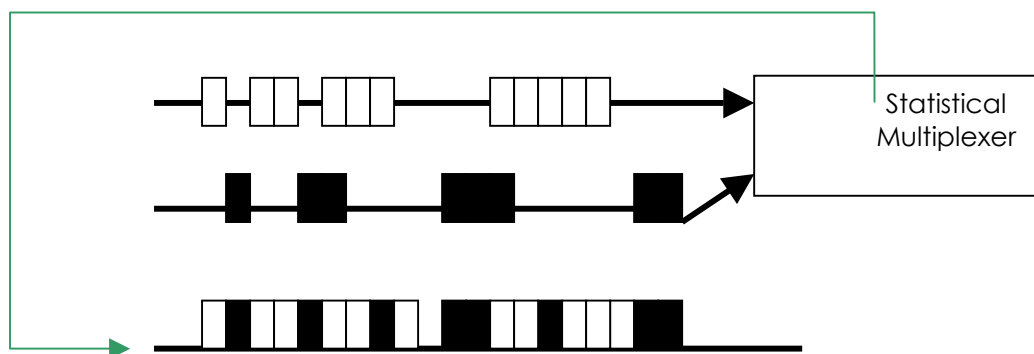
A.1 LES PROTOCOLES RTP ET RTCP

A.1.1 Généralités

RTP/RTCP (Real Time Protocol, Real Time Control Protocol) sont décrits dans le RFC 1889. Il s'agit de la suite protocolaire employée pour transmettre la première conférence sur Internet.

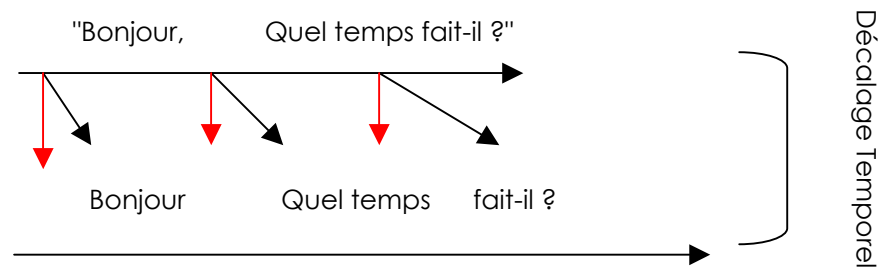
Pour pouvoir décrire le fonctionnement du protocole RTP il faut introduire la notion de "Jitter".

Lorsque nous parlons au téléphone, nous ne conversons pas les 100% du temps avec notre correspondant. Seule une faible partie du temps de connexion sert à transmettre notre voix, pour le reste du temps notre conversation est constituée de silence. C'est pourquoi on utilise des techniques qui permettent de partager plus efficacement la bande passante. Une de celles-ci est le multiplexage statistique où on associe à une même ligne de transmission plusieurs flux de données simultanés. Le multiplexage statistique entraîne plusieurs incertitudes sur le réseau.



Si la ligne de transmission est libre, on peut l'utiliser immédiatement pour transmettre. Si par contre elle est utilisée, il faut attendre un temps avant qu'elle soit à nouveau libre. Ce délai est appelé *Jitter*.

Ce Jitter doit être corrigé du côté du récepteur pour que le discours transmis ne devienne inintelligible. Il faut attendre de recevoir un ensemble suffisant de paquets "voix" pour les transformer en son.



Il faut donc prendre en considération les jitters lorsque l'on transmet un trafic audio.

Revenons maintenant au protocole RTP. Ce protocole a été développé de façon à permettre au récepteur d'une transmission multimédia de corriger les jitters. RTP peut être utilisé pour tout type de transmission temps réel.

RTP définit une méthode pour récupérer de manière la plus convenable les paquets possédant des informations isochrones. Il possède des champs portant sur l'identification du type d'informations transportées, les "timestamps", les numéros de séquence.

Le protocole RTCP est complémentaire à RTP. Il se charge de renvoyer des informations à l'émetteur sur le statut des messages reçus par le récepteur comme par exemple la qualité de la transmission et des informations sur les personnes mêlées à la conversation en cours.

RTP et RTCP n'influencent aucunement le comportement du trafic courant sur le réseau, mais permettent à l'émetteur d'organiser l'envoi de messages de manière efficace vers le récepteur, pour que celui-ci les distingue clairement. Ils ne sont donc pas à confondre avec le système de QoS (Quality of service) comme RSVP (Resource Reservation Protocol).

RTP et RTCP sont utilisés avec le protocole UDP car ils sont optimisés pour des transmissions rapides et sans interactivité. (Il permet le transport de données isochrones sur un réseau de paquets)

Voyons à présent le fonctionnement interne des protocoles RTP et RTCP.

A.1.2 Quelques définitions

RTP Session: Une session RTP est un regroupement de participants qui communiquent par le biais de RTP. Chacun d'eux utilisant deux adresses de transport pour chaque session. L'une pour le flux RTP, l'autre pour les retours du protocole RTP.

Synchronisation source SSRC: Source d'un flux RTP. Elle est identifiée par 32 bits dans l'entête RTP. Tous les paquets RTP issus d'une même source possèdent une même référence temporelle et séquentielle.

Contributing source CSRC: Lorsqu'un flux RTP est le résultat d'une combinaison de plusieurs flux RTP, on introduit dans l'entête RTP la liste de chacun des flux RTP participant à l'interconnexion. (Pas dans H.323)

NTP format: Il s'agit du résultat d'un calcul pour préciser un moment dans le temps (Timestamp) afin que chaque participant puisse avoir la même notion de temps. Ce calcul est basé sur l'année 1900 et sur le nombre de secondes écoulées depuis un jour précis de cette année.

A.1.3 Structure du paquet RTP

Tous les champs jusqu'au CSRC sont présents dans un paquet RTP.

2 bits réservés pour la version de RTP utilisée.

1 bit de padding. Si le payload a été pris pour des raisons d'alignement. Si oui alors le dernier octet du champ de payload indique combien d'octets de paddings ont été utilisés.

1 bit d'extension qui indique la présence d'une extension après les éventuels CSRCs de l'entête fixe.

4 bits de CSRC counts qui donnent le nombre d'identificateurs CSRC à la suite de l'en-tête fixe.

1 bit marqueur. Sa signification vient de H.225 qui demande que pour les codages audio qui supportent la compression des silences, il soit mis à 1 dans le premier paquet reprenant de la voix après une période de silence.

7 bits de payload. Définis dans la RFC 1889 et pour H.323 dans H.225

16 bits de séquence. Démarré de manière aléatoire et incrémenté à chaque paquet envoyé.

32 bits définissant un temps précis. La fréquence d'horloge est définie pour chaque type de payload et est généralement de 8kHz pour la plupart des CODECS vidéo.

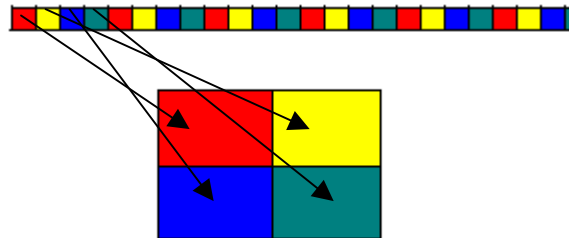
V	P	X	CC	M	Payload type	Sequence Number
Timestamp						
Synchronisation Source Identifier (SSRC)						
Contributing Source Identifier (CSRC)						
Profile dependent					Size	
Data						

Comme dit précédemment, le protocole RTP permet le transport de données isochrones sur un réseau de paquets.

Voici les différentes manières d'utiliser le protocole RTP:

Numéros de séquence et timestamp

Chaque paquet RTP porte un numéro de séquence et un timestamp. En fonction de l'application, ces paquets peuvent être utilisés de différentes manières. Par exemple, une application vidéo peut déduire directement à partir du timestamp quelle partie de l'écran est décrite par le paquet IP.



A.2 SYNTAXE ASN.1

Voici pour commencer l'annexe traitant de ASN.1¹ une phrase récupérée du livre de Mr le professeur John Larmouth qui reprend une bonne description de ce que peut être la syntaxe de notation abstraite (ASN) dans son livre "ASN.1 Complete" .

"Par excellence, ASN.1 (Abstract Syntax notation One) est une notation formelle qui permet à des spécifications d'information d'être traitées par des protocoles de télécommunication de haut niveau sans perte de généralité, d'un point de vue de systèmes logiciels ou matériels."

Nous savons que dans un domaine à architecture distribuée doivent transiter des informations entre les différentes machines en présence. Ces informations peuvent être constituées par des données complexes. On peut donner de beaux exemples de ces transferts entre ces architectures comme par exemple les transferts de communications financières entre des organismes bancaires ou les transferts d'informations entre les architectures client-serveur pour généraliser les cas. On sait également que les informations à transmettre transiteront certainement par divers autres nœuds intermédiaires qui ne possèdent pas forcément les mêmes architectures que les nœuds sources ou destination. D'ailleurs, rien ne dit si les nœuds sources ou destination possèdent la même architecture matérielle, et bien souvent d'ailleurs ce n'est pas le cas. Malgré tout il est essentiel que le message arrive d'un bout à l'autre.

Pour nous rendre compte de quelques problèmes d'architecture il suffit de considérer deux systèmes qui utilisent des représentations d'encodage différentes: certains des mainframes IBM travaillent avec un encodage EBCDIC alors que la plupart des autres utilisent l'encodage ASCII. De même il existe des architectures de processeurs qui traitent avec des mots de 16 ou 32 bits mémoire

Exemple de code tournant sur deux systèmes d'exploitation différents et réalisant le même objectif:

```
typedef struct Record{
    char name[31]
    int age;
    enum {          unknown=0;
          male=1;
          female=2} gender;
} Record;
```

```
type record = {
    name :string;
    age : num;
    gender: t gender }
and t gender = Unknowwn | male | female
```

¹ ASN.1. Ref ITU:X680 ISO IEC: 8824

Le premier code constitue un code C alors que le second est en Objective Caml.

Nous montrons par ces simples exemples également la diversité des langages de programmation qui existent. Ceci démontre le besoin d'une normalisation de notations pour le transfert d'informations entre différentes machines.

Venons en maintenant aux définitions précises des syntaxes concrètes, abstraites et des syntaxes de transfert.

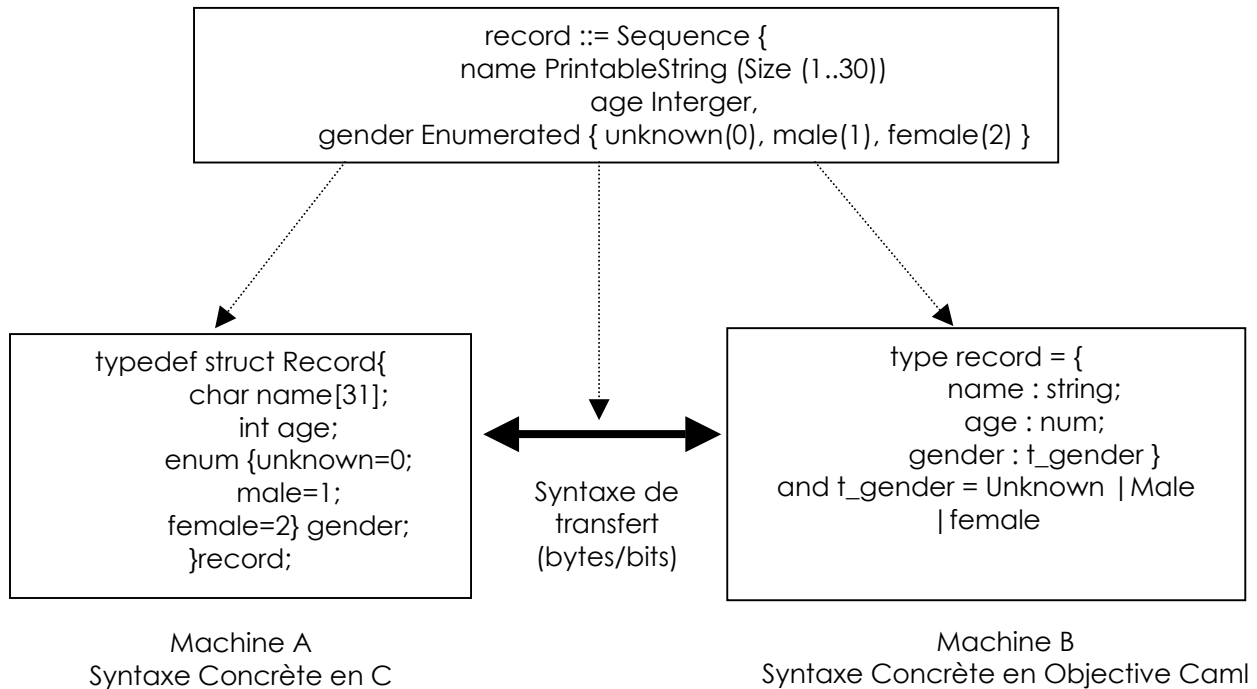
Une syntaxe concrète est une représentation, dans un langage de programmation donné, d'une structure de données qui doit être transférée. Il s'agit d'une syntaxe car elle respecte les règles syntaxiques et grammaticales d'un langage (Le C par exemple). Elle est concrète car elle est traitée par une application et elle respecte les caractéristiques de l'architecture de la machine. Les deux exemples donnés précédemment constituent des syntaxes concrètes.

Pour éviter de tenir compte de l'architecture matérielle du système sur lequel on travaille, il faut donner une description abstraite aux données à transférer. Cette description devra respecter les règles grammaticales et syntaxiques d'un certain langage mais devrait rester indépendante de langages de programmation et ne jamais être implémentées directement sur une machine. Cette description est donc une description abstraite et nous appelons "Syntaxe de notation abstraite" le langage utilisé pour donner cette description.

Plusieurs messages différents peuvent être transférés entre applications. La syntaxe abstraite décrirait alors l'ensemble de ces messages d'une manière plus condensée. A cette fin, la syntaxe abstraite est décrite au moyen d'une grammaire que les données à transférer devront respecter.

La syntaxe abstraite définit précisément la structure de données mais ne précise rien quant à leur interprétation sémantique.

Les données à transférer doivent donc répondre à certaines contraintes pour qu'elles soient compréhensibles sur les machines sur lesquelles elles aboutiront. Ces mêmes données constitueront des données "reçues" sur les machines réceptrices et sont récupérées sous forme de bit ou de flux qui respectent une syntaxe de transfert pour que ces données soient compréhensibles sur la machine réceptrice même.



Bien entendu pour arriver à traiter avec les syntaxes de notation abstraites, il faut que le compilateur soit doté d'une fonction qui permette de traiter avec des règles d'encodage qui définissent des liens entre la notation abstraite et la notation de transfert.

A.3 SIGNALISATION SS7

A.3.1 Les points de signalisation

A chaque point de signalisation dans un réseau SS7 est assigné un numéro unique appelé "Point Code". Ces numéros sont utilisés pour connaître les destinataires et émetteurs des messages. Il existe des tables de routage qui permettent un choix du meilleur chemin pour émettre les messages.

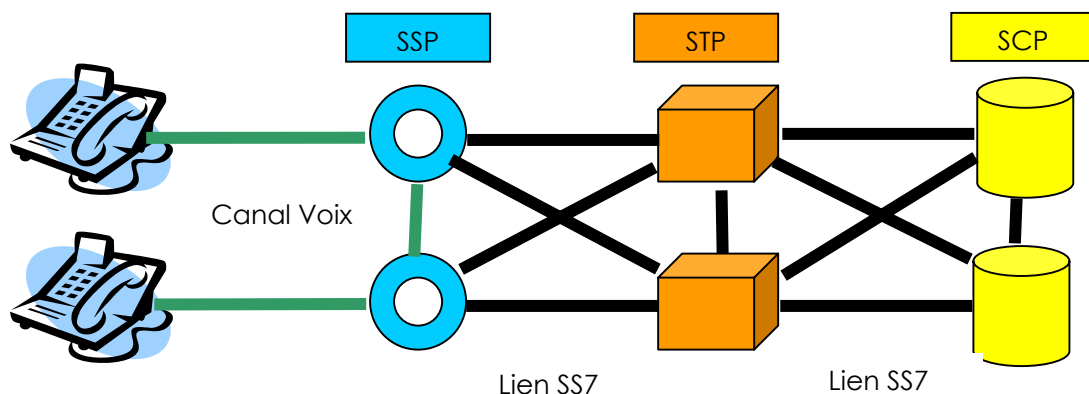
Il existe 3 types de point de signalisation:

Les SSP (Service Switching Point)
Les STP (Signaling Transfer Point)
Les SCP (Service Control Point)

Les SSP sont des commutateurs qui envoient des messages de signalisation à d'autres SSP pour démarrer, gérer et arrêter un circuit vocal. Ils peuvent également lancer des requêtes vers les bases de données centralisées comme les SCP pour déterminer comment router un appel.

Les SCP envoient, suite aux requêtes des SSP, des messages qui contiennent les informations de routage nécessaires.

Le trafic réseau entre les points de signalisation peut être routé via des commutateurs STP. Un STP route chaque message entrant vers un lien de signalisation sortant, sur base de l'information de routage contenue dans le message.



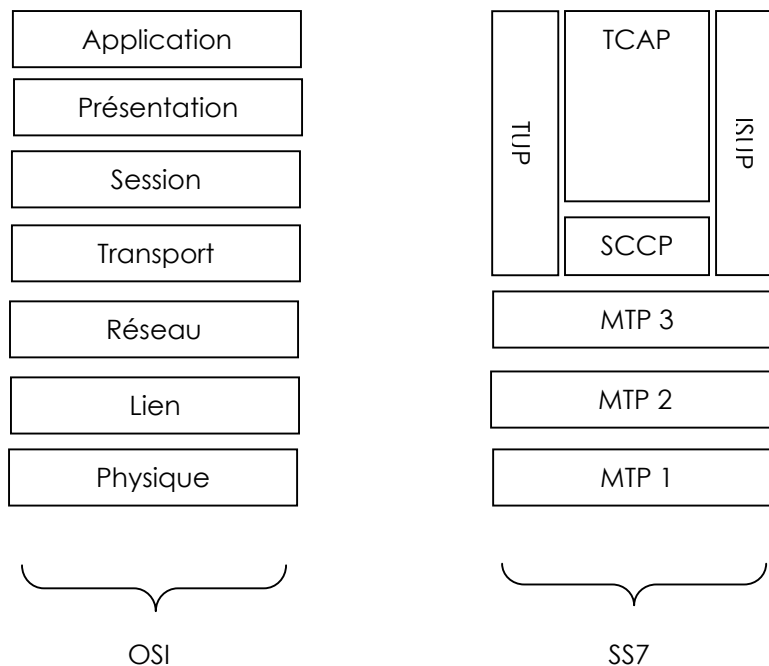
Le réseau SS7 est par mesure de sécurité dédoublé, c'est à dire que les STP et SCP sont installés par paire dans des endroits physiques différents.

Il existe plusieurs types de lien de signalisation en fonction de leur usage. Ceux-ci sont catalogués de A à F.

A.3.2 Pile du protocole SS7

Les fonctions matérielles et logicielles du protocole SS7 sont réparties suivant des couches. Elles ressemblent fortement aux couches du modèle OSI. Le schéma suivant représente ces différentes couches en rapport avec celles du modèle OSI.

La couche MTP1 joue le même rôle que la couche physique OSI. La couche MTP2 définit les fonctions et les procédures relatives au transfert de message de signalisation sur un lien de signalisation. Elle implémente un contrôle de flux, vérifie l'ordre séquentiel des messages, détecte les erreurs. Lors d'une erreur, le message est retransmis. La couche MTP3 fournit la commutation des messages entre les nœuds du réseau SS7.



L'ISDN User Part (ISUP) définit les protocoles et les procédures pour gérer les grandes lignes qui transportent la voix et les données sur les réseaux à commutation de circuit. ISUP est utilisé pour les appels ISDN et non-ISDN. Les appels originaires et à destination d'un même commutateur n'utilisent pas la signalisation ISUP.

SCCP des services avec ou sans connexions au dessus des MTP. SCCP permet l'assignation de nombre à des sous système d'application. SCCP est utilisé comme couche de transport pour les services intelligents tels les numéros verts.

TCAP permet le déploiement de service de réseaux intelligents en supportant des échanges d'information entre des services sans connexion SCCP.

A.4 CALCUL DE DIMENSIONNEMENT

Il se peut que l'on soit confronté à un calcul de performance en traitant du sujet Voice Over IP. Cette annexe présente un calcul simple de dimensionnement de réseau.

Soit un réseau que nous allons dimensionner de manière à transmettre un certain nombre de transmissions vocales simultanément. Nous cherchons à connaître la bande passante minimum qu'il sera nécessaire à réserver pour laisser transiter ce trafic. Nous nous placerons dans le cas précis de l'utilisation du CODEC G.723.1. Pour rappel, un CODEC est un utilitaire qui permet de convertir un signal analogique, tel la voix, en signal numérique pour qu'il puisse être transmis par un réseau de télécommunication numérique.

Le CODEC utilisé pour la conversion de notre signal analogique est doté d'un algorithme qui détecte les moments de silence et adapte grâce à celui-ci le taux d'informations à transmettre; il n'est évidemment pas nécessaire de garder un débit d'information constant pour garder une connexion en activité. Nous noterons par "M" le débit d'informations durant les périodes d'activité et par "m" le débit d'informations en présence de silences.

Les caractéristiques du CODEC utilisé sont présentées dans le tableau 1 où l'on pourra également découvrir les nombres de trames transmises par paquet IP et les différentes valeurs de "m" et "M" pour d'autres CODECS. Ce tableau peut à première vue paraître incorrect de par le fait des valeurs supérieures de "M" par rapport aux valeurs du CODEC en lui-même. En fait on tient compte pour "M" et "m" des débits d'informations utiles à insérer pour les protocoles de transmission en eux-mêmes (RTP, UDP,...).

CODEC	Trames/Paquet IP	"M" (kbit/s)	"m"(kbits/s)
G.723.1 (5.3 kbits/s)	4	8	3.73
G.723.1 (6.4 kbit/s)	4	9.07	3.73
SX7003P	2	20.27	13.87

Tableau 1
Caractéristiques des principaux CODECS

Il est évidemment obligatoire de tenir compte des débits d'informations utiles et des débits des protocoles pour assurer un dimensionnement correct du réseau. Le tableau 2 présente un aperçu des quelques suppléments d'informations nécessaires à l'encapsulation pour les couches physiques.

Ethernet	HDLC	PPP	ATM
26 Octets	7 octets	7 octets	7 octets

Tableau 2
Surplus d'informations nécessaires pour l'encapsulation

Nous présentons à présent quelques calculs détaillés qui permettent de mieux comprendre les valeurs obtenues au tableau 1. Ces valeurs sont calculées d'abord pour "M" du CODEC G.723.1 dans sa version 5.3 kbits/s ensuite pour "m" toujours pour le même CODEC.

Taille de la trame	= 20 octets
Nbre de trame/ paquet IP	= 4
Octets / paquet IP	= 4*20 = 80 octets
Durée de la trame	= 30 ms
Durée paquet IP	= 4*30ms = 120 ms
Débit (kbit/s)	= 80 * 8 / 120 = 5.333 kbit/s

Tableau 3
Valeur pour "M" du G.723.1 (5.3 kbits/s)

Taille de la trame	= 4 octets
Nbre de trame/ paquet IP	= 4
Octets / paquet IP	= 4*4 = 16 octets
Durée de la trame	= 30 ms
Durée paquet IP	= 4*30ms = 120 ms
Débit (kbit/s)	= 16 * 8 / 120 = 1.07 kbit/s

Tableau 4
Valeur pour "m" du G.723.1 (5.3 kbits/s)

Le débit final est celui obtenu sans considérer les informations nécessaires à l'encapsulation. Nous allons maintenant corriger ce débit en tenant compte des informations du tableau 2. Nous constatons qu'il y a un supplément de 40 octets pour l'encapsulation par les protocoles IP(v4)+UDP+RTP. Ce qui nous mène à des débits sensiblement plus élevés de respectivement 8 et 3.73 kbit/s pour "M" et "m".

"M" (kbits/s)	"m" (kbits/s)
8	3.73

Tableau 5
Valeur des débits corrigés

Considérons maintenant notre réseau à dimensionner. Si nous avons N utilisateurs connectés au réseau, il est fort peu probable que ces N utilisateurs communiquent simultanément avec un correspondant. Ce sont les lois des statistiques... Et pour être plus précis, faisons intervenir la théorie de la stochastique (cfr Prof Latouche).

La probabilité qu'il y ait **I** conversations simultanées parmi **N** possibles est donnée simplement par:

$$P(I) = C_N^I * (1-a)^{N-1} = \frac{N!}{I!(N-I)!} * a^I * (1-a)^{N-1}$$

Où "a" est le taux d'activité et est ici approximé par la valeur 0.5 ...

Il ne nous reste plus qu'à calculer le débit moyen pour N conversations simultanées:

$$\text{Débit moyen} = \sum_{I=0}^N P(I) * [IM + (N-I)m]$$

Grâce à un artifice de calcul que nous ne développerons pas, nous obtenons

Débit moyen = N(Ma+m(1-a))

Pour venir concrètement au dimensionnement de notre réseau, nous allons calculer le taux de pertes de paquets qu'il se produit lorsque nous allouons un débit B pour nos N conversations simultanées. Nous en déduirons ensuite le débit B à allouer pour avoir nos N conversations simultanées avec un taux de perte satisfaisant.

Soit **I** le débit nécessaire à I conversations actives. Le taux de perte sera, si on a alloué une bande passante **B**, de $\tau = (I-B)/I$. Il ne reste plus qu'à augmenter B pour qu'on atteigne un τ raisonnable.

Le tableau 6 donne un aperçu des valeurs obtenues pour les transmissions de N canaux vocaux pour un taux de perte de 1%.

Nbre Conversations	1	3	5	7
Taux moyen	6.4	19.2	32.0	44.8
Taux minimum	8.9	25.7	39.2	52.5
%age surplus	40	34	22	17

Tableau 6

Aperçu des valeurs des débits nécessaires en fonction du nombre de conversations

En conclusion, s'il y a peu de communications à faire transiter, il faut dimensionner le réseau comme si toutes les communications étaient actives, ce qui engendre une très mauvaise utilisation des ressources avec un taux de surplus de 40%...

A.5 DEPENDANCES

La compilation du logiciel OpenGatekeeper nécessite l'installation préalable de deux librairies. Il s'agit des librairies PWLib et OpenH323. Nous allons sans entrer dans les détails décrire dans cette section les principales caractéristiques de ces deux librairies.

La librairie PWLib a été réalisée dans l'optique de pouvoir écrire des applications tournant à la fois sur les systèmes d'exploitation Windows et Unix dans son environnement X. Le projet initial comptait également l'intégrer pour les systèmes d'exploitation Macintosh mais cela n'a jamais été réalisé. Il s'agit d'une librairie assez complexe et moyennement volumineuse. Elle contient des classes pour traiter la portabilité des entrées-sorties, du multi-threading. De même elle porte les démons Unix sous Windows et les services Windows sous Unix. Elle gère également toutes sortes de protocoles Internet.

Voici un exemple repris de la littérature qui décrit l'application "Hello World" en utilisant la librairie PWLib:

```
//hello.cxx

class Hello : public PProcess      //héritage de PApplication si GUI
{
    PCLASSINFO(Hello,PProcess)
    public:
        void Main();
};

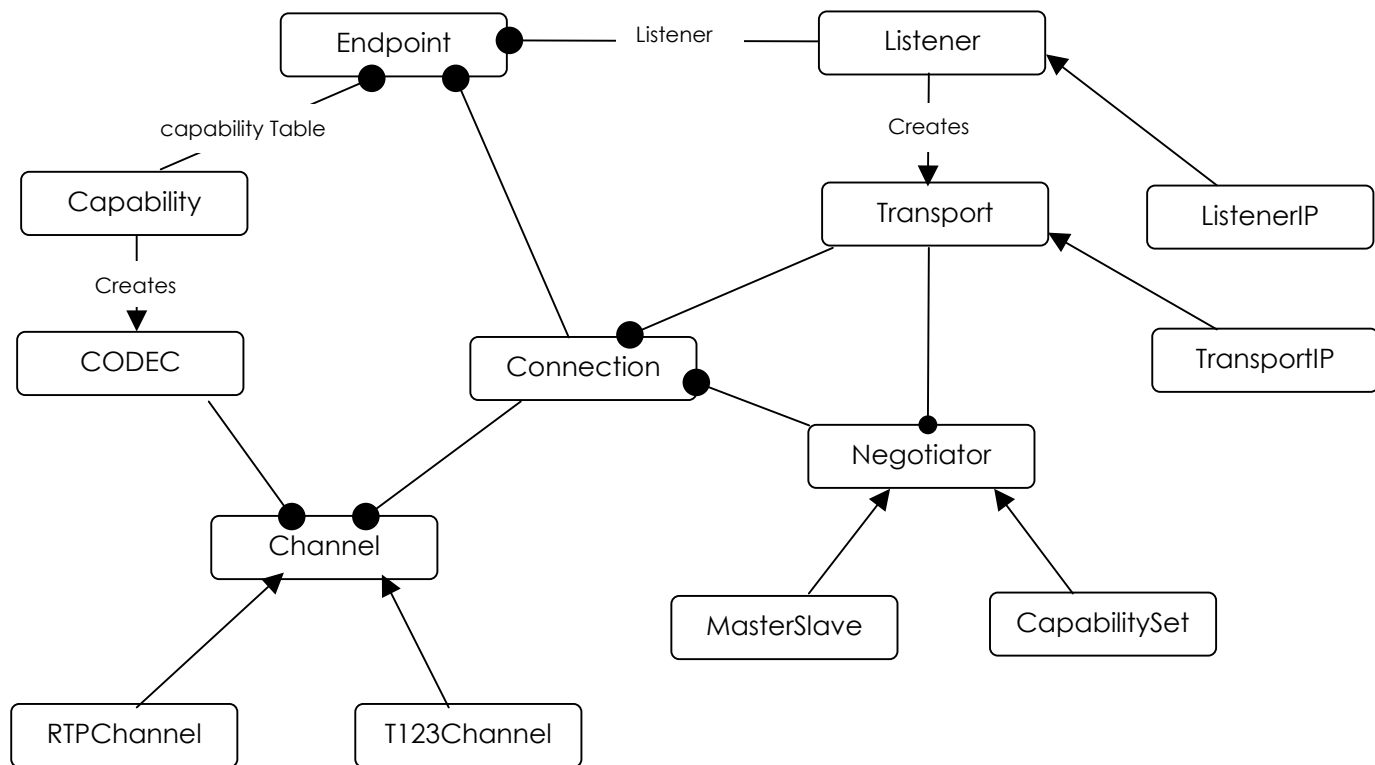
PCREATE_PROCESS(Hello) //macro qui définit la fonction main() et
                        //crée une instance de Hello. Ceci assure
                        //une initialisation dans un bon ordre.

void Hello::Main()
{
    cout << "Hello world!\n";
}

//end of hello.cxx
```

La librairie H.323 possède l'ensemble des classes qui permettent de définir les modules tels que les définit le standard H.323. Cette librairie se compile en ayant au préalable installé la librairie PWLib. Le schéma de la figure suivante représente l'architecture des classes présentes dans la librairie selon un diagramme de Booch.

La classe fondamentale de cette librairie est la classe H323Endpoint. Les applications développées sur base du standard H.323 possèdent au moins une classe descendant de cette dernière.



Ce schéma mérite une petite explication: Le ListenerIP est une classe qui est définie pour l'écoute des paquets IP. Chaque Listener lance un thread qui monitorise son protocole et quand un nouvel appel entrant est détecté, il crée une instance de la classe Transport, qui hérite d'une classe transport propre au protocole en question (TransportIP).

Bibliographies

- [1] Prof John Larmouth, *ASN.1 Complete*. Open Systems Solutions, 1999.
- [2] Olivier Dubuisson, *ASN.1 Communication between Heterogeneous System*. Traduit du français par Philippe Fouquart, OSS Nokalva, 2000.
- [3] Andrew S. Tanenbaum, *Computer Networks*, 3rd ed. New York: Prentice Hall.
- [4] Floyd Wilder, *A Guide to the TCP/IP Protocol Suite*. Boston London: Artech House.
- [5] Uyless Black, *Voice over IP*. New York: Prentice Hall series in Advanced Communications Technologies.
- [6] O. Hersent, D. Gurle and H. Petit, *IP Telephony Packet Based Multimedia Communications Systems*. Addison-Wesley.
- [7] D. Curry, *C on the UNIX System*. Sebastopol: O'Reilly & Associates, 1991.
- [8] J. Ryan, *Voice Over IP (VoIP)*. Natick: The Technology Guide Series, 1998.
- [9] H. Schulzrinne, J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony".
- [10] H. Liu and P. Mouchtaris, "Voice over IP Signaling: H.323 and Beyond," in *IEEE Communications Magazine*, pp. 142-148, Oct 2000.
- [11] H. Schulzrinne and J. Rosenberg, "The Session Initiation Protocol: Internet-Centric Signaling," in *IEEE Communications Magazine*, pp. 134-141, Oct 2000.
- [12] H. Schulzrinne and J. Rosenberg, "The IETF Internet Telephony Architecture and Protocols," in *IEEE Network*, pp. 18-23, May June 1999.
- [13] M. Hamdi, O. Verscheure and J-P Hubaux, "Voice Service Interworking for PSTN and IP Networks," in *IEEE Communications Magazine*, pp. 104-111, May 1999.
- [14] K. Asatani, "IP and Telecommunication Integration: De Jure and De Facto Standards Have Entered a New Era," in *IEEE Communications Magazine*, pp. 140-147, Jul 1999.
- [15] *White Paper, Application Powered Networks with SIP*. Release 2.0 Ubiquity Software Corporation.
- [16] "TTT-Services update presentation" <<http://www.ttt-services.org/>> Accessed Feb 2001.

- [17] B. Michael , *SIP Rules!* . Computer Telephony, 2000,
<<http://www.computertelephony.com/article/CTM20000515S0003>> .
- [18] "Ten Telecom (Trans European Telecom)," <http://156.54.253.12/tentelecom/fr/context.html>
- [19] H. Schulzrinne, "H. Schulzrinne's Home Page,"
<<http://www.cs.columbia.edu/~hgs/resume/resume.html>> Accessed May 2001.
- [20] "The Resolution Services," <<http://www.resolution-service.org/>> Accessed May 2001.
- [21] "The OpenH323 Project," <<http://www.openh323.org/>> Accessed May 2001.
- [22] "The Sipcenter.com," <<http://www.sipcenter.com>> Accessed May 2001.
- [23] "Ubiquity Software Corporation "
<<http://www.ubiquity.net/solutions/wpaper.htm>> Accessed May 2001.
- [24] "H.323 vs. SIP Telephony," <
<http://www.fokus.gmd.de/research/cc/globe/projects/ipt/sip.html>> Accessed May 2001.
- [25] "Session Initiation Protocol (SIP)," <
<http://www.cs.columbia.edu/~hgs/sip/>> Accessed May 2001.
- [26] "TTT Services," < <http://www.ttt-services.org/>> Accessed May 2001.
- [27] "OpenGatekeeper," <<http://www.opengatekeeper.org>> Accessed Feb 2001.
- [28] " A resource for packet-switched conversational protocols,"
<<http://www.packetizer.com/>> Accessed May 2001.
- [29] " IEC tutorials " <<http://www.iec.org/tutorials/>> Accessed May 2001.
- [30] "International Telecommunication Union,"
<<http://www.itu.int/home/index.html>> Accessed May 2001.