N° d'ordre : 03/RS/TCO Année universitaire : 2007-2008

UNIVERSITE D'ANTANANARIVO

ECOLE SUPERIEURE POLYTECHNIQUE

DEPARTEMENT TELECOMMUNICATION

MEMOIRE DE FIN D'ETUDES

en vue de l'obtention

du DIPLOME d'INGENIEUR

Spécialité : Télécommunication

Option : Réseaux et Systèmes

par : **ANDRIANARISON Maherizo Valinaina**

LA TELEPHONIE SUR IP : OPTIMISATION DU TEMPS DE LATENCE PAR AMELIORATION AU NIVEAU DU ROUTAGE IP

Soutenu le Mardi 06 Janvier 2009 à 10h devant la commission d'Examen composée de :

Président :

Monsieur RANDRIAMITANTSOA Paul Auguste

Examinateurs :

Monsieur RAKOTOMALALA Mamy Alain

Monsieur RADONAMANDIMBY Edmond

Monsieur RASAMOELINA Jacques Nirina

Directeur de mémoire :

Monsieur ANDRIAMIASY Zidora

REMERCIEMENTS

Je rends gloire à Dieu pour sa bonté et sa fidélité de m'avoir donné la force et la santé durant la réalisation de ce mémoire.

J'exprime ma gratitude à Monsieur RAMANATSIZEHENA Pascal, Professeur titulaire, Directeur de l'Ecole Supérieure Polytechnique d'Antananarivo.

Je suis particulièrement reconnaissant à Monsieur RANDRIAMITANTSOA Paul Auguste, Professeur, Chef du Département Télécommunication qui, malgré ses lourdes responsabilités, me fait l'honneur de présider le jury de ce mémoire ;

Je suis très reconnaissant envers Monsieur ANDRIAMIASY Zidora, Maître de conférences, Enseignant au sein du Département Télécommunication, directeur de ce mémoire qui a été pour moi plus qu'un encadreur. Je le remercie pour le temps qu'il m'a accordé et ses précieux conseils.

Je tiens à témoigner toute ma gratitude à :

- Monsieur RAKOTOMALALA Mamy Alain, Assistant, Enseignant au sein du Département Télécommunication,
- Monsieur RADONAMANDIMBY Edmond, Assistant, Enseignant au sein du Département Télécommunication,
- Monsieur RASAMOELINA Jacques Nirina, Assistant, Enseignant au sein du Département Télécommunication,

pour l'honneur qu'ils me font de participer à mon jury.

Je remercie tous les Enseignants et personnel de l'Ecole Supérieure Polytechnique pour la formation durant ces cinq années.

Je témoigne toute ma reconnaissance à toute la grande famille qui m'a toujours assisté durant mes cinq années d'études loin de mes parents.

J'adresse une pensée spéciale à tous ceux qui ont, de près ou de loin, participé à la réalisation de ce mémoire.

Je termine ici en dédiant ce mémoire à mes parents qui m'ont donné le goût des études et offert un soutien sans faille.

TABLE DES MATIERES

REMERCIEMENTS ..... **i**
TABLE DES MATIERES ..... **ii**
NOTATIONS ..... **vi**
ACRONYMES ..... **vii**
INTRODUCTION GENERALE **1**

CHAPITRE 1 RESEAUX TCP/IP **3**
1.1. Introduction à TCP/IP ***3***
1.1.1. Généralités **3**
1.1.2. Principe des couches et protocoles **4**
1.2. Les architectures OSI et TCP/IP ***5***
1.2.1. Le modèle OSI **5**
1.2.1.1. Principe général ***5***
1.2.1.2. Rôles des couches ***6***
1.2.2. Le modèle TCP/IP **7**
1.2.3. Comparaison entre le modèle TCP/IP et le modèle OSI **8**
1.2.4. Mécanismes d'encapsulation **9**
1.3. Le protocole Internet ***11***
1.3.1. Description du protocole IP **11**
1.3.2. Principe de l'adressage des machines **11**
1.3.2.1. Structure d'adresses IP ***12***
1.3.2.2. Classes d'adresses IP ***12***
1.3.3. Transport et format de datagrammes IP **14**
1.3.3.1. Datagramme IP ***14***
1.3.3.2. Routage des datagrammes sur Internet ***16***
1.3.3.3. Différentes phases de routage IP ***21***
1.3.3.4. Détermination des distances des chemins de réseau à l'aide de métriques ***24***
1.3.4. Résolution d'adresses logiques **25**
1.3.4.1. Le protocole ARP ***25***
1.3.4.2. Le protocole RARP ***25***
1.4. Le protocole IPv.6 ..... ***26***
1.5. Les protocoles de la couche « Transport » ***27***
1.5.1. Le protocole TCP **27**
1.5.2. Le protocole UDP **28**
1.5.3. Comparaison des protocoles TCP et UDP **28**
1.6. Conclusion ***29***

CHAPITRE 2 TELEPHONIE SUR IP **30**
2.1. Présentation de la Voix sur IP (VoIP) ***30***
2.1.1. Introduction à la VoIP **30**

2.1.1.1. Qu'est-ce que la VoIP ? *30*
2.1.1.2. Contexte technologique de la VoIP *31*
2.1.1.3. De la voix à l'IP en 8 étapes *31*
2.1.1.4. Les différentes architectures *31*
2.1.2. Les codages de la VoIP **34**
2.1.2.1. Introduction *34*
2.1.2.2. La VoIP et les codecs *35*
2.2. Normes et protocoles de la VoIP ***37***
2.2.1. Protocole H.323 ..... **37**
2.2.1.1. Introduction *37*
2.2.1.2. Fonctionnement *38*
2.2.1.3. Architecture H.323 ..... *38*
2.2.1.4. Les protocoles secondaires de H .323 ..... *39*
2.2.1.5. Phases d'appel de H.323 ..... *40*
2.2.1.6. H323 dans le modèle OSI *41*
2.2.2. Protocole SIP **42**
2.2.2.1. Introduction *42*
2.2.2.2. Fonctionnement *42*
2.2.2.3. Les clients et les serveurs de SIP *42*
2.2.2.4. Exemple d'établissement d'un appel SIP à travers un proxy *44*
2.2.2.5. Comparaison avec H.323 ..... *45*
2.2.2.6. Conclusion *46*
2.2.3. Protocole MGCP **46**
2.2.4. Protocoles RTP et RTCP **46**
2.2.4.1. Protocole RTP *46*
2.2.4.2. Le couple RTP et RTCP *47*
2.3. Les problèmes de la VoIP et qualité de service ***48***
2.3.1. Problèmes pour la VoIP **48**
2.3.2. Les paramètres de la VoIP **48**
2.3.2.1. La latence *49*
2.3.2.2. La gigue *51*
2.3.2.3. La perte et le déséquencement de paquets *52*
2.3.3. Mécanismes de QoS **53**
2.3.3.1. Mécanisme IntServ *53*
2.3.3.2. Mécanisme DiffServ *55*
2.3.3.3. MPLS *55*
2.4. La Téléphonie sur IP (ToIP) ***55***
2.4.1. Introduction **55**
2.4.2. Principe de base **56**
2.4.2.1. Traitement de la voix *56*
2.4.2.2. Les trois étapes d'une communication en ToIP *57*
2.4.3. Différence entre VoIP et ToIP **58**

2.4.4. Les différents éléments pouvant composer un réseau 59
2.4.5. Avantages 60
2.4.6. Sécurité de la ToIP 60
2.4.6.1. Pourquoi la sécurité de la téléphonie 60
2.4.6.2. Vulnérabilités et contraintes de la ToIP 61
2.4.6.3. Définitions relatives à la sécurité 61
2.4.6.4. Sécurité et VoIP 62
2.4.7. ToIP et NGN 65
2.4.7.1. Objectif 65
2.4.7.2. NGN et la Téléphonie IP 65
2.5. Conclusion 67

CHAPITRE 3 ROUTAGE IP ET OPTIMISATION 69
3.1. Commutation et routage 69
3.1.1. Commutation 69
3.1.1.1. Commutation de circuits 69
3.1.1.2. Transfert de paquets 69
3.1.2. Routage 71
3.1.2.1. Routage centralisé 72
3.1.2.2. Routage distribué 74
3.1.2.3. Routage mixte 75
3.2. Optimisation du temps de latence 75
3.2.1. Enoncé du problème 75
3.2.2. Solutions 77
3.2.2.1. Choix des routes 77
3.2.2.2. Mise en forme du trafic 77
3.2.2.3. Ingénierie de trafic 78
3.2.2.4. Calcul du chemin 78
3.2.2.5. Ordonnancement 78
3.2.2.6. Matériel 79
3.2.2.7. Techniques d'amélioration de routage 79
3.3. Conclusion 81

CHAPITRE 4 SIMULATION SOUS C++ 82
4.1. Présentation du logiciel 82
4.1.1. But du logiciel 82
4.1.2. Spécifications du logiciel 83
4.1.3. Modèles utilisés 84
4.1.3.1. Gestion de files d'attente 84
4.1.3.2. Modèle de trafic 84
4.1.3.3. Capacité nécessaire 85
4.1.4. Fonctionnement de l'ensemble 87
4.1.5. Algorithme final utilisé 88

4.2. Mode d'utilisation du logiciel *89*
4.2.1. Les données du réseau **89**
4.2.1.1. Les nœuds du réseau *89*
4.2.1.2. Les arcs du réseau *89*
4.2.2. Simulation **90**
4.3. Conclusion *95*
CONCLUSION GENERALE **96**
ANNEXE 1 NOTION DE THEORIE DES GRAPHES **97**
ANNEXE 2 NOTION SUR LA THEORIE DE FILES D'ATTENTE **101**
ANNEXE 3 CODE SOURCE DE LA SIMULATION **104**
BIBLIOGRAPHIE **107**

NOTATIONS

φ_u	Quantité de flot sur l'arc u
γ_u	Coût de passage d'une unité de flux sur l'arc u
λ	Taux de naissance
μ	Taux de mort
$\sum$	Somme
A	Trafic offert
b_u	Bornes inférieures de flux
C_i	Capacité de la $i^{ème}$ ligne de communication
c_{ij}	Capacité de l'arc
c_u	Bornes supérieures de flux
$d^-(x)$	Degré entrant du sommet x
$d^+(x)$	Degré sortant du sommet x
d(x,y)	Distance entre deux sommets x et y
$E(A,n)$	Formule d'Erlang en fonction de A et n
f(x,y)	Quantité de flot passant par l'arc (x,y)
$G_{k,c}$	Graphe auxiliaire
$\bar{G}(\varphi)$	Graphe d'écart associé à φ
Gbps	Gigabit par seconde
GHz	Gigahertz
kbps	Kilobit par seconde
km	Kilomètre
kHz	Kilohertz
Mbps	Mégabit par seconde
MHz	Mégahertz
ms	Milliseconde
n	Nombre de ressources disponibles
s	Seconde
T	Temps de réponse du routeur

ACRONYMES

ACELP	Algebraic Codebook Excited Linear Prediction
ADPCM	Adaptive Differential Pulse Code Modulation
ADSL	Asymetric Digital Subscriber Line
AH	Authentication Header
ARP	Address Resolution Protocol
ARPANET	Advanced Research Projects Agency Network
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
CBQ	Class Based Queuing
CELP	Code Excited Linear Prediction
CODEC	Codeur Décodeur
CRA	Customer Response Application
CS-ACELP	Conjugate Structure Algebric Code Excited Linear Prediction
CTI	Computer Telephony Integration
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiate Service
DNS	Domain Name Service
DOD	Department Of Defense
DOS	Deny Of Service
DRARP	Dynamic Reverse Address Resolution Protocol
DSCP	Differentiated Service Code Point
DSL	Digital Subscriber Line
DTLS	Datagram Transport Layer Security
EGP	Exterior Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
EoIP	Everything over Internet Protocol
ESP	Encapsulating Security Payload
ETSI	European Telecommunication Standards Institute
E&M	Earth & Mouth
FCFS	First Come First Served

FIFO	First In First Out
FTP	File Transfer Protocol
GLP	Gateway Location Protocol
GSM	Global System for Mobile communications
HDLC	High Data level Link Control
HTTP	HyperText Transfer Protocol
ICMP	Internet Control Message Protocol
IDRP	InterDomain Routing Protocol
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IGMP	Internet Group Membership Protocol
IGRP	Interior Gateway Router Protocol
ILS	Internet Locator Server
INTERNET	Interconnected Network
INTERNIC	International Network Information Center
IntServ	Integrated Services
IP	Internet Protocol
IPBX	IP Private Branch eXchange (IP-PBX)
IPSec	IP Security
IRTF	Internet Research Task Force
IS-IS	Intermediate System to Intermediate System
ISO	International Standard Organization
ISDN	Integrated Services Digital Networks
ISP	Internet Service Provider
ITU	International Telecommunication Union
IVR	Interactive Voice Response
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LD-CELP	Low-Delay CELP
LLC	Logical Link Control
LPC	Linear Predictive Coding
LS	Location Server

MAC	Media Access Control
MAN	Metropolitan Area Network
MCU	Multipoint Control Unit
MEGACO	Media Gateway Controller
MGCP	Media Gateway Control Protocol
MIC	Modulation par Impulsion Codée
MMUSIC	Multiparty Multimedia Session Protocol
MOS	Mean Opinion Score
MPLS	Multi Protocol Label Switching
MP-MLQ	Multi Pulse - Maximum Linklihood Quantization
MTU	Maximum Transfer Unit
NFS	Network File System
NGN	Next Generation Network
NIC	Network Information Center
OMERT	Office Malagasy d'Etudes et de Régulation des Télécommunications
OSI	Open System Interconnexion
OSPF	Open Shortest Path First
PABX	Private Automatic Branch eXchange
PAPS	Premier Arrivé Premier Servi
PBX	Private Branch eXchange
PC	Personal Computer
PCM	Pulse Code Modulation
PDU	Protocol Data Unit
PGPS	Packet Generalized Processor Sharing
POP	Post Office Protocol (Point Of Presence)
PPP	Point to Point Protocol
PS	Proxy Server
PSTN	Public Switched Telephone Network
QoS	Quality of Service
QSIG	Q-Signaling
RARP	Reverse Address Resolution Protocol
RAS	Registration, Admission and Status

RED	Random Early Detection
RFC	Request For Comments
RIP	Routing Information Protocol
RNIS	Réseau Numérique à Intégration de Service
RS	Redirect Server
RSVP	Resource reSerVation Protocol
RTC	Réseau Téléphonique Commuté
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SIP	Session Initiation Protocol
SPIT	SPam over IP Telephony
SRTP	Secure Real-time Transport Protocol
TCP	Transport Control Protocol
TCP/IP	Transport Control Protocol/Internet Protocol
TLS	Transport Layer Security
TTL	Time To Live
ToIP	Telephony over Internet Protocol
ToS	Type of Service
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UIT-T	Union Internationale des Télécommunications - Télécommunications
UMTS	Universal Mobile Telecommunication System
VLAN	Virtual Local Area Network
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
WAN	Wide Area Network
WAP	Wireless Application Protocol
WFQ	Weighted Fair Queuing
WiFi	Wireless Fidelity
WRED	Weighted Random Early Detection

INTRODUCTION GENERALE

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.

Parmi les nouvelles technologies de communication, il en existe une qui s'est laissé progressivement découvrir au grand public durant la fin du siècle dernier. Il s'agit bien entendu d'Internet. Actuellement, le monde entier est relié au moyen de routes virtuelles qui constituent la toile mondiale. 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 90, justement pour pallier les problèmes des coûts des communications téléphoniques nationales et internationales. Une nouvelle 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, c'est la Voice Over IP (VoIP).

La VoIP 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.

La Téléphonie sur IP (ToIP) s'annonce d'ores et déjà comme un des changements majeurs dans les années à venir en matière de télécommunications d'entreprise. Au-delà des apports technologiques (simplification de l'exploitation, unification des applications, etc.), la ToIP représente un gain de confort et de productivité pour les utilisateurs. Elle constitue une avancée significative dans la convergence entre les réseaux de téléphonie et les réseaux de données. Elle rapproche deux mondes, deux infrastructures, gérés jusqu'à maintenant de manière distincte au sein des entreprises. En effet, la convergence du triple play (voix, données et vidéo) fait partie des enjeux principaux des acteurs de la télécommunication aujourd'hui. Au lieu de disposer à la fois d'un réseau informatique et d'un réseau téléphonique commuté (RTC), l'entreprise peut donc, grâce à la VoIP, tout fusionner sur un même réseau.

Cependant, la téléphonie sur Internet est encore loin de satisfaire aux exigences de qualité de service attendues pour ce type de service, même si de fortes améliorations sont prévisibles. En effet, la plus grande contrainte de la téléphonie sur IP est le délai de bouche à oreille, qui est déjà dans la plupart des cas dans la limite de ce qui est considéré comme acceptable par un utilisateur normal. Il devient donc particulièrement important de savoir contrôler et caractériser la qualité de service d'un réseau IP, surtout ce délai que l'on appelle aussi temps de latence.

Ce mémoire traite le problème de l'optimisation du temps de latence en téléphonie sur IP en utilisant plusieurs algorithmes : recherche du plus court chemin entre les différents routeurs du réseau, gestion de files d'attente, contrôle de flux et contrôle de congestion.

Le premier chapitre aura comme rôle de présenter une vue d'ensemble des réseaux TCP/IP avec les principes des protocoles utilisés et des différentes couches ainsi que les architectures.

Le second chapitre sera consacré à la technologie de voix sur IP et de téléphonie sur IP, où seront abordés une présentation générale de cette technologie, puis les différentes normes et protocoles utilisés. Ensuite, nous allons détailler les problèmes de la VoIP et la qualité de service, qui sont les vifs du sujet. Et enfin, une description de la ToIP avec les différentes sécurités et un petit aperçu de la nouvelle génération de réseau NGN.

Le troisième chapitre parlera du routage IP et l'optimisation à faire pour contrôler la QoS de la VoIP. Nous allons essayer d'exposer les hypothèses et problématiques liées à ce mémoire en faisant une modélisation du réseau sous forme de graphe. L'algorithme de Dijkstra est utilisé pour la recherche du plus court chemin dans le réseau.

Le dernier chapitre « Simulation de l'optimisation du temps de latence » s'occupera de la présentation du logiciel conçu sous C++.

Une conclusion générale est présentée à la fin de ce mémoire. Elle résume le travail effectué ainsi que les points qui restent à étudier et à définir dans un proche avenir.

CHAPITRE 1 RESEAUX TCP/IP

1.1. Introduction à TCP/IP [1] [2] [26] [27]

1.1.1. Généralités

Un ***réseau*** est constitué d'un ensemble d'ordinateurs et de périphériques interconnectés par différents liens de communication, capables d'échanger des données en temps réel ou quasi-réel. Chaque élément d'un réseau est appelé nœud.

L'***Internet*** (Interconnected Network) n'est pas un réseau au sens classique du terme, il n'y a rien qui puisse ressembler à une énorme multinationale qui fournirait des services dans tous les continents. En fait, l'Internet est une connexion anarchique de réseaux appartenant à divers propriétaires et de différents types. Le mode opératoire de l'Internet est très simple, des réseaux sont reliés par des programmes inter-réseaux. Tout hôte de l'Internet qui souhaite faire tourner une application Internet est supposé abriter une instance de ce programme. Les applications accèdent en général au programme Internet à travers un programme de transport TCP ou UDP.

Les données qui transitent sur le réseau sont fragmentées en tronçons appelés ***paquets***. Chacun est manipulé individuellement et mentionne les adresses d'origine et de destination. Le paquet est emballé pour traverser le réseau dans lequel il est apparu puis déballé et acheminé par un programme Internet suivant un autre type d'encapsulation. En d'autres termes, à l'intérieur de chaque réseau, les nœuds utilisent la technologie spécifique de leur réseau (Ethernet, X.25, ...). Le logiciel d'interconnexion de la couche réseau encapsule ces spécificités et offre un service commun à tous les applicatifs, faisant apparaître l'ensemble de ces réseaux disparates comme un seul et unique réseau.

La figure 1.01 suivante montre un exemple d'interconnexion de réseaux.

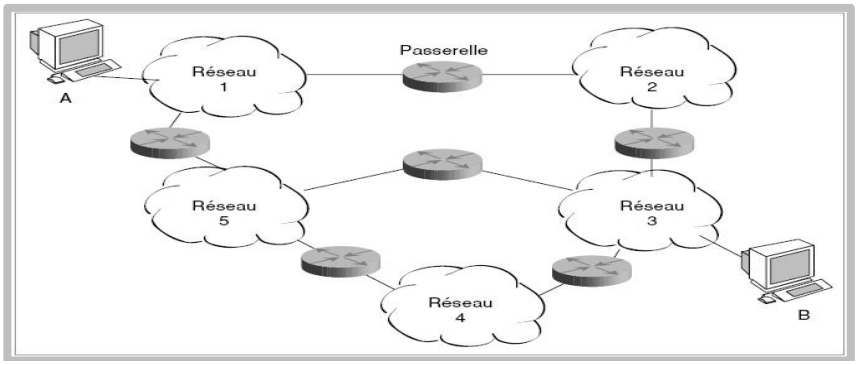

Figure 1.01 : Notion d'interconnexion de réseaux

1.1.2. Principe des couches et protocoles

Pour réduire la complexité de conception, les réseaux sont organisés en série de couches ou niveaux, chacune étant construite sur la précédente. Le nombre de couches, leur nom et leur fonction varie d'un réseau à un autre. Cependant, dans chaque type de réseau, l'objet de chaque couche est d'offrir certains services aux couches plus hautes, en leur épargnant les détails de la mise en œuvre de ces services.

La couche N d'une machine gère la conversation avec la couche N d'une autre machine. Les règles et conventions pour cette conversation sont connues sous le nom de protocole de la couche N. En réalité, aucune donnée n'est transférée directement d'une couche N d'une machine à la couche N d'une autre machine mais chaque couche passe les données et les contrôles à la couche immédiatement inférieure, jusqu'à la plus basse. Entre chaque paire de couches adjacentes, on trouve une interface. L'interface définit les opérations élémentaires et les services que la couche inférieure offre à la couche supérieure.

TCP/IP est un des langages utilisés dans les réseaux. Le terme TCP/IP n'est pas limité à l'expression Transport Control Protocol / Internet Protocole. TCP/IP recouvre toute une famille de protocoles comme UDP (User Datagram Protocol), Telnet, FTP (File Transfer Protocol), http (HyperText Transfer Protocol) …

L'ensemble des couches et protocoles est appelé *architecture du réseau.*

La hiérarchie d'implémentation de TCP/IP est représentée dans la figure 1.02 suivante.

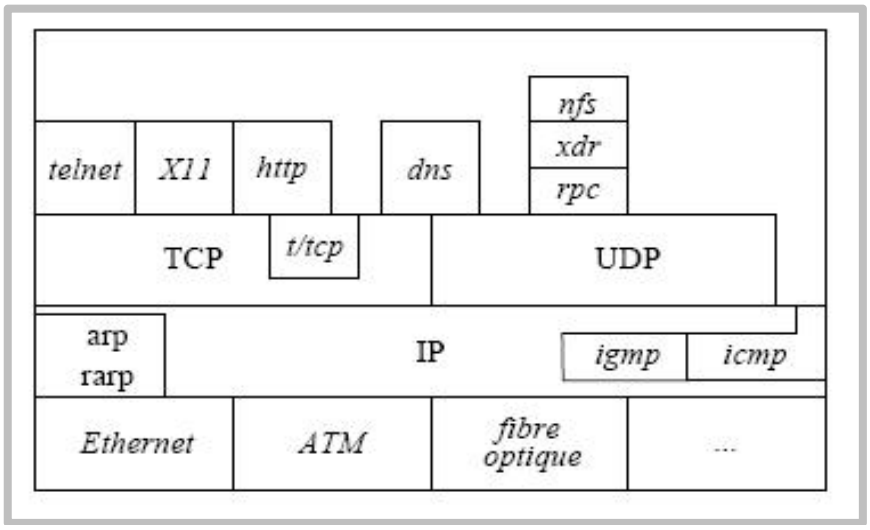

Figure 1.02 : Hiérarchie d'implémentation de TCP/IP

1.2. Les architectures OSI et TCP/IP [1] [2] [3] [4] [5] [26] [27]

1.2.1. Le modèle OSI

1.2.1.1. Principe général

Le modèle OSI (Open System Interconnexion) a été développé en 1978 par l'ISO (International Standard Organization) afin que soit défini un standard utilisé dans le développement de système ouvert. Les réseaux s'appuyant sur le modèle OSI parlent le même langage, ils utilisent des méthodes de communication semblables pour échanger des données. Le modèle OSI a sept couches : la couche ***physique***, la couche ***liaison de données***, la couche ***réseau***, la couche ***transport***, la couche ***session***, la couche ***présentation***, et la couche ***application***.

Figure 1.03 : Le modèle OSI

Les principes ayant conduit aux sept couches sont les suivantes :

- Une couche ne peut être créée que quand un niveau différent d'abstraction est nécessaire.
- Chaque couche doit fournir une fonction bien définie.
- Les fonctions de chaque couche doivent être choisies en pensant à la définition de protocoles normalisés internationaux.
- Les caractéristiques d'une couche doivent être choisies pour qu'elles réduisent les informations transmises entre les couches.
- Le nombre de couches doit être suffisamment grand pour éviter la cohabitation dans une même couche de fonctions très différentes et suffisamment petit pour éviter que l'architecture ne devienne difficile à maîtriser.

1.2.1.2. Rôles des couches

La couche ***physique*** s'occupe de la transmission des bits de façon brute sur un circuit de communication. Les bits peuvent être encodés sous forme de 0 ou de 1 ou sous forme analogique. Elle fait intervenir des interfaces mécaniques et électriques sur le média utilisé.

La couche ***liaison de données*** prend les données de la couche physique et fournit ses services à la couche réseau. Les bits reçus sont groupés en unité logique appelée *trame***.** Les fonctions de contrôle de flux et d'erreurs y sont éventuellement assurées avec un contrôle d'accès au support.

La couche ***réseau*** gère la connexion entre les différents nœuds du réseau. Il comporte trois fonctions principales : le contrôle de flux, le routage et l'adressage.

La couche ***transport*** effectue des contrôles supplémentaires à la couche réseau. Elle réalise le découpage des messages en paquets pour la couche réseau. Elle doit également gérer les ressources de communication en gérant un contrôle de flux ou un multiplexage. C'est l'ultime niveau qui s'occupe de l'acheminement de l'information.

Le rôle de la couche ***session*** est de fournir aux entités de présentation les moyens nécessaires à l'organisation et à la synchronisation de leur dialogue. Elle a pour but d'ouvrir et de fermer des sessions entre les utilisateurs et possède par conséquent des fonctionnalités nécessaires à l'ouverture, à la fermeture et au maintien de la connexion. L'insertion de points de synchronisation est recommandée ; ils permettent, en cas de problèmes, de disposer d'un point précis à partir duquel l'échange pourra redémarrer.

Pour que deux systèmes puissent se comprendre, ils doivent utiliser le même système de représentation des données, et cette tâche est assurée par la couche ***présentation***. Elle se charge donc de la syntaxe des informations, de la représentation des données transférées entre applications.

Quant à la couche ***application***, elle fournit les fonctions nécessaires aux applications utilisateurs qui doivent accomplir des tâches de communication. Elle intègre les logiciels qui utilisent les ressources du réseau.

1.2.2. Le modèle TCP/IP

Dans les années 70, la défense américaine, DOD (Department Of Defense), devant le foisonnement de machines utilisant des protocoles de communication différents et incompatibles, décide de définir sa propre architecture. Cette architecture TCP/IP est à la source de l'Internet.

Contrairement au modèle OSI, le modèle TCP/IP ne comporte que quatre couches : la couche ***accès réseau***, la couche ***Internet***, la couche ***transport*** et la couche ***application***. Les deux principaux protocoles définis sont : TCP et IP.

Figure 1.04 : Le modèle TCP/IP

La couche ***accès réseau*** représente la connexion physique avec les câbles, les cartes réseau, les protocoles d'accès au réseau...

La couche ***Internet*** doit fournir une adresse logique pour l'interface physique. Cette couche fournit un mappage entre l'adresse physique et l'adresse logique grâce aux protocoles ARP (Address Resolution Protocol) et RARP (Reverse Address Resolution Protocol). Quant à ICMP (Internet Control Message Protocol), il s'occupe des problèmes et diagnostics associés au protocole IP. Cette couche gère également le routage des paquets entre les hôtes.

La couche ***transport*** définit la connexion entre deux hôtes. Deux protocoles sont associés à cette couche, le TCP et l'UDP. TCP est responsable du service de transmission fiable avec la fonction de détection et de correction d'erreurs. UDP est quant à lui, un protocole peu fiable, il est spécialement utilisé dans les applications n'exigeant pas la fiabilité de TCP, comme dans les applications temps réel.

La couche ***application*** renferme les protocoles d'application fournissant des services à l'utilisateur. Elle interface donc les applications utilisateurs avec la pile de protocole TCP/IP.

1.2.3. Comparaison entre le modèle TCP/IP et le modèle OSI

Les modèles OSI et TCP/IP sont tous les deux fondés sur le concept de piles de protocoles indépendants. De plus, les fonctionnalités des couches sont globalement les mêmes. Tous deux supposent également l'utilisation de la technologie de paquets, par opposition à la commutation de circuits de la téléphonie traditionnelle.

La commutation de circuits est une technique qui permet de relier, à la demande, deux ou plusieurs utilisateurs, qui auront l'usage exclusif de la liaison pendant la durée de la communication.

A propos de différences, on peut remarquer la chose suivante : les fonctionnements des différentes couches ne sont pas semblables d'un modèle à un autre, même si la dénomination est identique. Ensuite, une autre différence est liée au mode de connexion: orienté connexion ou sans connexion. Certes, les deux modes de connexion existent dans les deux modèles mais à des niveaux de couches différents. Pour le modèle OSI au niveau de la couche transport, seul le mode orienté connexion est disponible, mais au niveau de la couche réseau, nous pouvons avoir soit du mode orienté connexion ou du mode sans connexion. En revanche dans le modèle TCP/IP, les deux modes sont disponibles au niveau de la couche transport et la couche Internet ne dispose que du mode sans connexion. Le modèle TCP/IP a donc cet avantage par rapport au modèle OSI : les applications (qui utilisent directement la couche transport) ont véritablement le choix entre les deux modes de connexion.

De plus, TCP/IP intègre la couche présentation et la couche session dans sa couche application et les couches physiques et liaison de données OSI au sein d'une seule couche : la couche accès réseau. Enfin, TCP/IP semble plus simple, car il comporte moins de couches, ce qui fait sa réputation au niveau des architectures réseaux.

On classe généralement les protocoles en deux catégories selon le niveau de contrôle des données que l'on désire:

Les *protocoles orientés connexion*: Il s'agit des protocoles opérant un contrôle de transmission des données pendant une communication établie entre deux machines. Dans un tel schéma, la machine réceptrice envoie des accusés de réception lors de la communication, ainsi la machine émettrice est garante de la validité des données qu'elle envoie. Les données sont ainsi envoyées sous forme de flot TCP : TCP qui est un protocole orienté connexion.

Les *protocoles non orientés connexion*: Il s'agit d'un mode de communication dans lequel la machine émettrice envoie des données sans prévenir la machine réceptrice, et la machine réceptrice reçoit les données sans envoyer d'avis de réception à la première. Les données sont ainsi envoyées sous forme de blocs (datagrammes). UDP est un protocole non orienté connexion.

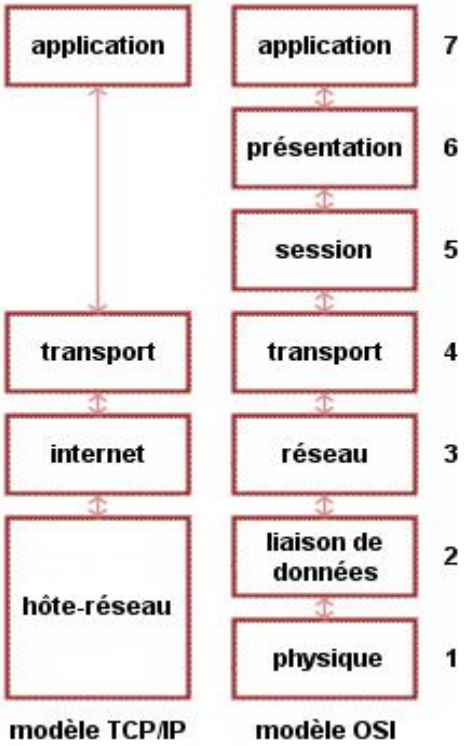

Figure 1.05 : Comparaison des fonctionnalités des couches du modèle OSI et celles des protocoles TCP/IP

1.2.4. Mécanismes d'encapsulation

Que ce soit dans le modèle OSI ou dans le modèle TCP/IP, les données utilisateurs sont toujours encapsulés à l'envoi et désencapsulés à partir des trains de bits au niveau de la réception. A chaque passage au niveau d'une couche donnée, il y a ajout d'informations relatives aux protocoles utilisées (en-tête/en queue) comme enveloppes de ces couches de façon que le contrôle soit plus aisé. Le processus d'encapsulation commence à la couche application et ne se termine qu'à la couche physique. Par contre, la désencapsulation commence à la couche physique et se termine à la couche application afin de restituer les données applicatives envoyées par l'émetteur.

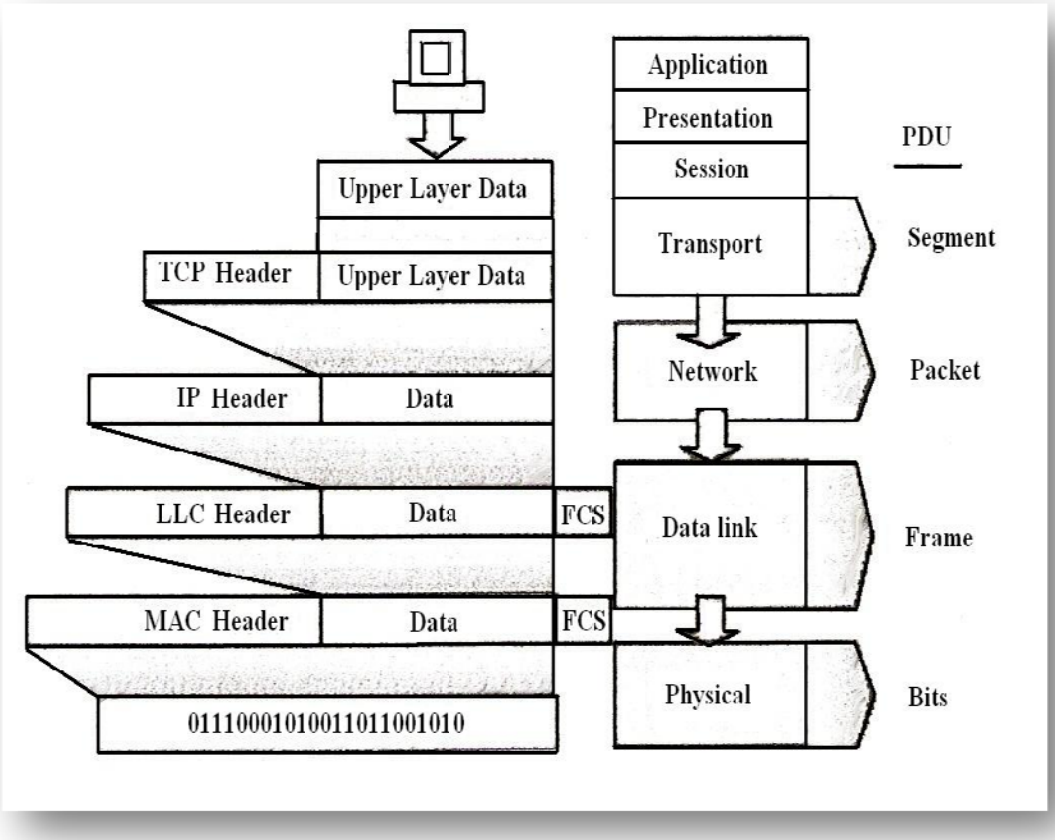

Figure 1.06 : Encapsulation des donnécs

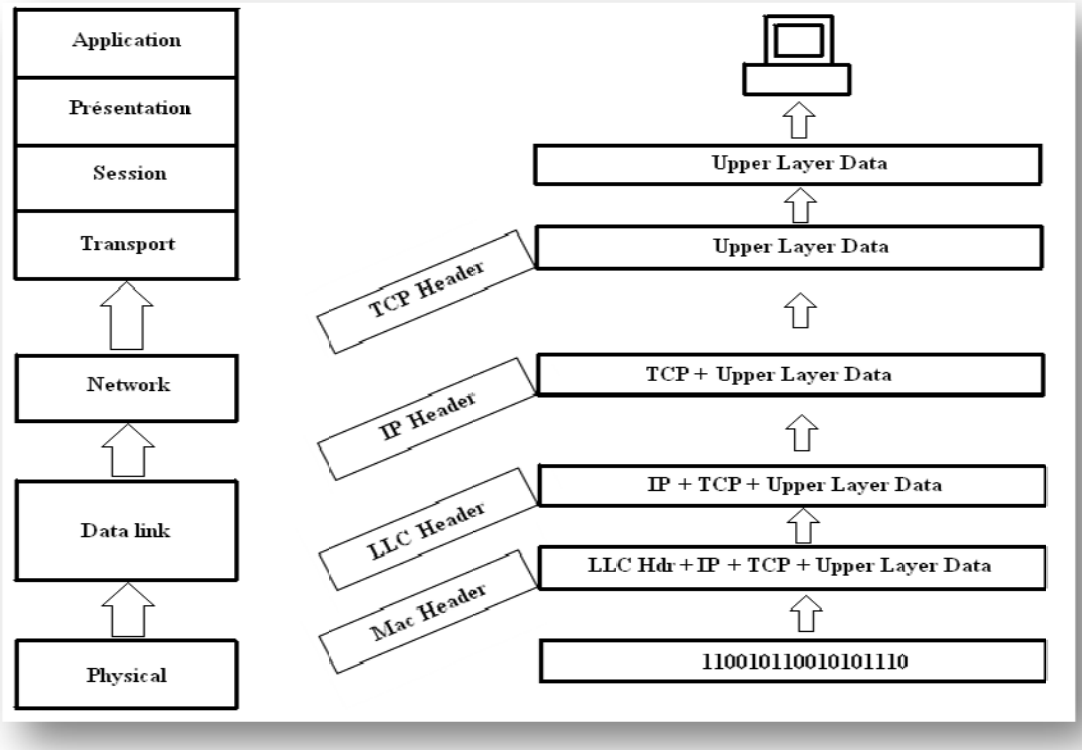

Figure 1.07 : Désencapsulation des données

1.3. Le protocole Internet [1] [2] [3] [6] [7] [8]

La couche Internet, similaire à la couche réseau du modèle OSI, assure le transport des données. Le rôle de cette couche est de trouver le meilleur chemin au sein d'un réseau. Les unités utilisent le système d'adressage de la couche réseau pour déterminer la destination des données durant leur déplacement. Le protocole de base de la couche réseau est le protocole IP.

1.3.1. Description du protocole IP

Comme son nom l'indique (Internet Protocol), le protocole IP a pour rôle de router le trafic à travers des réseaux. Il a été conçu pour réaliser l'interconnexion de réseaux informatiques et permettre ainsi les communications entre systèmes. Ce protocole assure la transmission des paquets de données, appelés datagrammes entre un ordinateur source et un ordinateur destination.

Par exemple, les applications qui tournent sur une machine cliente génèrent des messages qui doivent être envoyés sur une autre machine d'un autre réseau. IP reçoit ces messages de la couche transport et les envoie vers sa destination grâce à l'adressage IP.

Le protocole IP multiplexe les protocoles de la couche transport et a la faculté de détruire les paquets ayant transités trop longtemps sur le réseau. Il permet également de fragmenter et de rassembler de nouveau les fragments de données. Cependant, il n'effectue ni contrôle d'erreur, ni contrôle de flux.

Remarque :
Dans le modèle OSI, nous parlons plutôt de paquets au niveau de la couche réseau. En revanche, dans le modèle TCP/IP, l'unité transférée est le datagramme. Cependant, dans la suite de l'ouvrage, nous les mentionnerons indifféremment.

1.3.2. Principe de l'adressage des machines

Lorsque nous envoyons des données à travers l'Internet, les données ne sont pas envoyées de manière brute mais elles sont découpées en messages, puis en segments, en datagrammes et enfin en trames. Les datagrammes, outre l'information, sont constitués d'en-tête contenant l'adresse IP de l'expéditeur (votre ordinateur) et celle du destinataire (l'ordinateur que vous voulez atteindre), ainsi qu'un nombre de contrôle déterminé par l'information emballée dans le paquet : ce nombre de contrôles, communément appelé en-tête de total de contrôles ou « checksum », permet au destinataire de savoir si le datagramme IP a été "abîmé" pendant son transport. L'adressage joue donc un rôle très important dans ce processus.

1.3.2.1. Structure d'adresses IP

Chaque machine d'Internet possède une adresse IPv.4 représentée sur un entier de 32 bits, ce qui lui permet d'être identifiée de manière unique dans le réseau. L'adresse est constituée de deux parties: un identificateur de réseau (netid) et un identificateur de machine pour ce réseau (hostid). Il existe quatre classes d'adresses, chacune permettant de coder un nombre différent de réseau et de machines. Pour assurer l'unicité des numéros de réseau, les adresses Internet sont attribuées par un organisme central, l'InterNIC (pour Madagascar, il s'agit du NIC-mg).

Bien qu'IPv.6, la nouvelle génération du protocole IP existe déjà, IPv.4 reste largement le plus utilisé étant donnée que la saturation d'adressage ne semble pas encore être un problème majeur. Aussi, dans la suite de l'ouvrage, nous nous référerons au protocole IPv4.

IPv6 utilise un adressage utilisant huit groupes de quatre lettres hexadécimales séparés par « : ». Les enjeux majeurs de l'IPv6, outre l'extension de l'espace d'adressage sont : un traitement plus rapide grâce à un en-tête plus simplifié, la sécurité, la notion de flux (qualité de service)...

1.3.2.2. Classes d'adresses IP

L'adresse réseau est placée sur les bits de poids forts, alors que l'adresse de machine est calculée sur les bits de poids faibles. Il existe plusieurs classes d'adresses. On parle des classes A, B, C, D et E. Elles sont différenciées par les bits de poids forts qui les composent.

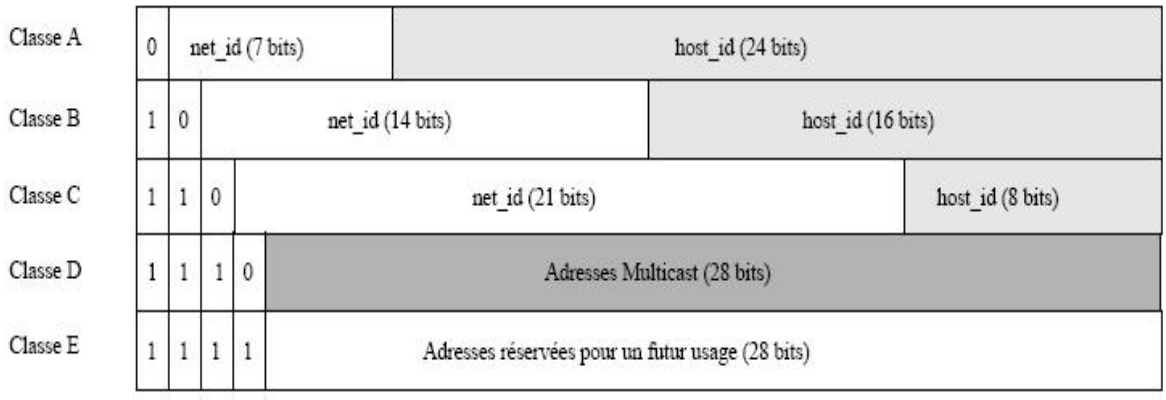

Figure 1.08 : Classes d'adresses IP

Une adresse IP est toujours de la forme X1.X2.X3.X4 (les Xi sont des blocs de 8 bits). La spécification du netid dépend de la classe. Dans le cas d'une classe A, la valeur de X1 permet de reconnaître ce réseau ; les X2, X3, X4 permettent de constituer des adresses individuelles. On

pourra donc adresser théoriquement 16.777.214 machines. Dans le cas d'une classe B, il est spécifié par XI et X2. On pourra alors adresser 65.534 machines. Une classe C fixe les valeurs de XI, X2, X3 pour le netid. On pourra donc adresser 254 machines. La classe D est une classe quelque peu différente, puisqu'elle est réservée à une utilisation particulière : le multicast. La classe E est quant à elle une classe non usitée à ce jour.

Le multicast ou multi diffusion est une technique utilisée par les protocoles spéciaux pour transmettre simultanément des messages à un groupe donné de nœuds différents.

Remarque :

Il y a des adresses spéciales que le public ne peut pas utiliser comme adresse d'identification :

- Adresse machine locale : adresse IP dont le champ réseau (netid) ne contient que des zéros;
- Adresses réseau : adresse IP dont la partie hostid ne comprend que des zéros; => la valeur zéro ne peut être attribuée à une machine réelle : 192.20.0.0 désigne le réseau de classe B
- hostid = 0 (=> tout à zéro), l'adresse est utilisée au démarrage du système afin de connaître l'adresse IP
- hostid = l (=> tout à 1) et netid = l (=> tout à 1) ou hostid = 0 (=> tout à zéro) et netid=0 (=> tout à zéro)

L'I.A.N.A a réservé les trois blocs d'adresses IP suivants pour l'adressage des réseaux privés :

- 1 adresse de classe A : 10.0.0.0 - 10.255.255.255

- 16 adresses de classe B : 172.16.0.0 - 172.31.255.255

- 255 adresses de classe C : 192.168.0.0 - 192.168.255.255

Classe d'adresse IP	*Minimum*	*Maximum*	*Nombre de réseaux*	*Nombre de nœuds*
A	1	126	127	16.777.214
B	128	191	16.383	65.534
C	192	223	2.097.151	254
D	224	239		
E	240	247		

Tableau 1.01: Comparaison entre les différentes classes d'adresse IP

1.3.3. Transport et format de datagrammes IP

1.3.3.1. Datagramme IP

La structure du datagramme IP sera détaillée ci-après :

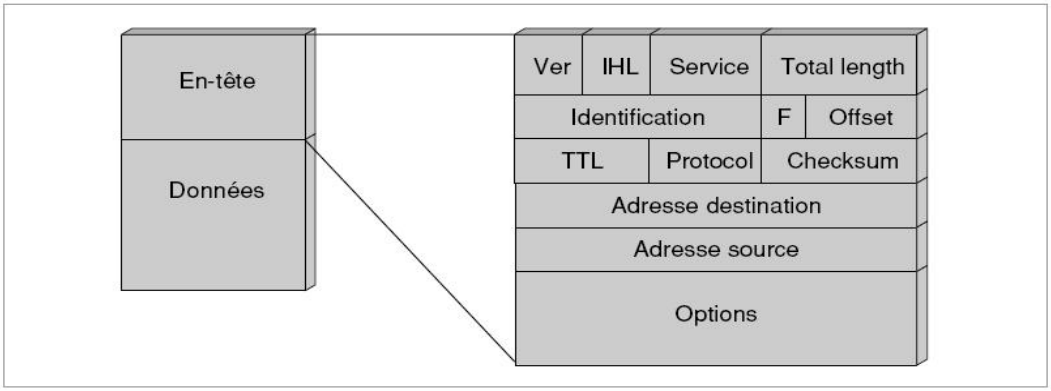

Figure 1.09 : Champs d'un datagramme IP

- *Ver (version) :* indique la version de protocole IP utilisée (4 bits).
- *IHL (IP header length - Longueur de l'en-tête IP ou LET)* : indique la longueur de l'en-tête du datagramme en mots de 32 bits (4 octets). Ce champ ne peut prendre une valeur en dessous de 5 pour être valide.
- *Service (type de service):* indique l'importance qui lui a été accordée par un protocole de couche supérieure donné (8 bits). Il indique au dispositif chargé de l'acheminement des datagrammes, le routeur, l'attitude à avoir vis-à-vis des datagrammes. Les bits de ce champ sont répartis comme suit :

  Bits 0 - 2 : Priorité.

  Bit 3 : 0 = Retard standard, 1 = Retard faible.

  Bit 4 : 0 = Débit standard, 1 = Haut débit.

  Bits 5 : 0 = Taux d'erreur standard 1 = Taux d'erreur faible.

  Bit 6 - 7 : Réservé

Priorité	R	D	T	0	0
0 2	3	4	5	6	7

Figure 1.10 : Le champ Type of service IP

- *Total length (longueur totale):* précise la longueur du paquet IP en entier, y compris les données et l'en-tête, en octets (16 bits). Ce champ ne permet de coder qu'une longueur de datagramme d'au plus 65.535 octets étant donné qu'il est codé sur 16 bits, comme indiqué précédemment.
- *Identification :* contient un nombre entier qui identifie le datagramme actuel (16 bits). Il contient une valeur entière utilisée pour identifier les fragments d'un datagramme. Ce champ doit être unique pour chaque nouveau datagramme.
- *F ou Flags (indicateurs):* un champ de 3 bits dont les 2 bits inférieurs contrôlent la fragmentation - un bit précise si le paquet peut être fragmenté et le second indique si le paquet est le dernier fragment d'une série de paquets fragmentés (3 bits).

  Bit 0 : réservé, doit être laissé à zéro

  Bit 1 : (AF) 0 = Fragmentation possible, 1 = Non fractionnable.

  Bit 2 : (DF) 0 = Dernier fragment, 1 = Fragment intermédiaire.

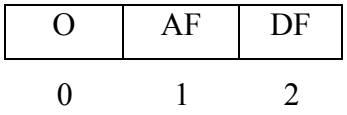

Figure 1.11 : Le champ flag ou indicateur

- *Offset ou Fragment Offset (décalage de fragment)* : ce champ sert à rassembler les fragments du datagramme (13 bits). Il indique le décalage du premier octet du fragment par rapport au datagramme complet. Cette position relative est mesurée en blocs de 8 octets (64 bits). Le décalage du premier fragment vaut zéro.
- *TTL (durée de vie)* : un compteur qui décroît graduellement, par incréments, jusqu'à zéro. A ce moment, le datagramme est supprimé, ce qui empêche les paquets d'être continuellement en boucle (8 bits)
- *Protocol (protocole) :* précise le protocole de couche supérieure qui recevra les paquets entrants après la fin du traitement IP (8 bits). Les différentes valeurs admises pour divers protocoles sont listées dans la RFC 1060. Exemples : 17 pour UDP, 6 pour TCP...

- *Checksum ou Header checksum (somme de contrôle d'en-tête)* : assure l'intégrité de l'en-tête IP (16 bits). Le Checksum doit être recalculé et vérifié en chaque point du réseau où l'en-tête est réinterprété puisque certains champs de l'en-tête sont modifiés (ex., durée de vie) pendant leur transit à travers le réseau.
- *Adresse source ou adresse d'origine* : indique le nœud émetteur (32 bits).
- *Adresse destination* : indique le nœud récepteur (32 bits).
- *Options ou IP Options :* cet élément permet au protocole IP de supporter différentes options, telles que la sécurité (longueur variable).

Les données contenues dans les datagrammes sont analysées (et éventuellement modifiées) par les routeurs permettant leur transit.

1.3.3.2. Routage des datagrammes sur Internet

Le ***routage*** désigne une technique permettant de déterminer le chemin emprunté par un message ou un paquet de données. C'est une méthode d'acheminement d'informations vers la bonne destination. Le routage s'opère au niveau de la couche 3 du modèle OSI, c'est-à-dire la couche réseau ou couche IP. Ainsi, les couches TCP et UDP restent donc à l'écart du concept de routage.

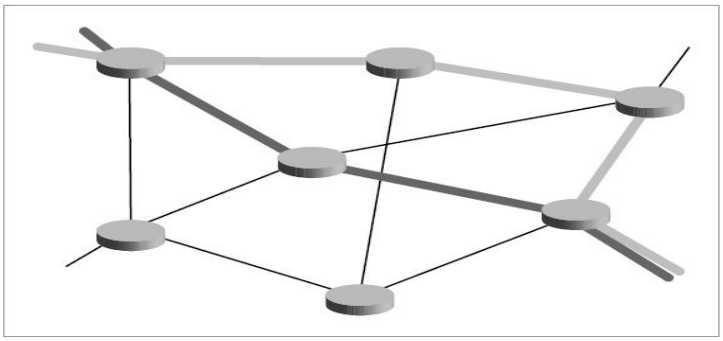

Figure 1.12 : Routage de paquets

Comme nous l'avons dit précédemment les données circulent à travers le réseau sous forme de paquets ou datagrammes IP. Afin d'atteindre leur destination, ils traversent maints routeurs. Les ***routeurs*** sont des unités d'interconnexion de réseaux qui fonctionnent au niveau de la couche 3 OSI (couche réseau) ou couche Internet du modèle DOD. Ils interconnectent des segments de réseau ou des réseaux entiers. Leur rôle consiste à acheminer les paquets de données entre les réseaux, en fonction des informations de la couche 3. Ils possèdent l'intelligence nécessaire pour déterminer le meilleur chemin de transmission des données sur le réseau.

Le routeur assure trois fonctions principales :

- permettre la communication entre des machines n'appartenant pas au même réseau.
- offrir un accès Internet à des utilisateurs d'ordinateurs se trouvant dans un réseau local.
- filtrer les datagrammes IP pouvant avoir accès à un réseau.

Il existe différents niveaux de routeurs, ceux-ci fonctionnent donc avec des protocoles différents:

- Les ***routeurs noyaux*** sont les routeurs principaux car ce sont eux qui relient les différents réseaux.
- Les ***routeurs externes*** permettent une liaison des réseaux autonomes entre eux. Ils fonctionnent avec un protocole appelé EGP (Exterior Gateway Protocol) qui évolue petit à petit en gardant la même appellation.
- Les ***routeurs internes*** permettent le routage des informations à l'intérieur d'un réseau autonome. Ils s'échangent des informations grâce à des protocoles appelés IGP (Interior Gateway Protocol).

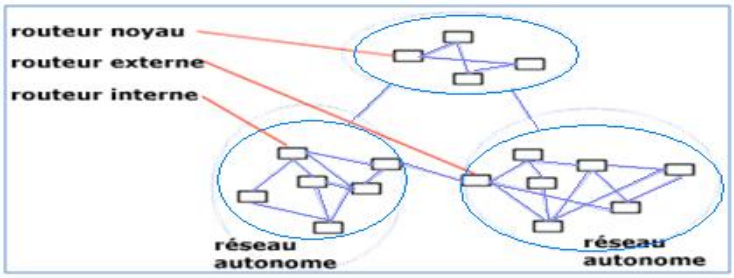

Figure 1.13 : Les différents niveaux de routeurs

Les routeurs possèdent des tables de routage leur permettant de choisir l'interface de sortie d'un datagramme à partir des informations stockées dans cette dernière. Les routeurs prennent également des décisions en fonction de la densité du trafic et du débit des liaisons (bande passante). La sélection du chemin permet à un routeur d'évaluer les chemins disponibles vers une destination donnée et de définir le meilleur chemin pour traiter un datagramme.

Dans le cas où le chemin correspondant figurant dans la table de routage est introuvable, IP supprime le datagramme.

Il existe plusieurs protocoles de routage comme le RIP (Routing Information Protocol), l'OSPF (Open Shortest Path First), l'IGRP (Interior Gateway Router Protocol), le BGP (Border Gateway Protocol), l'IDRP (Inter Domain Routing Protocol), l'IS-IS (Intermediate System to Intermediate System)...

Adresse de destination	*Adresse du prochain routeur directement accessible*	*Interface*
194.56.32.124	131.124.51.108	2
110.78.202.15	131.124.51.108	2
53.114.24.239	194.8.212.6	3
187.218.176.54	129.15.64.87	1

Tableau 1.02: Exemple de table de routage

Remarque :

Dans un réseau maillé, le routage des paquets (voir figure 1.14) nécessite des algorithmes complexes. Un nœud devrait connaître l'état de l'ensemble des autres nœuds avant de décider de la direction dans laquelle envoyer un paquet.

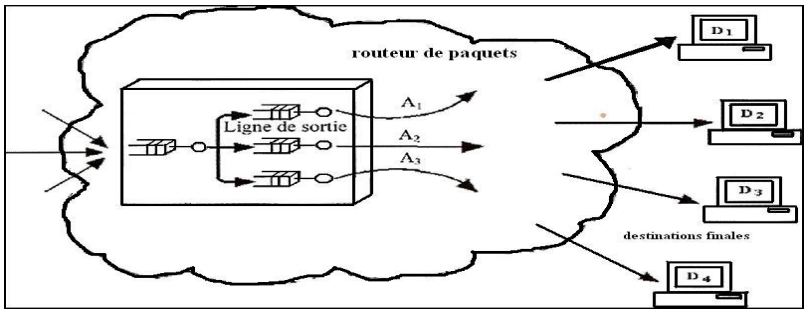

Figure 1.14 : Routage du paquet par le nœud de transfert

1.3.3.2.1. Le protocole RIP

1.3.3.2.1.1. Définitions

La méthode de routage à vecteur de distance détermine la direction (vecteur) et la distance vers n'importe quel lien de l'inter- réseau. Les algorithmes de routage à vecteur de distance transmettent des copies périodiques d'une table de routage d'un routeur à un autre. Ces mises à jour régulières entre les routeurs permettent de communiquer les modifications topologiques.

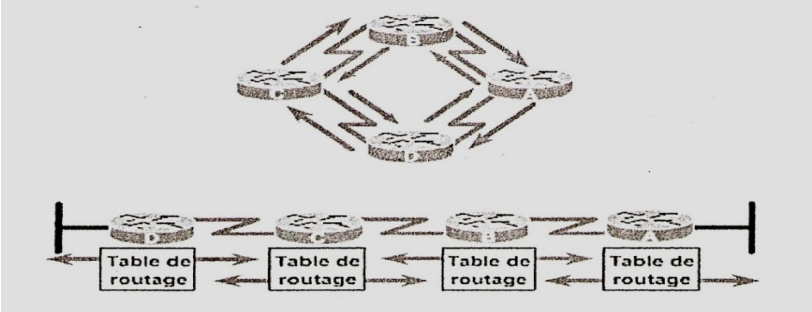

Figure 1.15 : Concepts de vecteur de distance

RIP est le protocole le plus fréquemment utilisé pour transférer des informations de routage entre des routeurs situés sur un même réseau. Ce protocole calcule la distance vers une destination donnée. Il permet aux routeurs qui l'utilisent de mettre à jour leurs tables de routage à des intervalles programmables, en général toutes les 30 secondes. Cependant, comme les routeurs établissent sans cesse des connexions avec leurs routeurs voisins, il en résulte une augmentation du trafic réseau.

RIP utilise le concept de " vecteur de distance ", qui s'appuie sur un algorithme de calcul du chemin le plus court dans un graphe. Le graphe est celui des routeurs, la longueur du chemin est établie en nombre de sauts ("hop"), ou métrique, entre la source et la destination, c'est-à-dire en comptant toutes les liaisons. Cette distance est exprimée comme un nombre entier variant entre 1 et 15 ; la valeur 16 est considérée comme l'infini et indique une mise à l'écart de la route.

1.3.3.2.1.2. Fonctionnement

Le protocole RIP permet aux routeurs de déterminer le chemin d'envoi des données, sur la base du concept de vecteur de distance. Lorsque les données passent par un routeur, elles exécutent un saut. Un chemin comportant quatre sauts indique que les données empruntant ce chemin doivent passer par quatre routeurs avant d'atteindre leur destination.

S'il existe plusieurs chemins vers une destination, le routeur recourt au protocole RIP pour sélectionner celui qui comporte le moins de sauts. Toutefois, comme le nombre de sauts est la seule métrique de routage utilisée par le protocole RIP pour déterminer le meilleur chemin, il n'est pas certain que le chemin sélectionné soit le plus rapide. Néanmoins, l'utilisation du protocole RIP est toujours largement répandue. Sa popularité est surtout due au fait qu'il a été l'un des premiers protocoles de routage à être développé.

Le protocole RIP pose un autre problème lorsque la destination choisie ne peut pas être atteinte parce qu'elle est trop éloignée. Avec le protocole RIP, les données peuvent exécuter 15 sauts au maximum. Par conséquent, il n'est pas possible d'atteindre un réseau de destination situé à une distance de plus de 15 routeurs.

1.3.3.2.2. Le protocole OSPF

Le protocole OSPF (Open Shortest Path First) a été conçu au sein de l'IETF à la fin des années 80. Il utilise plusieurs critères pour sélectionner le meilleur chemin vers une destination. Ces critères

incluent des métriques de coût qui tiennent compte notamment de la vitesse d'acheminement, du trafic, de la fiabilité et de la sécurité. OSPF est un protocole ouvert, c'est-à-dire qu'il n'a pas de copyright. II est beaucoup plus complexe que RIP mais ses performances et sa stabilité sont supérieures. Le protocole OSPF utilise une base de données distribuée qui permet de garder en mémoire l'état des liaisons. Ces informations forment une description de la topologie du réseau et de l'état de l'infrastructure.

Son avantage majeur réside dans le fait qu'il y a moins d'informations à faire circuler entre les routeurs, le protocole OSPF n'envoyant pas à ses voisins le nombre de sauts qui les sépare mais l'état de la liaison qui les sépare, ce qui permet d'avoir une meilleure bande passante disponible. Cependant, il est important de noter que l'algorithme utilisé par OSPF pour ses calculs de routes est extrêmement gourmand en ressources processeurs. En effet, plus une zone est importante et plus le nombre de calculs exécutés est conséquent. Les problèmes de performances sont donc accrus.

Pour éviter ces problèmes de performances, il existe une solution. En effet, lorsque des modifications interviennent sur le réseau, les routeurs ayant de nombreux voisins ont beaucoup de travail à accomplir. C'est pourquoi il est conseillé de limiter à cinquante le nombre de routeurs par zone.

Remarque :

Lorsqu'un algorithme de routage met à jour une table de routage, son principal objectif est de déterminer les meilleures informations à présenter dans la table. Chaque algorithme de routage interprète à sa façon les meilleures informations qui peuvent être : la bande passante, le délai, la charge (la quantité de trafic sur une ressource réseau), la fiabilité, le nombre de sauts. Ces informations constituent la *métrique de routage* ou *métrique de coût***.** En règle générale, *plus le nombre est petit, meilleur est le chemin.*

1.3.3.2.3. Protocole IGRP

Le protocole IGRP (Interior Gateway Router Protocol) a été défini par la société Cisco pour ses routeurs. En effet, alors qu'elle utilisait presque exclusivement à ses débuts le protocole RIP, cette compagnie a été amenée à mettre en place un routage plus performant, l'IGRP, qui n'est autre qu'une version améliorée de RIP. Elle intègre le routage multichemin, la gestion des routes par défaut, la diffusion de l'information toutes les 90 secondes au lieu de toutes les 30 secondes, la détection des bouclages, etc. Ce protocole a lui-même été étendu par une meilleure protection contre les boucles : il s'agit du protocole EIGRP « Extended IGRP ».

1.3.3.2.4. Protocole IS-IS

Le protocole IS-IS (Intermediate System to Intermediate System) a été principalement développé par l'ISO; il décrit un routage hiérarchique fondé sur la décomposition des réseaux de communication en domaines. Dans un domaine, les différents nœuds indiquent leur état aux routeurs IS-IS afférents. Les communications interdomaines sont effectuées par un routage vers un point d'accès au domaine, déterminé par les routeurs chargés des communications externes au domaine.

1.3.3.3. Différentes phases de routage IP

Les données utilisateurs se présentant dans le nœud, pour être livrées à sa destination, doivent passer par différentes phases.

1.3.3.3.1. Phase d'encapsulation des données

Lors d'une transmission, les données traversent chacune des couches de haut en bas au niveau de la machine émettrice. A chaque couche, une information est ajoutée au paquet de données, il s'agit d'un en-tête, ou d'un en-queue, ensemble de données de supervision qui garantit la transmission. Au niveau de la machine réceptrice, c'est l'opération inverse. Lors du passage dans chaque couche, l'en-tête, ou 1' en-queue est lu, puis supprimé. Ainsi, à la réception, dans la couche de même niveau que la couche émettrice, le message redevient dans son état originel.

Ainsi, à chaque couche, le paquet de données change d'aspect et les appellations changent :

- le paquet de données est appelé ***message*** au niveau de la couche application,
- le message est ensuite encapsulé sous forme de ***segment*** dans la couche transport,
- le segment une fois encapsulé dans la couche Internet prend le nom de ***datagramme,***
- le nom ***trame*** est utilisé au niveau de la couche accès réseau.

1.3.3.3.2. Phase de fragmentation des datagrammes IP

Dans les réseaux à transfert de paquets, les données des utilisateurs sont découpées ou fragmentées en paquets, ce qui facilite la retransmission.

Fragmenter un datagramme revient à le diviser en plusieurs morceaux. Chaque morceau a le même format que le datagramme de départ. Chaque nouveau fragment a un en-tête, qui reprend la plupart des informations de l'en-tête d'origine et il doit tenir dans une seule trame.

Sur Internet, dès qu'un datagramme a été fragmenté, les fragments sont transmis indépendamment les uns des autres jusqu'à leur destination, où ils doivent être réassemblés. Chaque datagramme fragmenté porte un numéro de séquencement pour que le récepteur puisse les réassembler dans le bon ordre.

Si un des fragments est perdu, le datagramme ne peut pas être réassemblé et les autres fragments doivent être détruits sans être traités. La probabilité de perte d'un datagramme augmente avec la fragmentation.

Sur toute machine ou passerelle mettant en œuvre TCP/IP, une unité maximale de transfert (Maximum Transfer Unit ou MTU) définit la taille maximale d'un datagramme véhiculé sur le réseau physique correspondant.

Lorsque le datagramme est routé vers un réseau physique dont le MTU est supérieur au MTU courant, la passerelle route les datagrammes tels quels. Par contre, si le MTU est plus petit que le MTU courant, la passerelle fragmente le datagramme en un certain nombre de fragments, véhiculés par autant de trames sur le réseau physique correspondant.

Le destinataire final reconstitue le datagramme initial à partir de l'ensemble des fragments reçus, la taille de ces fragments correspond au plus petit MTU emprunté sur le réseau.

La fragmentation d'un datagramme se fait au niveau des routeurs, c'est-à-dire lors de la transition d'un réseau dont le MTU est important à un réseau dont le MTU est plus faible.

Figure 1.16 : Fragmentation d'un datagramme dans un routeur

Chaque type de réseau a son MTU. Voici quelques exemples de MTU :

Type de réseau	*MTU (en octets)*
Arpanet	1000
Ethernet	1500
FDDI	4470

Tableau 1.03: Exemples de MTU

Résumé :

Pour gérer la fragmentation, les champs suivants de l'en-tête du datagramme IP sont utilisés : identification, indicateurs, longueur totale et le décalage de fragment.

Pour réassembler les fragments d'un datagramme Internet, le module destination réassemble les datagrammes dont les valeurs des quatre champs suivants sont identiques : Identification, Adresse IP source, Adresse IP destination et Protocole.

La recombinaison est réalisée en replaçant la portion de données contenue dans chaque fragment dans un tampon à la position relative indiquée par le champ "Fragment Offset" lu dans l'entête. Le dernier fragment aura son bit "Dernier fragment" à zéro.

1.3.3.3.3. Phase de lecture

Dans le routeur, le processus de couche réseau examine l'en-tête du paquet entrant afin de déterminer le réseau de destination, il consulte ensuite la table de routage qui associe les réseaux aux interfaces sortantes. Le paquet est de nouveau encapsulé dans la trame de liaison de données appropriée à l'interface sélectionnée, puis il est placé en file d'attente en vue d'être transmis au nœud suivant le chemin.

1.3.3.3.4. Phase de routage

Dans la phase de routage, deux cas peuvent se présenter.

1.3.3.3.4.1. Routage direct

C'est le cas où les deux machines qui veulent communiquer sont rattachées au même réseau, elles ont donc le même numéro de réseau IP. Il peut s'agir de deux hôtes, ou d'un routeur et d'un hôte. Il suffit donc de déterminer l'adresse physique destinataire et d'encapsuler le datagramme dans une trame avant de l'envoyer sur le réseau.

1.3.3.3.4.2. Routage indirect

Dans ce cas, le routage est plus complexe parce qu'il faut déterminer le routeur auquel les datagrammes doivent être envoyés. Ces derniers peuvent ainsi être transmis de routeur en routeur, jusqu'à ce qu'ils atteignent l'hôte destinataire. La fonction de routage se fonde principalement sur les tables de routage.

Grâce à la table, le routeur, connaissant l'adresse du destinataire encapsulé dans le message, est capable de savoir sur quelle interface envoyer le message (cela revient à connaître quelle carte réseau utiliser), et à quel routeur directement accessible sur le réseau auquel cette carte est connectée, remettre le datagramme. Ce mécanisme consistant à ne connaître que l'adresse du prochain routeur menant à la destination est appelé routage par sauts successifs.

Remarque :

Cependant, il se peut que le destinataire appartienne à un réseau non référencé dans la table de routage. Dans ce cas, le routeur utilise un *routeur par défaut* (appelé aussi *passerelle par défaut).*

Le message est ainsi remis de routeur en routeur par sauts successifs, jusqu'à ce que le destinataire appartienne à un réseau directement connecté à un routeur. Celui-ci remet alors directement le message à la machine visée.

1.3.3.4. Détermination des distances des chemins de réseau à l'aide de métriques

La table de routage est mise à jour dans un intervalle de temps régulier selon les algorithmes utilisés (à chaque 30 s pour RIP). En outre, cette mise à jour est aussi effectuée dès qu'une modification topologique se produit à savoir : pannes des équipements de routage (routeurs ou lignes de transmissions), ajout ou suppression des hôtes ou autres modifications.

Lorsqu'un algorithme de routage met à jour une table de routage, son principal objectif est de déterminer les *meilleures informations* à présenter dans la table. L'algorithme génère un nombre, appelé "valeur métrique", pour chaque chemin du réseau.

Nous pouvons calculer les métriques en fonction d'une seule caractéristique de chemin ou en combinant plusieurs caractéristiques parmi lesquelles :

- le *débit* : le débit d'une liaison, mesuré en bits par seconde (en règle générale, une liaison Ethernet à 10 Mbits/s est préférable à une ligne louée de 64 Kbits/s).
- le *délai* : le temps nécessaire à l'acheminement d'un paquet, pour chaque liaison, de la source à la destination.
- la *charge* : la quantité de trafic sur une ressource réseau telle qu'un routeur ou une liaison.
- la *fiabilité* : cette notion indique généralement le taux d'erreurs sur chaque liaison du réseau.
- le *nombre de sauts* : le nombre de routeurs par lesquels un paquet doit passer avant d'arriver à destination.
- le *coût* : une valeur arbitraire, généralement basée sur la bande passante, une dépense monétaire ou une autre mesure, attribuée à un lien par un administrateur réseau.

1.3.4. Résolution d'adresses logiques

Toute interface du réseau est modélisée par une adresse logique IP. Or, pour la transmission sur l'interface physique, cette information n'est pas accessible à cause de l'encapsulation dans la trame qui contient des adresses matérielles, adresses MAC (Media Access Control), qui sont généralement attribuées par les fabricants de la carte réseau et codées en dur sur la carte. Cela suppose donc qu'il faut connaître au préalable l'adresse physique du destinataire. Ce qui s'avère fastidieux pour l'être humain. De plus, nous ne connaissons, au mieux, que l'adresse IP de ce dernier. Nous devons ainsi mettre en place des techniques de correspondances d'adresses. Ces techniques sont modélisées par les protocoles ARP et RARP.

1.3.4.1. Le protocole ARP

Le protocole ARP (Address Resolution Protocol) permet à un ordinateur de trouver l'adresse MAC de l'ordinateur associé à une adresse IP. La requête ARP est envoyée avec une adresse matérielle de diffusion, ou adresse MAC de diffusion, ayant le format FF-FF-FF-FF-FF-FF.

Comme un paquet de données doit contenir une adresse MAC de destination et une adresse IP de destination, aussi, si l'une ou l'autre manque, les données ne sont pas transmises depuis la couche 3 aux couches supérieures. Ainsi, les adresses MAC et IP se contrôlent et s'équilibrent mutuellement. Une fois que les équipements ont déterminé les adresses IP des équipements de destination, ils peuvent ajouter les adresses MAC de destination aux paquets de données.

Les tables ARP, mappent les adresses IP avec les adresses MAC correspondantes. Ces tables sont des sections de mémoire RAM dans lesquelles la mémoire cache est mise à jour automatiquement dans chaque équipement. Chaque ordinateur du réseau met à jour sa propre table ARP. Chaque fois qu'un équipement souhaite envoyer des données sur le réseau, il utilise les informations contenues dans sa table ARP.

1.3.4.2. Le protocole RARP

Le protocole RARP (Reverse Address Resolution Protocol) est beaucoup moins utilisé, il signifie *Protocole ARP inversé,* il s'agit donc d'une sorte d'annuaire inversé des adresses logiques et physiques. L'utilisateur ne connaît donc pas l'adresse IP de celui à qui il veut correspondre. En réalité le protocole RARP est essentiellement utilisé pour les stations de travail n'ayant pas de disque dur et souhaitant connaître leur adresse physique...

Les principaux problèmes de RARP sont : qu'un serveur ne peut servir qu'un LAN et il nécessite beaucoup de temps d'administration pour maintenir des tables importantes dans les serveurs. Pour pallier à ces deux problèmes d'administration, le protocole RARP peut être remplacé par le protocole DRARP (Dynamic RARP), qui en est une version dynamique.

1.4. Le protocole IPv.6 [3] [9] [10]

Le protocole IPv.6 améliore la qualité de service par rapport à l'ancienne version IP. Voici une figure (figure 1.17) permettant de voir son architecture :

<table>
<tr><td>Version 6</td><td>Priorité</td><td colspan="2">Etiquette de flot</td></tr>
<tr><td colspan="2">Longueur des données</td><td>En-tête suivant</td><td>Nombre de nœuds traversés</td></tr>
<tr><td colspan="4">Adresse de la source sur 16 octets</td></tr>
<tr><td colspan="4">Adresse de la destination sur 16 octets</td></tr>
</table>

Figure 1.17 : Architecture d'un paquet IPv.6

Ces améliorations se traduisent par l'utilisation de deux champs spécifiques :

- Le champ *Priorité*, qui indique un niveau de priorité qui permettra de traiter les paquets plus ou moins rapidement dans les nœuds du réseau. Les principales valeurs sont illustrées dans le tableau 1.04 ci-dessous :

0	Pas de priorité particulière
1	Trafic de base (news)
2	Transfert de données sans contrainte temporelle (e-mail)
3	Réservé pour le futur
4	Transfert en bloc avec attente du récepteur (transfert de fichiers)
5	Réservé pour le futur
6	Trafic interactif (rlogin, terminal virtuel)
7	Trafic pour le contrôle (routage, contrôle de flux)

Tableau 1.04 : Valeur des différentes priorités

- Le champ *Etiquette de flot*, qui permet d'indiquer la qualité de service associée aux informations transportées. Dans le cas du transport de la voix, cette indication donne la possibilité aux routeurs de prendre des décisions, afin de supplanter les contraintes temps réels.

1.5. Les protocoles de la couche « Transport » [1] [3] [11] [12]

1.5.1. Le protocole TCP

TCP, acronyme de « Transport Control Protocol », est un protocole défini dans RFC 793 conçu pour être utilisé avec IP. TCP apporte des services élaborés en termes de transmission de données entre des machines raccordés sur un réseau.

Les fonctions principales de TCP sont :

- l'établissement d'une connexion fiable « full duplex ».
- le formatage des données en segments où chaque segment est numéroté et sa réception est confirmée par le récepteur.
- le contrôle des erreurs (détection d'erreurs et reprise).
- le contrôle de flux pour ne pas déborder le récepteur en imposant à l'émetteur une limite lors de la transmission d'un segment.
- le bon séquencement des segments reçus (en utilisant leur numéro)
- l'acheminement des données normales ou urgentes.

La connexion TCP s'explique par l'établissement d'un chemin entre les points échangeant des informations appelé : « *circuit virtuel* ». On dit dans un certain temps que TCP fonctionne en mode « *orienté-connexion* », c'est à dire l'une des applications doit effectuer un appel que l'autre doit accepter avant que la connexion soit établie et que le transfert puisse débuter.

Conceptuellement, cinq éléments doivent-être réunis pour établir un circuit virtuel :

- le protocole
- l'adresse IP de la machine émettrice
- le port local (le numéro de port de l'application locale)
- l'adresse IP de la machine distante
- le port distant (le numéro de port de l'application distante).

Ces points réunis, la connexion peut s'établir. Comme la connexion TCP est *«full-duplex* », les données circulent indépendamment dans un sens et dans l'autre. A chaque envoi de paquets par l'émetteur, le récepteur doit acquitter chaque paquet par un accusé ACK pour s'assurer que les données sont bien reçues.

1.5.2. Le protocole UDP

UDP, acronyme de « User Datagram Protocol », est un protocole défini dans la RFC 768. Avec plus de simplicité que TCP, UDP est également conçu pour être utilisé avec IP. UDP permet de fournir une communication par paquets entre deux processus dans un environnement réseau étendu. Le protocole IP lui sert de support de base à la communication.

Les fonctions principales du protocole UDP sont :

- le transport non-fiable en mode non connecté.
- l'envoi d'un message court d'une application vers une autre avec un mécanisme simple.
- le mécanisme de gestion des ports.

UDP est caractérisé par l'absence d'un accusé de réception au niveau du protocole pour apporter un moyen de contrôle sur l'acheminement des données. L'émission des messages se fait également sans garantie ce qui signifie qu'UDP ne garantit pas la livraison et la vérification de la mise en séquence des datagrammes. L'hypothèse de la non-fiabilité de la transmission avec UDP est alors vérifiée. En même temps, le contrôle de flux limitant les données volumineuses reçues par le récepteur n'existe pas avec le protocole UDP.

On utilise UDP pour un transport rapide où la perte occasionnelle des paquets n'est pas critique.

1.5.3. Comparaison des protocoles TCP et UDP

En se référant aux caractéristiques de ces deux protocoles ; on peut déduire que contrairement à TCP qui ne tolère aucune erreur en transmettant et en retransmettant plusieurs fois jusqu'à ce qu'ils soient corrects à l'arrivée, les impératifs de la transmission en temps réel rendent UDP plus adapté aux programmes qui présentent des exigences en temps réel (la voix, la vidéo,...). D'autant plus, UDP tolère les erreurs donc il est adapté aux situations où la perte occasionnelle des paquets n'est pas critique.

Exemple : une perte de quelques pixels dans une image est tolérable vue que cela ne soit pas perçue par l'œil humain.

Le tableau suivant établit une comparaison des différents modes de gestion des communications suivant que le protocole UDP ou TCP soit utilisé ou non pour le transport des données.

UDP	*TCP*
Service sans connexion : aucune session n'est établie entre les machines.	Service orienté connexion : une session est établie entre les machines.
UDP ne garantit pas la réception des données de livraison ou de séquence (pas d'accusé de réception)	TCP garantit la livraison à l'aide des accusés de réception et la livraison des données en séquence.
UDP tolère les pertes dans la transmission des données.	TCP ne tolère aucune perte dans la transmission des données.
UDP est rapide et présente des exigences de faible délai et peut prendre en charge des communications point à point et point à multipoint.	TCP est plus lent et présente des exigences de délai plus élevé. Il ne peut prendre en charge que les communications point à point.

Tableau 1.05 : Comparaison de TCP et UDP

1.6. Conclusion

On entend souvent qu'avant le déploiement de l'IPv.6, la prochaine version du protocole IP, il sera impossible de déployer à grande échelle des applications de voix sur IP à cause de manque d'adresses. Ceci est totalement faux. La sophistication des outils de routage disponibles en IPv.4 et l'utilisation combinée de passerelles applicatives de voix sur IP rendent possible l'utilisation d'un ensemble limité d'adresses pour fournir un service à un ensemble potentiellement illimité d'usagers. A cause de la taille plus faible des paquets de données IPv.4, la latence sur de tels réseaux, particulièrement sur les liens d'accès, sera meilleure que sur un réseau IPv.6 équivalent, et l'efficacité du protocole IPv.4 pour de tels paquets est également bien meilleure.

Enfin, UDP est plus adapté aux programmes qui présentent des exigences en temps réel (voix sur IP, téléphonie sur IP, …).

CHAPITRE 2 TELEPHONIE SUR IP

2.1. Présentation de la Voix sur IP (VoIP) [13] [14] [15] [18] [22]

2.1.1. Introduction à la VoIP

2.1.1.1. Qu'est-ce que la VoIP ?

La voix sur IP constitue actuellement l'évolution la plus importante du domaine des Télécommunications. La VoIP correspond à l'ensemble des technologies permettant la transmission d'une conversation vocale sur un réseau de protocole IP. La communication par VoIP est plus flexible et permet de déployer simultanément d'autres services que le transport de la voix, comme par exemple la vidéo ou les données (ce qu'on entend par triple play : voix, données et vidéo). Avec la technologie VoIP, la voix est donc numérisée, compressée et envoyée sous forme de paquets sur un réseau fonctionnant avec le protocole IP. Les données reçues sont décompressées et converties en voix audible.

La VoIP peut être définie comme la possibilité d'effectuer des appels téléphoniques et d'envoyer des fichiers à travers un réseau de données à base du protocole IP à une qualité acceptable et à des prix abordables voire symboliques. On peut ainsi communiquer, par écran interposé, n'importe où dans le monde sans aucune considération financière puisque le prix est toujours celui d'une communication locale. Nous assistons à une révolution au niveau des tarifs qui s'annoncent démesurément bas.

Le but de la VoIP est la convergence voix/données autour d'un protocole unique IPv.4 (éventuellement IPv.6). En effet, la VoIP se base sur la même architecture que l'Internet et utilise les mêmes infrastructures. En intégrant la voix et les données, la VoIP simplifie l'administration des réseaux car tout est centralisé dans un unique réseau (réseau LAN) et non deux (réseau téléphonique et réseau LAN). Elle procure ainsi des facilités pour le développement d'applications utilisant la voix et les données. Nous pourrons par exemple bénéficier des applications coopératives ; les utilisateurs peuvent dialoguer avec un correspondant, tout en consultant simultanément les mêmes données sur leur ordinateur, grâce aux infrastructures Internet. Le développement actuel des technologies IP devrait faciliter le travail de coopération multimédia.

Ainsi, au contraire des téléphones analogiques filaires RTC dépendant de centraux téléphoniques dédiés, la voix sur IP permet le transport de conversations téléphoniques sur tout réseau numérique ou analogique acceptant le protocole TCP/IP (Ethernet, RMS, PPP,...).

2.1.1.2. Contexte technologique de la VoIP

Si à l'origine le protocole IP a été conçu pour fournir des services de type « best effort », les réseaux IP représentent désormais une part importante des infrastructures de télécommunications et ont vocation à transporter de nombreux services avec leurs contraintes de disponibilité (applications distantes, voix, vidéo, télévision, etc.).

Avant 1996, les solutions de voix sur IP reposaient sur des architectures propriétaires. Ces solutions présentaient des défauts parmi lesquels : le manque d'interopérabilité des équipements, l'impossibilité de raccordement au réseau public (seuls les ordinateurs pouvaient communiquer entre eux), l'absence d'architecture générale pour la connexion de n'importe quel type de terminal, chaque architecture était définie pour deux équipements d'extrémité spécifiques et ne pouvait pas interopérer avec d'autres architectures d'équipements.

2.1.1.3. De la voix à l'IP en 8 étapes

La technologie de la voix sur IP peut être découpée en 8 grandes étapes : acquisition, numérisation, compression, habillage des en-têtes, émission et transport, réception, conversion numérique/analogique, et restitution.

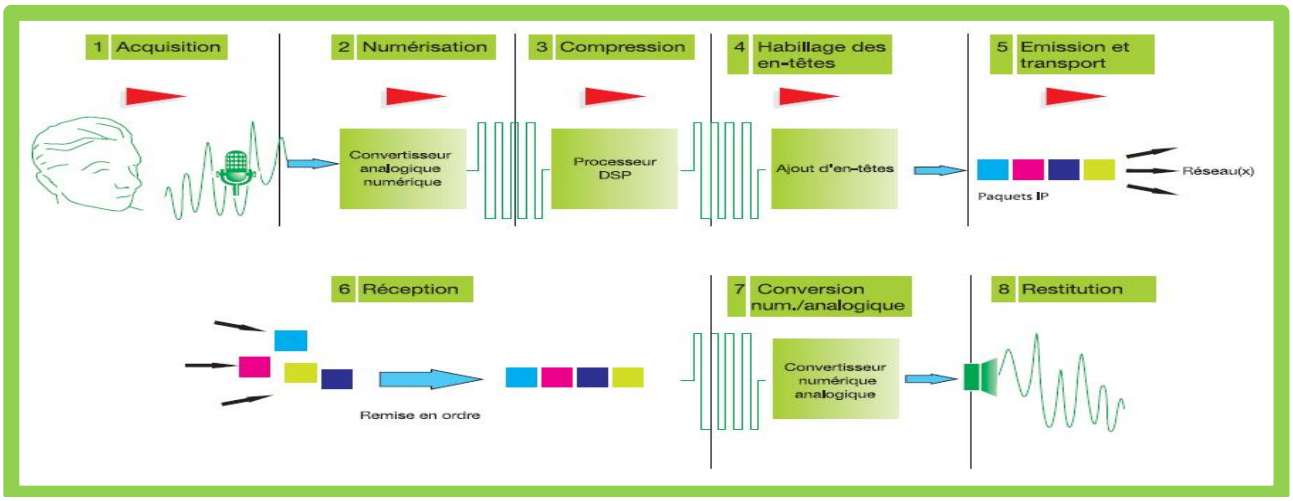

Figure 2.01 : Principe de la VoIP

2.1.1.4. Les différentes architectures

Différents scénarios de mise en œuvre du transport de la voix sur réseaux de données sont envisageables pour les entreprises ou les particuliers. Les scénarios sont les suivants : Communication de PC à PC, de PC à téléphone, de téléphone à téléphone.

2.1.1.4.1. De PC à PC

Cette solution est appelée la solution tout-IP comme illustrée sur la figure 2.02. Les terminaux sont conformes à la norme H.323. Dans ce cas, ce sont eux qui jouent le rôle de passerelle et permettent donc la numérisation et le codage du signal vocal entrant. Ces PC utilisent des logiciels de communication supportant également la norme H.323 tels que Netmeeting et Netphone. Afin que les différents PC puissent communiquer entre eux, il est nécessaire d'utiliser un serveur d'annuaire. Lorsqu'un terminal se connecte sur le réseau de données, il s'enregistre sur le serveur informant ainsi toutes les autres machines de sa présence sur le réseau. Il existe aujourd'hui un certain nombre de serveurs d'annuaire, le serveur de Netmeeting est ILS (Internet Locator Server). Le serveur d'annuaire a également pour rôle de faire la correspondance entre une adresse logique (exemple : e-mail, identifiant, numéro de téléphone) et une adresse identifiant la machine. Dans ce cas, la qualité de service pour la gestion de la voix est assurée par ces protocoles. Une adresse de niveau 3 est alors utilisée uniquement pour l'établissement de la communication réseau. L'échange voix se fera après cet établissement. On considère que la couche inférieure n'offre pas de qualité de service suffisante pour le transport de la voix. Nous pouvons distinguer 2 types de réseaux de données utilisant le protocole IP pour gérer la voix : Internet et Intranet.

Internet - Ici, le réseau de données est l'Internet. La principale limitation est due à l'incapacité du réseau Internet de garantir un transfert fiable et rapide de l'information, provoquant une qualité d'écoute assez faible. Le principal avantage de l'Internet est le faible coût pour les utilisateurs. Dans ce cas les utilisateurs sont des particuliers.

Intranet - Dans ce cas le réseau Intranet de l'entreprise vient se substituer à Internet et la limitation précédente est en partie levée. La gestion des routeurs et le dimensionnement du réseau sont alors sous le contrôle d'un seul opérateur et de l'entreprise elle-même, ceci permettant ainsi de mieux gérer les problèmes de congestion et de file d'attente.

Figure 2.02 : La solution Tout-IP

2.1.1.4.2. De téléphone à téléphone

Tout le monde n'étant pas forcément équipé d'un PC, une évolution importante est apparue lorsque le codage et le décodage de la voix ont été mis en œuvre sur des serveurs (et non plus sur des terminaux PC) assurant une fonction avec les réseaux téléphoniques classiques.

2.1.1.4.2.1. Téléphone à téléphone sur réseau de données via le RTC

Cette configuration permet aux utilisateurs équipés de terminaux téléphoniques (ou de télécopieurs) de bénéficier de service voix à bas prix. L'opérateur du service constitue un réseau dont les passerelles forment des points de présence locaux comme illustré sur la figure 2.03 ci-dessous. Elles sont reliées entre elles par un réseau sous contrôle de l'opérateur permettant le respect des débits et des délais. Les points critiques concernant l'utilisateur final sont :

- le rapport qualité-complexité/coût. En effet dans les produits actuels, il n'est pas rare dans les passerelles de numéroter plus de 30 chiffres.
- la facturation du service

Figure 2.03 : Téléphone à téléphone sur réseau de données via RTC

2.1.1.4.2.2. Téléphone à téléphone via réseau de données

Dans ce cas, le réseau de données correspond à un réseau privé fourni par l'opérateur. Ce type d'offre correspond à la transposition des services voix sur le réseau de données de l'entreprise. Les passerelles sont raccordées aux PABX des entreprises (certains constructeurs prévoient même une offre intégrée aux PABX) en leur permettant d'écouler leur trafic téléphonique inter site et leurs données sur un même réseau (voir figure 2.04). Le principal intérêt reste donc le gain économique. Là encore, les principales difficultés techniques concernent la perception de la qualité par rapport au coût engendré par la mise en place de mécanismes de contrôle de flux, la gestion de plans de numérotation cohérents et la traversée des pare-feu.

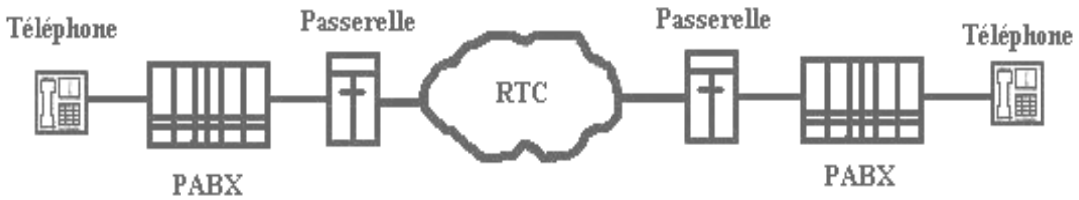

Figure 2.04 : Téléphone à téléphone via réseau de données

2.1.1.4.3. De PC à téléphone

Cette architecture est une combinaison des deux précédentes. Dans ce cas, on se rend compte qu'il est possible de faire interagir des équipements différents confirmant ainsi la notion de fusion de l'informatique et de la téléphonie (voir figure 2.05).

Nous avons présenté les différentes architectures possibles, mais la principale application utilisant le transport de la voix sur des réseaux de données est l'interconnexion de PABX d'une même entreprise en utilisant son propre réseau.

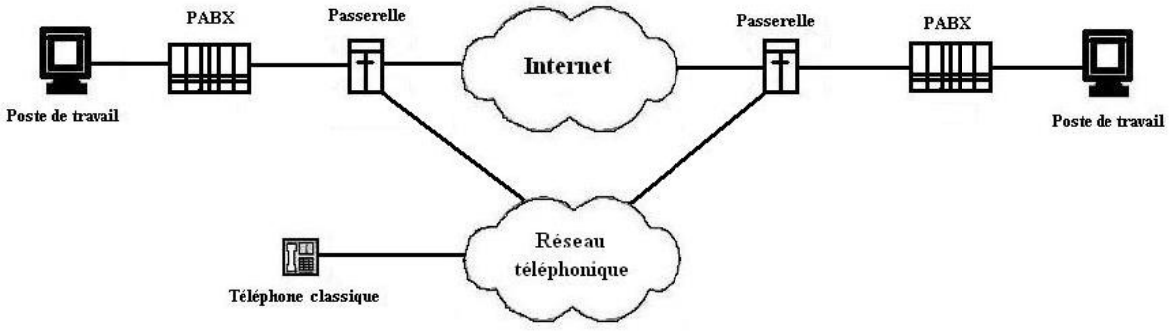

Figure 2.05 : Connexion téléphone à PC

2.1.2. Les codages de la VoIP

2.1.2.1. Introduction

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).

Le choix d'un codeur est un processus en général très long. Avant de sélectionner un codeur, l'UIT en teste la qualité subjective par le score ***MOS*** (Mean Opinion Score) qui est la note moyenne attribuée par un panel d'auditeurs écoutant des séquences enregistrées dans diverses conditions (voix, voix et musique, voix et bruit de fond…) et qui varie de 1 à 5. L'UIT exige aussi une qualité équivalente ou supérieure à la qualité commerciale « toll quality » correspondant à des notes de 3,5 à 4 de MOS.

Afin de numériser la voix, différents types de codage sont utilisés tels que PCM (Puise Code Modulation), ADPCM (Adaptive Differential Puise Code Modulation), LPC (Linear Predictive Coding), CELP (Code Excited Linear Prédictive), LPAS (Linear Prediction Analysis by Synthesis), CS-ACELP (Conjugate Structure Algebric-CELP) et MP-MLQ (MultiPulse-Maximum Linklihood Quantization). Le choix du codage se fera en fonction d'éléments tels que le débit, la qualité de la parole (MOS), le retard codeur/décodeur.

2.1.2.2. La VoIP et les codecs

2.1.2.2.1. Définition

Le terme codec est construit d'après les mots codeur et décodeur, et fait souvent appel à la compression et décompression des données.

Il s'agit d'un procédé algorithmique permettant de compresser et de décompresser un signal, de l'audio ou de la vidéo, le plus souvent en temps réel. Le procédé peut être sous forme de logiciel ou encore de matériel (hardware). Par extension, le codec est le logiciel ou le circuit qui contient cet algorithme.

Ces algorithmes de compression de données permettent, en général, de réduire la taille du fichier original par des facteurs allant de 2 à 100 (voire plus pour certaines applications).

La compression se fait soit avec des algorithmes purement mathématiques de compression de données sans perte d'information ou par des algorithmes prenant en compte les caractéristiques des données à compresser et qui peuvent perdre des informations dites « non pertinentes ».

Pour faire passer la voix dans un réseau IP, il faut tenir compte de certains paramètres. On estime la bande passante de la voix à 4 kHz (300 - 3400 Hz). Ce qui donne après numérisation une bande passante de 8 kHz et après codage un débit de 64 kbps. Ceci entraîne que si on veut transmettre la voix sur un réseau IP sans mécanisme d'optimisation de la taille il nous faudra un débit «continu» de 64 kbps rien que pour la partie données à transmettre. Toutefois, ceci est rarement le cas, car à l'aide de mécanisme de codage optimisé grâce aux lois de compression de l'information, on arrive à réduire ce débit nécessaire de plus de 8 fois pour les meilleurs algorithmes de codage. Les codecs déterminent à quelle vitesse la voix sera échantillonnée et par là même le flux de données numériques que va générer la transformation d'un échantillon temporel de voix analogique.

Afin d'avoir la meilleure qualité possible de la voix après avoir passé au travers d'un codec, on a défini les exigences suivantes comme résultantes de la compression :

- Robustesse contre les erreurs binaires
- Robustesse contre les pertes de paquets puisque, généralement, il n'y pas de temps pour demander la retransmission de paquets et qu'il y a dégradation progressive de la qualité vocale en cas de perte (par exemple par la production d'un supplément vocal et d'un bruit de fond)

Les principaux codecs utilisés en VoIP sont résumés dans le tableau suivant :

Standards	*Méthode de codage*	*Débits (kbps)*	*Délais de traitement (ms)*
G.711	PCM	64	0,125
G.726	ADPCM	32	0,125
G.728	LPAS, CELP	16	0,625
G.729	LPAS, CS-ACELP	8	15
G.723.1	ACELP, MP-MLQ	6,4 / 5,3	37,5

Tableau 2.01 : Les codecs de la VoIP

Généralement l'utilisateur peut paramétrer son logiciel de VoIP et ainsi choisir le codec qui lui convient. Il est cependant recommandé d'utiliser soit le G.711 soit le G.729 vu qu'ils détiennent les meilleurs scores de MOS.

Remarque :

Le G.723.1 possède deux débits différents car il utilise deux types de compression qui sont le MP-MLQ (6,4 kbits/s) et ACELP (5,3 kbits/s). Mais pour les autres, bien qu'utilisant des méthodes de codage différents, le débit est toujours identique (exemple : G.729).

2.1.2.2.2. Le MOS

La qualité d'un codec est mesurée de façon subjective en laboratoire par une population test de personnes. Cette méthode est définie par la spécification ITU P800. Ces dernières écoutent tout un ensemble de conversations compressées selon les différents codecs à tester et les évaluent qualitativement selon la table suivante :

Qualité de la parole	*Score*
Excellente	5
Bonne	4
Correcte	3
Pauvre	2
Insuffisante	1

Tableau 2.02 : Echelle utilisée pour l'évaluation de la qualité de la voix

Sur la base des données numériques des appréciations, une opinion moyenne de la qualité d'écoute (Mean Opinion Score MOS) est ensuite calculée pour chaque codec. Les résultats obtenus pour les principaux codecs sont résumés dans le tableau ci-dessous :

Codecs VoIP	*Débits (kbps)*	*Score MOS*
G.711	64	4,2
G.726	32	3,85
G.729	8	4,0
G.723.1	6,4	3,9
G.723.1	5,3	3,7

Tableau 2.03 : Score MOS des différents codecs

2.2. Normes et protocoles de la VoIP [13] [14] [15] [16] [18] [19] [22]

La VoIP met en œuvre des techniques de télécommunications sur un réseau de paquets. Une normalisation de la signalisation est donc bien entendu nécessaire pour garantir l'interopérabilité des appareils. Outre le protocole IP, plusieurs protocoles assurent le bon fonctionnement de la technologie 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. Ensuite, plus ancien, le MGCP (Media Gateway Control Protocol). Et le plus récent est le SIP. Les plus populaires sont le H.323 et le SIP. H.323 est plus difficile à mettre en œuvre par rapport à SIP et que sa description est plus complexe.

2.2.1. Protocole H.323

2.2.1.1. Introduction

Avec le développement du multimédia sur les réseaux, il est devenu nécessaire de créer des protocoles qui supportent ces nouvelles fonctionnalités, telles que la visioconférence : l'envoi de son et de vidéo avec un souci de données temps réel. Le protocole H.323 est l'un d'eux. Il permet de faire de la visioconférence sur des réseaux IP.

H.323 est un protocole de communication englobant un ensemble de normes utilisées pour l'envoi de données audio et vidéo sur Internet. Il existe depuis 1996 et a été initié par l'ITU.

2.2.1.2. Fonctionnement

Le protocole H.323 est utilisé pour l'interactivité en temps réel, notamment la visioconférence (signalisation, enregistrement, contrôle d'admission, transport et encodage). C'est le leader du marché pour la téléphonie IP. Il s'inspire du protocole H.320 qui proposait une solution pour la visioconférence sur un réseau numérique à intégration de service (RNIS). Le protocole H.323 est une adaptation de H.320 pour les réseaux IP. A l'heure actuelle, la visioconférence sur liaison RNIS est toujours la technique la plus déployée. Elle existe depuis 1990. Les réseaux utilisés sont à commutation de circuits. Ils permettent ainsi de garantir une Qualité de Service (QoS) aux utilisateurs (pas de risque de coupure du son ou de l'image). Aujourd'hui, c'est encore un avantage indiscutable. Par contre, comme pour le téléphone, la facturation est fonction du débit utilisé, du temps de communication et de la distance entre les appels.

2.2.1.3. Architecture H.323

H.323 définit plusieurs éléments de réseaux :

Les ***terminaux*** - Dans un contexte de téléphonie sur IP, deux types de terminaux H.323 sont aujourd'hui disponibles. Un poste téléphonique IP raccordé directement au réseau Ethernet de l'entreprise. Un PC multimédia (muni d'une carte son et d'un micro) sur lequel est installée une application compatible H.323. 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 clé 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.

Les passerelles (GW: ***Gateway***) - Elles assurent l'interconnexion entre un réseau IP et le réseau téléphonique, ce dernier pouvant être soit le réseau téléphonique public, soit un PABX d'entreprise. Elles assurent la correspondance de la signalisation et des signaux de contrôle et la cohésion entre les médias. Pour ce faire, elles implémentent les fonctions suivantes de transcodage audio (compression, décompression), de modulation, démodulation (pour les fax), de suppression d'échos, de suppression des silences et de contrôle d'appels. Les passerelles sont le plus souvent basées sur des serveurs informatiques standards (Windows NT, Linux) équipés d'interfaces particuliers pour la téléphonie (interfaces analogiques, accès de base ou accès primaire RNIS, interface E1, etc.) et d'interfaces réseau, par exemple de type Ethernet. La fonctionnalité de passerelle peut toutefois être intégrée directement dans le routeur ainsi que dans les PBX eux-mêmes.

Les portiers ou *gardes-barrière* (GK: ***Gatekeeper***) - Ils sont des éléments optionnels dans une solution H.323. Ils ont pour rôle de réaliser la traduction d'adresse (numéro de téléphone - adresse IP) ou translation d'adresse entre les alias des terminaux et leur adresse (les alias peuvent être de type E.164 (numéro de téléphone) ou un identifiant tel qu'un nom de machine ou une adresse e-mail) et la gestion des autorisations. Cette dernière permet de donner ou non la permission d'effectuer un appel, de limiter la bande passante si besoin et de gérer le trafic sur le Lan. Les "gardes-barrière" permettent également de gérer les téléphones classiques et la signalisation permettant de router les appels afin d'offrir des services supplémentaires. Ils peuvent enfin offrir des services d'annuaires.

Les unités de contrôle multipoint (***MCU***, Multipoint Control Unit) - Référence au protocole T.120 qui permet aux clients de se connecter aux sessions de conférence de données. Les unités de contrôle multipoint peuvent communiquer entre elles pour échanger des informations de conférence.

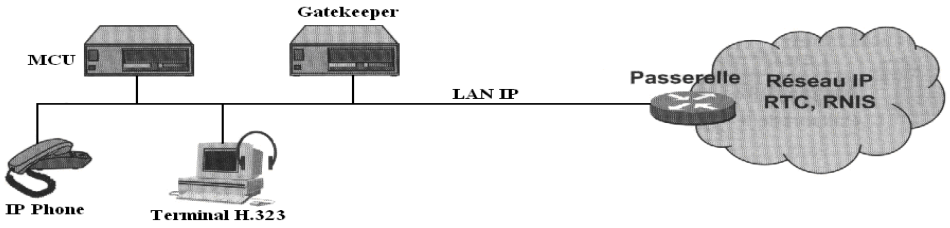

Figure 2.06 : Architecture matérielle d'une zone H.323

2.2.1.4. Les protocoles secondaires de H .323

Comme indiqué précédemment, H.323 ressemble davantage à une association de plusieurs protocoles différents et qui peuvent être regroupés en trois catégories : la signalisation, la négociation de codec, et le transport de l'information.

Trois protocoles de signalisation sont spécifiés dans le cadre de H.323 à savoir :

RAS (Registration, Admission and Status) - Ce protocole est utilisé pour communiquer avec un Gatekeeper. Il sert notamment aux équipements terminaux pour découvrir l'existence d'un Gatekeeper et s'enregistrer auprès de ce dernier ainsi que pour les demandes de traduction d'adresses. La signalisation RAS utilise des messages H.225.0 transmis sur un protocole de transport non fiable (UDP, par exemple).

Q.931 - H.323 utilise une version simplifiée de la signalisation RNIS Q.931 pour l'établissement et le contrôle d'appels téléphoniques sur IP. Cette version simplifiée est également spécifiée dans la norme H.225.0.

H.245 - Ce protocole est utilisé pour l'échange de capacités entre deux équipements terminaux. Par exemple, il est utilisé par ces derniers pour s'accorder sur le type de codec à activer. Il peut également servir à mesurer le retard aller-retour (Round Trip Delay) d'une communication.

La figure ci-dessous illustre les relations protocolaires entre un terminal H.323 et un équipement non H.323. Le terminal téléphonique, non H.323, établit un appel en direction du PC multimédia, le routeur (Gateway H.323) interprète la numérotation et initialise un appel Q.931 vers l'agent H.323 distant. L'agent H.323 réalise la correspondance entre une adresse E.164 et une adresse IP, il établit un canal de communication entre le demandé et le demandeur.

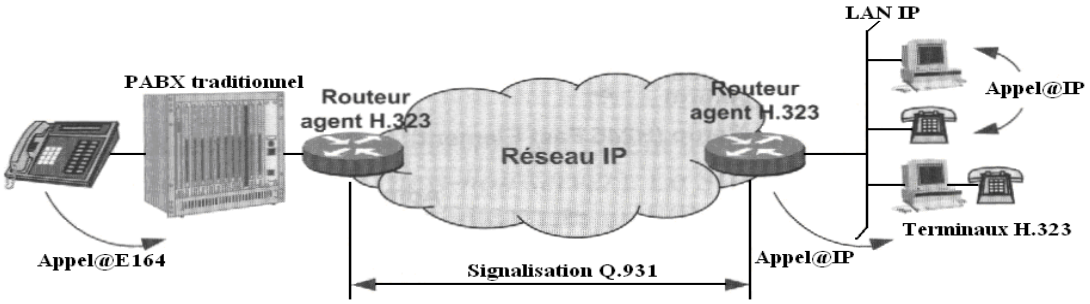

Figure 2.07 : Mise en relation selon H.323

L'établissement d'une communication H.323 diffère selon que le système utilise ou non un Gatekeeper. En mode direct (sans Gatekeeper), le terminal qui établit un appel E.164 assure lui-même la traduction d'adresse, la liaison est établie directement par un échange de messages H.225 et H.245. En mode Gatekeeper, le terminal appelant interroge, au préalable, le gatekeeper pour traduire et obtenir l'autorisation d'appeler son correspondant (garantie de bande passante pour accepter un nouveau flux). De même, le correspondant n'accepte l'appel qu'après autorisation de son Gatekeeper.

2.2.1.5. Phases d'appel de H.323

Une communication H.323 se déroule en cinq phases :

- Etablissement d'appel
- Echange de capacité et réservation éventuelle de la bande passante à travers le protocole RSVP (Ressource reSerVation Protocol)
- Etablissement de la communication audio-visuelle
- Invocation éventuelle de services en phase d'appel (par exemple, transfert d'appel, changement de bande passante, etc.)
- Libération de l'appel

2.2.1.6. H323 dans le modèle OSI

Les différents protocoles sont représentés ci-dessous dans le modèle OSI :

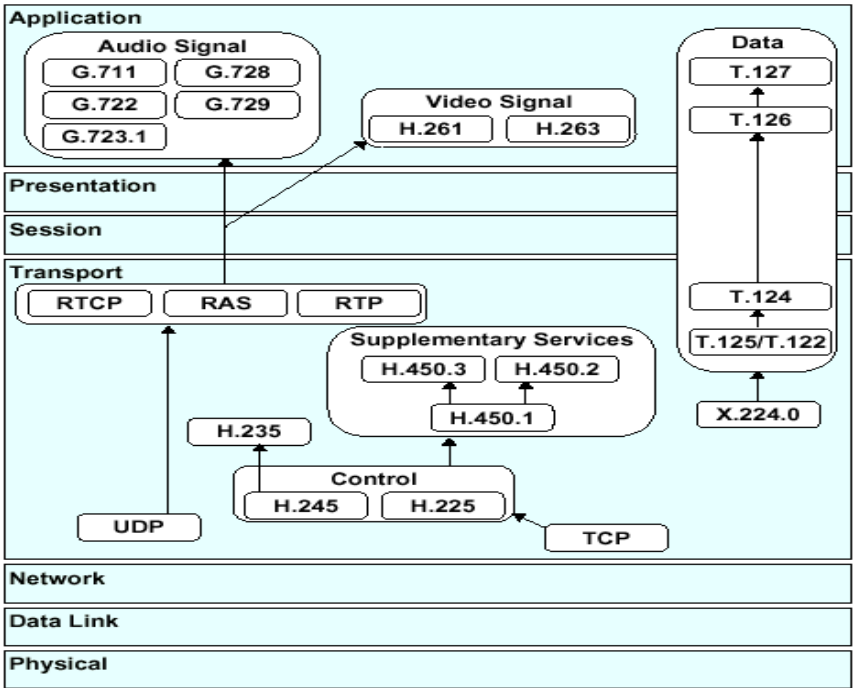

Figure 2.08 : H.323 dans OSI

Cette structure nous fait remarquer que H-323 rassemble de nombreux protocoles représentés dans le tableau 2.04 suivant :

Rôle	*Normes*	*Description*
Terminal, Contrôle et Administration	H.225	Signalisation d'appels, paquétisation des signaux, état et admission au garde-barrière
	H.245	Contrôle (négociation et établissement de sessions)
	RTP	Protocole de transport pour les applications temps réels
	RTCP	Contrôle de flux de données multimédia
	RSVP	Réservation d'une ressource en bande passante
Données	T.120	Contrôle des données
Applications audio/vidéo	G.7xx	Codecs audio
	H.26x	Codec vidéo
Services supplémentaires	H.450	Divers services supplémentaires (attente,…)
Sécurité	H.235	Procédure de sécurité dans l'environnement H.323

Tableau 2.04 : Les normes de H.323

2.2.2. Protocole SIP

2.2.2.1. Introduction

Le protocole SIP (Session Initiation Protocol) a été initié par le groupe MMUSIC (Multiparty Multimedia Session Control) et désormais repris et maintenu par le groupe SIP de l'IETF donnant la RFC 3261 rendant obsolète la RFC 2543. SIP est un protocole de signalisation appartenant à la couche application du modèle OSI. Son rôle est d'ouvrir, modifier et libérer les sessions. L'ouverture de ces sessions permet de réaliser de l'audio ou vidéoconférence, de l'enseignement à distance, de la voix (téléphonie) et de la diffusion multimédia sur IP essentiellement. Un utilisateur peut se connecter avec les utilisateurs d'une session déjà ouverte. Pour ouvrir une session, un utilisateur émet une invitation transportant un descripteur de session permettant aux utilisateurs souhaitant communiquer de s'accorder sur la compatibilité de leur média, SIP permet donc de relier des stations mobiles en transmettant ou redirigeant les requêtes vers la position courante de la station appelée. Enfin, SIP possède l'avantage de ne pas être attaché à un médium particulier et est sensé être indépendant du protocole de transport des couches basses.

2.2.2.2. Fonctionnement

SIP intervient aux différentes phases de l'appel :

- Localisation du terminal correspondant ;
- Analyse du profil et des ressources du destinataire ;
- Négociation du type de média (voix, vidéo, données...) et des paramètres de communication ;
- Disponibilité du correspondant, détermine si le poste appelé souhaite communiquer, et autorise l'appelant à le contacter ;
- Etablissement et suivi de l'appel, avertit les parties appelant et appelé de la demande d'ouverture de session, gestion du transfert et de la fermeture des appels ;
- Gestion de fonctions évoluées : cryptage, retour d'erreurs, ...

2.2.2.3. Les clients et les serveurs de SIP

Dans un système SIP on trouve deux types de composantes, les users agents (UAS, UAC) et un réseau de serveurs :

L'UAS (User Agent Server) - Il représente l'agent de la partie appelée. C'est une application de type serveur qui contacte l'utilisateur lorsqu'une requête SIP est reçue. Et elle renvoie une réponse au nom de l'utilisateur.

L'U.A.C (User Agent Client) - Il représente l'agent de la partie appelante. C'est une application de type client qui initie les requêtes SIP.

2.2.2.3.1. Les clients SIP

Le protocole SIP fonctionne en mode client/serveur, dans un même terminal client on trouve deux applications : UAC (User Agent Client) qui émet les requêtes et UAS (User Agent Server) qui reçoit les requêtes. Et le terminal fonctionne seulement comme l'un ou l'autre par transaction. Ces deux applications ensemble forment le UA (User Agent). Un terminal A peut appeler directement un terminal B ou passer par des serveurs. Un client doit avoir un minimum de méthodes afin d'établir la connexion. Les échanges entre un terminal appelant et un terminal appelé se font par l'intermédiaire de requêtes.

2.2.2.3.2. Les serveurs SIP

Les serveurs SIP sont des applications qui acceptent les requêtes d'un terminal SIP. Il y a quatre types de serveurs SIP :

Le PS *(Proxy Server)* ou serveur de délégation ou encore relais mandataire, auquel est relié un terminal fixe ou mobile, agit à la fois comme un client et comme un serveur. Un tel serveur peut interpréter et modifier les messages qu'il reçoit avant de les retransmettre. Il interprète et route l'appel en direction du destinataire. Il se peut que le proxy ne sache pas où se trouve le destinataire, c'est pourquoi dans ce cas là, il consulte un serveur de localisation. Il y a deux sortes de proxy, les proxys «stateful» et «stateless», la différence est le fait que le proxy «stateful» enregistre la position du destinataire tandis que le proxy «stateless» ne la mémorise pas. Ce qui fait que le proxy «stateful» consulte une seule fois le serveur de localisation par destination jusqu'à ce que la destination soit effacée de sa «table de routage».

Le RS *(Redirect Server)* ou serveur de redirection - Il réalise simplement une association (mapping) d'adresses vers une ou plusieurs nouvelles adresses (lorsqu'un client appelle un terminal mobile - redirection vers le PS le plus proche, ou en mode multicast - le message émis est redirigé vers toutes les sorties auxquelles sont reliés les destinataires). Notons qu'un Redirect Server est consulté par l'UAC comme un simple serveur et ne peut émettre de requêtes contrairement au PS.

Le LS *(Location Server)* ou serveur de localisation - Il fournit la position courante des utilisateurs dont la communication traverse les RS et PS auxquels il est rattaché. Cette fonction est assurée par le service de localisation.

Le RG *(Registrar)* - C'est un serveur qui accepte les requêtes Register et offre également un service de localisation comme le LS. Chaque PS ou RS est généralement relié à un Registrar.

2.2.2.4. Exemple d'établissement d'un appel SIP à travers un proxy

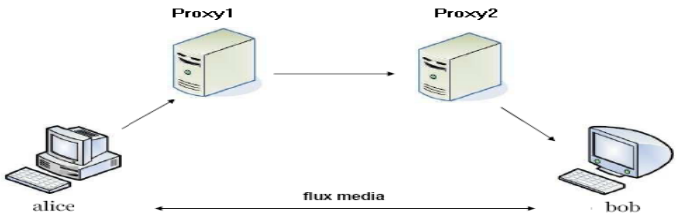

Figure 2.09 : Exemple d'architecture SIP

La figure 2.10 suivante illustre le principe d'établissement de session SIP entre deux usagers du réseau Alice et Bob, au travers de deux serveurs proxy, le premier utilisé pour diriger la requête vers le bon réseau, le deuxième servant à localiser le destinataire Bob dans son réseau.

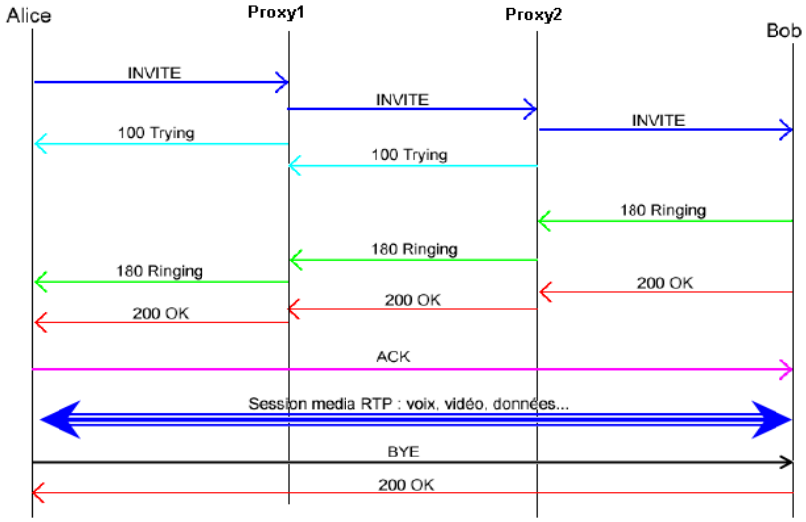

Figure 2.10 : Exemple d'établissement de session SIP

En général, on peut découper en trois grandes étapes l'établissement de communication SIP :

L'initialisation de la session : le message INVITE utilisé pour établir une session, contient l'adresse de Bob (sous forme URL : bob@domain.com). Ce message contient aussi la description des paramètres de la session demandée par Alice. Le proxy1 traduit le domaine du destinataire (domain.com) en adresse Internet, et transmet la requête vers le proxy2. A son tour, le proxy2 va chercher l'adresse Internet de Bob afin de lui faire parvenir le message INVITE. Le code réponse 180 indique à Alice que le requête est parvenue jusqu'au terminal de Bob. Le message 100 est envoyé pour stopper la retransmission de message INVITE.

L'établissement : le code 200 est émis par le terminal de Bob lorsque la session est acceptée (équivalent du « décroché »). Ce message contient des paramètres qui peuvent éventuellement différer de ceux demandés par Alice, en fonction des capacités du terminal de Bob. Le message « ACK » signifie que la session est finalement acceptée par Alice, avec les paramètres négociés entre les deux entités distantes.

La fin de la session de communication signifiée par Alice au travers le message BYE

Résumé :

Les étapes principales d'appel de SIP sont :

- Localiser les terminaux.
- Contacter les terminaux pour déterminer s'ils sont prêts à établir une session.
- Echanger les informations de média pour permettre l'établissement de la session.
- Communiquer puis libérer la connexion

2.2.2.5. Comparaison avec H.323

Les 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. Ils remplissent les mêmes rôles mais suivant des complexités nettement différentes. SIP est plus évolutif que H.323. Cependant, on ne saurait dire lequel des deux est le plus avantageux. Aussi, pour le choix d'une solution SIP ou H.323, il sera bien de se renseigner sur des calculs de performances de chacun d'eux et d'adopter la meilleure solution suivant les besoins.

	SIP	***H.323***
Nombre d'échanges pour établir la connexion	1,5 aller-retour	6 à 7 aller-retour
Maintenance du code protocolaire	Simple par sa nature textuelle à l'exemple de http	Complexe et nécessitant un compilateur
Evolution du protocole	Protocole ouvert à de nouvelles fonctions	Ajout d'extensions propriétaires sans concertation entre vendeurs
Fonction de conférence	Distribuée	Centralisée par l'unité MC
Fonction de téléservices	Oui, par défaut	H.323 v2 + H.450
Détection d'un appel en boucle	Oui	Inexistante sur la version 1 : un appel routé sur l'appelant provoque une infinité de requêtes
Signalisation multicast	Oui, par défaut	Non

Tableau 2.05 : Comparaison entre SIP et H.323

2.2.2.6. Conclusion

La simplicité, la rapidité et la légèreté d'utilisation, tout en étant très complet, du protocole SIP sont autant d'arguments qui pourraient permettre à SIP de convaincre les investisseurs. De plus, ses avancées en matière de sécurité des messages sont un atout important par rapport à ses concurrents.

2.2.3. Protocole MGCP

Le protocole MGCP (Media Gateway Control Protocol) est complémentaire à H.323 ou SIP, et traite des problèmes d'interconnexion avec le monde téléphonique. C'est un protocole de contrôle des gateways. Les éléments de contrôle utilisés sont les « Call Agents » ou « Media Gateway Controllers ». Ces éléments :

- fournissent des signaux d'appels, de contrôle et des processus intelligents aux gateways.
- envoient et reçoivent également des commandes des gateways

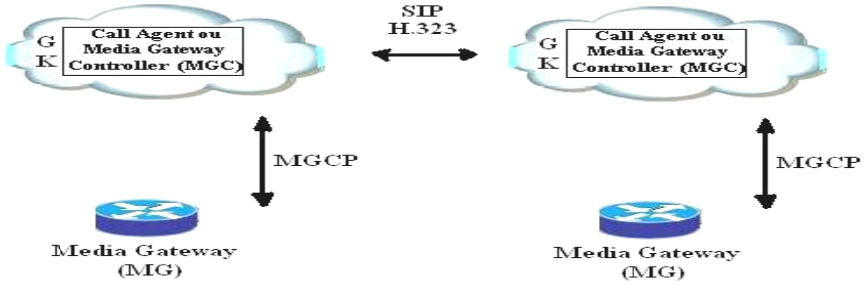

Figure 2.11 : Fonctionnement de MGCP

Les commandes du « call agent » sont: la configuration des terminaux, les requêtes de notification, la création, la modification et la suppression d'une connexion, et aussi, l'audit des terminaux ainsi que l'audit des connexions. Une amélioration de ce protocole s'appelle MEGACO (Media Gateway Controller) développé conjointement par l'IUT et I'IETF.

2.2.4. Protocoles RTP et RTCP

2.2.4.1. Protocole RTP

2.2.4.1.1. Objet du protocole

RTP (Real-time Transport Protocol), protocole développé par le groupe de travail « Audio and Video Transport » de ITETF, a pour objet de fournir un support pour le transport des données en temps réel comme celui appliqué à la visioconférence sur Internet.

Les services fournis par RTP consistent à :

- reconstituer la base de temps flux (horodatage des paquets : possibilité de resynchronisation des flux par le récepteur) ;
- détecter les pertes des paquets et informer la source ;
- sécuriser et identifier le contenu des données.

2.2.4.1.2. Caractéristiques de RTP

- RTP fournit des services de bout à bout pour les applications comme la visioconférence sur Internet. Cependant, il ne fournit aucun mécanisme pour assurer la délivrance à temps, mais seulement de restituer les données avec un débit correct comme l'émetteur les a envoyées.
- RTP n'offre pas de contrôle de flux ou de congestion. Il offre seulement une horloge de temps.
- RTP dépend de RSVP (Ressource reSerVation Protocol) pour la réservation des ressources et pour fournir la qualité de service demandée.

2.2.4.2. Le couple RTP et RTCP

RTCP (Real-time Transport Control Protocol) est un protocole de contrôle désigné pour travailler conjointement avec RTP. RTCP apporte à la source un retour d'informations sur la transmission et sur les éléments destinataires. L'information de contrôle est utile aux émetteurs et aux récepteurs. D'une part, l'émetteur peut ajuster sa transmission en se basant sur le rapport du récepteur. D'autre part, les récepteurs peuvent déterminer les cas de congestion du réseau afin d'informer l'interlocuteur. Le rapport du récepteur permet à la source d'adapter un type de codage ou de modifier le débit des données selon l'état de transmission.

En résumé, les services fournis par RTCP sont :

- l'information de contrôle ;
- le contrôle de congestion ;
- la synchronisation inter-média.

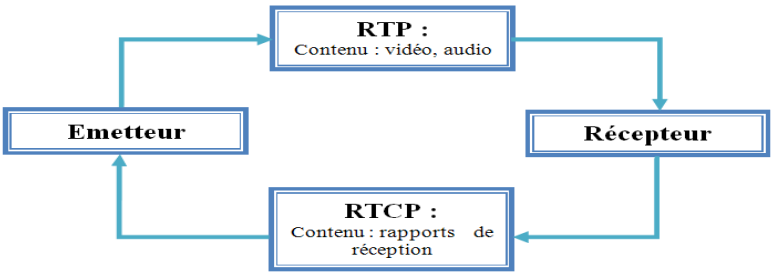

Figure 2.12 : Acheminement des paquets RTP et RTCP

2.3. Les problèmes de la VoIP et qualité de service [13] [14] [15] [18] [19] [20] [22]

2.3.1. Problèmes pour la VoIP

La première expérience avec la téléphonie sur Internet est encline à de nombreux problèmes tels que la lenteur de connexion, un délai de transmission trop long, les effets d'écho, etc.

Face aux opportunités offertes par la VoIP, il y a les risques.

La principale erreur serait que les administrateurs de réseaux informatiques considèrent que la voix étant transmise en paquets IP, il suffit de connecter les équipements Voix IP sur leurs réseaux déjà sécurisés pour rester sécurisé. En effet, une infrastructure voix IP a des exigences et des contraintes en matière de sécurité et de qualité de service très différentes du trafic de données traditionnelles.

Les risques inhérents à la mise en œuvre d'une application VoIP les plus évidents émanent d'Internet sont : le SPIT (Spam over IP Telephony), le déni de services pouvant aboutir à la saturation des boîtes vocales, les détournements ou interceptions d'appel, l'usurpation d'identité constituent également des risques mais nécessite une mise en œuvre plus complexe.

Il importe donc de sécuriser l'architecture de transport en séparant les trafics voix et data en 2 sous-réseaux (VLAN), de contrôler très rigoureusement l'accès et la liste des postes autorisés. Les équipements contribuant au projet VoIP doivent également être sécurisés : les serveurs doivent être isolés, les pare-feux existants adaptés ou de nouveaux pare-feux dédiés au projet, les mises à jour et correctifs effectués sur les postes IP et serveurs doivent être contrôlés et tracés ; sur les réseaux wifi, les communications doivent être cryptées.

2.3.2. Les paramètres de la VoIP

Le protocole IP, comme la majorité des autres technologies de transport de données en mode paquet, a été construit et optimisé pour transporter des fichiers des données, et non pas la voix ou la vidéo. Dans ce contexte, la seule qualité de service pertinente est de ne pas perdre ou corrompre les données transportées.

Aujourd'hui les avancées spectaculaires des technologies réseau rendent possible le transport de données temps réel comme la voix ou la vidéo en utilisant le protocole IP. Il devient donc particulièrement important de savoir contrôler et caractériser la qualité de service d'un réseau IP.

Plusieurs paramètres influent sur la qualité de la voix :

Le traitement de la voix (qualité de codage ou de compression) : influence du codec, du nombre de trames et la redondance.

La clarté : c'est la mesure de fidélité de la voix reçue par rapport à la voix émise, évaluée selon le MOS

La latence (délai de transmission ou délai de transit) : c'est le temps écoulé entre l'émission de la parole et sa restitution à l'arrivée. Pour permettre un échange interactif, la voix doit être transmise avec des contraintes de délai.

La gigue de phase : c'est la variation du délai de transmission, la gigue doit rester inférieure à 100 ms pour garder une qualité acceptable.

Le phénomène d'écho: l'écho est le délai entre l'émission du signal et la réception de ce même signal réverbéré. Ce phénomène est causé par les composantes électroniques des parties analogiques du système qui renvoient une partie du signal traité. Un écho inférieur à 50 ms n'est pas perceptible.

La bande passante : Généralement connu du grand public sous l'appellation débit maximum, elle constitue le taux de transfert maximum pouvant être maintenu entre deux points terminaux.

La disponibilité du réseau : c'est le taux moyen d'erreurs d'une liaison

Les pertes des paquets : le problème de perte de paquets est très lié au problème de la bande passante, ainsi qu'à l'utilisation appropriée d'algorithmes de contrôle de la congestion en bordure et dans le réseau cœur.

Les erreurs d'étiquetage/d'adressage.

2.3.2.1. La latence

La latence désigne le délai de transmission du signal de bout en bout. Plusieurs facteurs influent sur elle : la bande passante disponible, son occupation (trafic), les algorithmes de sécurisation (qui ont tendance à introduire des délais supplémentaires), etc. Trop de latence introduit des blancs dans la conversation qui font perdre son côté naturel.

La maîtrise du délai de transmission est un élément essentiel pour bénéficier d'un véritable mode conversationnel et minimiser la perception d'écho.

La durée de traversée d'un réseau IP dépend de nombreux facteurs :

- Le débit de transmission sur chaque lien
- Le nombre d'éléments réseaux traversés

- Le temps de traversée de chaque élément, qui est lui même fonction de la puissance et la charge de ce dernier, du temps de mise en file d'attente des paquets, et du temps d'accès en sortie de l'élément
- Le délai de propagation de l'information, qui est non négligeable si on communique à l'opposé de la terre.

Noter que le temps de transport de l'information n'est pas le seul facteur responsable de la durée totale de traitement de la parole. Le temps de codage et la mise en paquet de la voix contribuent aussi de manière importante à ce délai.

Il est important de rappeler que sur les réseaux IP actuels (sans mécanismes de garantie de qualité de service), chaque paquet IP « fait son chemin » indépendamment des paquets qui le précèdent ou le suivent: c'est ce qu'on appelle grossièrement le « Best effort » pour signifier que le réseau ne contrôle rien. Ce fonctionnement est fondamentalement différent de celui du réseau téléphonique où un circuit est établi pendant toute la durée de la communication.

Les chiffres suivants (tirés de la recommandation UIT-T G.114) sont donnés à titre indicatif pour préciser les classes de qualité et d'interactivité en fonction du retard de transmission dans une conversation téléphonique. Ces chiffres concernent le délai total de traitement, et pas uniquement le temps de transmission de l'information sur le réseau.

Classe n°	*Délai par sens*	*Commentaires*
1	0 à 150 ms	Acceptable pour la plupart des conversations
2	150 à 300 ms	Acceptable pour des communications faiblement interactives
3	300 à 700 ms	Devient pratiquement une communication half duplex (mode alterné, communication de type militaire)
4	Au delà de 700 ms	Inutilisable sans une bonne pratique de la conversation half duplex (mode alterné)

Tableau 2.06 : Classes de qualité en fonction du temps de latence

Un délai par sens de 150 ms correspond à une conversation avec un bond satellite et un délai par sens de 400 ms correspond à deux bonds satellite consécutifs.

En conclusion, on considère généralement que la limite supérieure "acceptable", pour une communication téléphonique, se situe entre 150 et 200 ms par sens de transmission (en considérant à la fois *le traitement de la voix et le délai d'acheminement*).

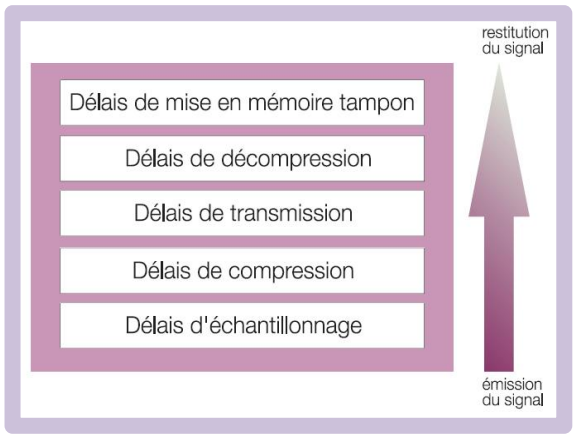

Figure 2.13 : Délais pouvant intervenir en VoIP

Le tableau 2.07 suivant représente l'influence du délai à la qualité vocale.

Délai par sens	*Indice de dégradation de la conversation*
200 ms	28 %
450 ms	35 %
700 ms	46 %

Tableau 2.07 : Qualité vocale en fonction du délai

Ceci montra bien que le délai est très important pour avoir une bonne qualité de la conversation.

La figure 2.14 suivante représente la possibilité de détérioration de la voix au cours de la transmission.

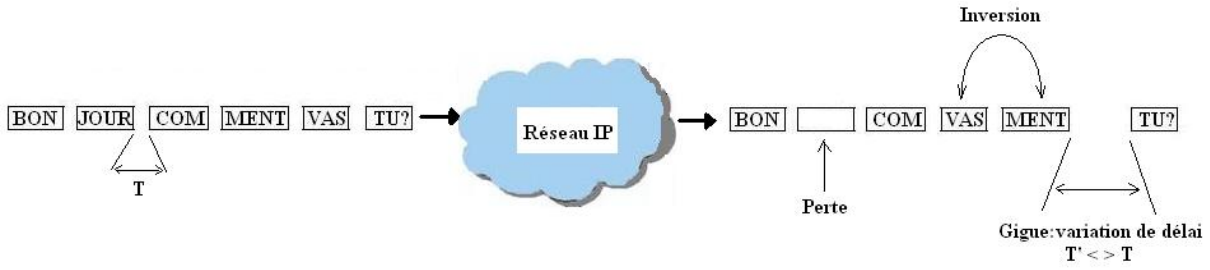

Figure 2.14 : Conséquences du délai

2.3.2.2. La gigue

La gigue est le phénomène provenant de la variation de la latence. En d'autres termes, la gigue mesure la variation temporelle entre le moment où deux paquets auraient dû arriver et le moment de leur arrivée effective. A certains moments, la latence peut être faible, et la voix peut être restituée avec un effet "temps réel" satisfaisant. Puis, une congestion temporaire du réseau peut

augmenter le délai d'arrivée des paquets, produisant un effet de parole hachée désagréable et rendant la conversation difficile à comprendre. Cette irrégularité d'arrivée des paquets est due à de multiples raisons dont: l'encapsulation des paquets IP dans les protocoles supportés, la charge du réseau à un instant donné, la variation des chemins empruntés dans le réseau, etc. Une solution pour éviter cela est la mise en mémoire tampon, c'est-à-dire la mémorisation préalable d'un certain nombre de paquets avant restitution sonore. L'ennui, c'est que ce processus (*bufferisation*) tend à augmenter les délais de bout en bout, altérant là encore l'aspect "temps réel".

La dégradation de la qualité de service due à la présence de gigue, se traduit en fait, par une combinaison des deux facteurs cités précédemment: le délai et la perte de paquets; puisque d'une part on introduit un délai supplémentaire de traitement (buffer de gigue) lorsque l'on décide d'attendre les paquets qui arrivent en retard, et que d'autre part on finit tout de même par perdre certains paquets lorsque ceux-ci ont un retard qui dépasse le délai maximum autorisé par le buffer.

2.3.2.3. La perte et le déséquencement de paquets

Ces problèmes sont des erreurs dans la transmission des paquets IP. Certains peuvent se perdre en cours de route, ou encore les paquets peuvent arriver dans le désordre avec une perte d'information ne permettant pas de les réordonnancer correctement. Parmi les solutions mises en œuvre pour lutter contre ces problèmes, l'émission redondante des paquets, l'analyse de leur intégrité et la mise en œuvre de processus d'interpolation pour remplacer les valeurs manquantes font partie de l'attirail disponible.

Lorsque les buffers des différents éléments réseaux IP sont congestionnés, ils « libèrent » automatiquement de la bande passante en se débarrassant d'une certaine proportion des paquets entrant, en fonction de seuils prédéfinis. Cela permet également d'envoyer un signal implicite aux terminaux TCP qui diminuent d'autant leur débit au vu des acquittements négatifs émis par le destinataire qui ne reçoit plus les paquets. Malheureusement, pour les paquets de voix, qui sont véhiculés au dessus d'UDP, aucun mécanisme de contrôle de flux ou de retransmission des paquets perdus n'est offert au niveau du transport. D'où l'importance des protocoles RTP et RTCP qui permettent de déterminer le taux de perte de paquet, et d'agir en conséquence au niveau applicatif.

Si aucun mécanisme performant de récupération des paquets perdus n'est mis en place (cas le plus fréquent dans les équipements actuels), alors la perte de paquet IP se traduit par des ruptures au

niveau de la conversation et une impression de hachure de la parole. Cette dégradation est bien sûr accentuée si chaque paquet contient un long temps de parole (plusieurs trames de voix par paquet). Par ailleurs, les codeurs à très faible débit sont généralement plus sensibles à la perte d'information, et mettent plus de temps à « reconstruire » un codage fidèle.

Enfin connaître le pourcentage de perte de paquets sur une liaison n'est pas suffisant pour déterminer la qualité de la voix que l'on peut espérer, mais cela donne une bonne approximation. En effet, un autre facteur essentiel intervient; il s'agit du modèle de répartition de cette perte de paquets, qui peut être soit « régulièrement » répartie, soit répartie de manière corrélée, c'est à dire avec des pics de perte lors des phases de congestion, suivies de phases moins dégradées en terme de QoS.

Le tableau 2.08 ci -dessous donne un bref aperçu des valeurs acceptables en VoIP.

	Bon	***Moyen***	***Mauvais***
Délai de transit	D < 150 ms	150 ms < D < 400	D > 400 ms
Gigue déphasé	G < 20 ms	20 ms < G < 50 ms	G > 50 ms
Perte de données	P < 1 %	1 % < P < 3 %	P > 3 %

Tableau 2.08 : Seuils de valeurs pour les paramètres critiques

2.3.3. Mécanismes de QoS

La problématique de qualité de la voix sur IP est particulière car la voix attend de son transporteur autre chose que les données. Le comportement attendu pour le transport de la voix est exactement l'inverse de celui des données : 1% ou 2% de perte de données de voix en ligne ne sont pas trop gênants pour la qualité du service de VoIP, mais en revanche une variation fréquente de 100 ms sur le délai de transit est catastrophique et rend le service inutilisable. La voix attend donc du transport IP l'inverse de ce qu'exigent les données.

2.3.3.1. Mécanisme IntServ

2.3.3.1.1. Présentation de IntServ

IntServ est un modèle d'architecture qui propose de réserver des ressources avant de commencer à les utiliser, de manière à ce que le flux soit constant et répondant toujours aux critères de qualité lors du transfert de données. Le modèle est basé sur la définition de classes de service et la réservation de ressources dans les différents éléments du réseau. Le modèle d'architecture IntServ est décrit dans la RFC 1633.

L'implémentation de IntServ nécessite quatre composants: un "packet scheduler", un "admission control", un classificateur ("classifier"), et un protocole de réservation de ressources.

- Packet Scheduler (Ordonnanceur de paquet) : Le Packet Scheduler contrôle l'acheminement vers la prochaine destination, qui peut être un autre routeur ou un LAN, des différents paquets.
- Le classificateur : afin de pouvoir effectuer un contrôle du trafic, il faut pouvoir identifier chaque paquet entrant et donc pouvoir l'associer à une certaine classe. Le classificateur se base sur le contenu de l'en-tête du paquet et détermine à quelle classe (catégorie de flux) appartient le paquet.
- L'admission control : L'admission control implémente l'algorithme qui détermine si un routeur ou un hôte, est à même de répondre à une nouvelle demande de QoS, sans entraver les demandes qui ont été déjà accordées. Elle est invoquée dans chaque nœud afin de prendre la décision d'accepter ou de refuser une demande de service temps réel. D'autres tâches, comme l'authentification de ceux qui effectuent des réservations, l'établissement des rapports concernant celui qui a demandé telle réservation, lui sont également confiés.
- Le protocole de réservation : Il est nécessaire afin de créer et maintenir un état décrivant les spécificités d'un flux dans chaque routeur le long du chemin, ainsi que dans les hôtes finaux. Il existe la possibilité de définir plusieurs protocoles de réservation, mais le plus utilisé est RSVP.

2.3.3.1.2. Fonctionnalités des Protocoles de Réservations

Il y a quatre fonctions que doit effectuer un protocole de réservation comme RSVP :

- Trouver une route qui permette la réservation de ressources. Cela notamment avec le champ "type-of-service".
- Trouver une route qui dispose encore de suffisamment de capacité pour de nouveaux flux.
- S'adapter à une route qui devient défaillante : le mécanisme de rafraîchissement d'une réservation, s'effectue à certains intervalles de temps et cela peut provoquer un retard au niveau d'un service temps réel, ainsi, l'application devrait s'adapter à la route autant que possible.
- Ne pas changer une route tant que celle-ci est encore viable.

2.3.3.2. Mécanisme DiffServ

2.3.3.2.1. Concept de base de DiffServ

Le modèle DiffServ est établi par le RFC 3260. Il consiste à classer le trafic grâce à un code présent dans le paquet IP (le champ Type Of Service de IPv4...). Des traitements différents sont ensuite appliqués aux diverses classes de trafics identifiés par le contenu de ce code. La différentiation des services peut porter sur la priorité des paquets par rapport à d'autres. La base essentielle de ce modèle est donc de remarquer que les informations transportées par les paquets IP sont diverses et que certains types de données ont besoin d'un mode de transmission plus efficace que d'autres.

2.3.3.2.2. Fonctionnement

Les trames IPv4 contiennent un champ permettant d'identifier son type de contenu et par la même occasion de trouver quels types de privilèges QoS lui appliquer. Dès lors qu'une valeur précise est attribuée à ce champ, on dit que le paquet est coloré.

Le champ ToS est actuellement représenté par une partie dite DSCP (Differentiated Service Code Point field). Selon ce type de code, le routeur adopte un comportement de priorisation selon le type de flux ou de mise en attente.

2.3.3.3. MPLS

Sur l'initiative de l'IETF, le mécanisme de commutation de label MPLS (Multi-Protocol Label Switching) est conçu pour séparer le routage de la commutation. Sur un réseau étendu, beaucoup de routeurs sont mis en place. Appartenant à un même réseau, ils sont capables de tirer partie de leur architecture commune au niveau Liaison. Il n'est donc plus nécessaire de faire le long traitement d'une trame IP au niveau réseau. C'est là où intervient MPLS, en affectant un label commun pour tous les paquets d'une même destination. Plus besoin de gérer les adresses sources et destinataires. Donc tous les paquets d'un flux seront traités de manière identique et ainsi routés selon un chemin fixe.

2.4. La Téléphonie sur IP (ToIP) [13] [16] [17] [18] [19] [20] [21] [22]

2.4.1. Introduction

La Voix sur IP est la technologie utilisée pour transporter le service de téléphonie sur IP (transport de la voix sur un Backbone, LAN, MAN/WAN). La téléphonie sur IP est un service de téléphonie

fourni sur un réseau de télécommunications ouvert au public ou privé utilisant principalement le protocole de réseau IP. Cette technologie permet d'utiliser une infrastructure existante de réseau IP pour raccorder des terminaux IP que l'on nomme IP-PHONE, ainsi que des logiciels sur PC raccordés sur le même réseau IP que l'on nomme SOFTPHONE.

Il est à noter que dans le cadre de la téléphonie sur IP, l'on distingue la téléphonie fixe sur IP et la téléphonie mobile sur IP (exemple : GSM utilisant l'IP). Cependant, nous nous limiterons dans le cadre de ce rapport à la téléphonie fixe sur IP.

La ToIP n'est qu'un des aspects d'un concept bien plus large: la messagerie unifiée, c'est-à-dire l'intégration des différentes technologies de communication (PSTN, e-mail, MI, VoIP, SMS, MMS, répondeur voix, répondeur vidéo, etc.) au sein d'un même et unique système.

2.4.2. Principe de base

2.4.2.1. Traitement de la voix

La téléphonie sur IP est une transmission de la voix en mode paquets au format TCP/UDP. Pour comprendre le traitement complexe de la voix analogique (signaux électriques) en signaux binaires, voici un synoptique explicatif :

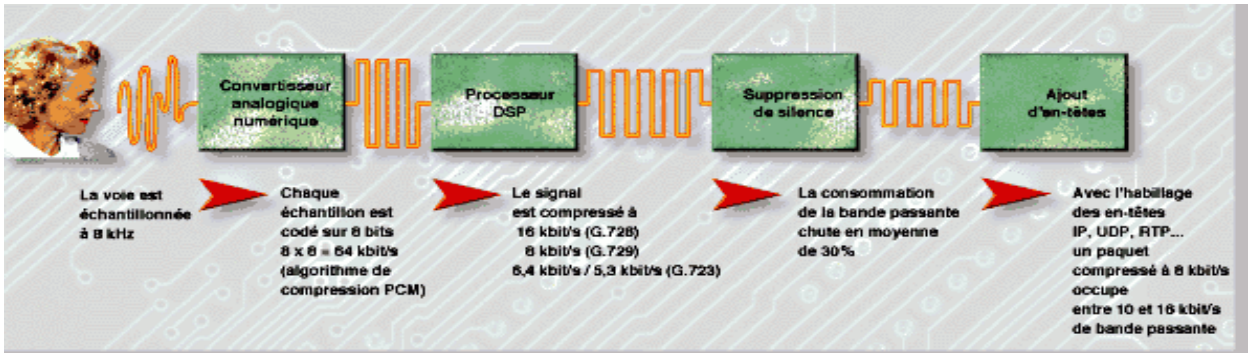

Figure 2.15 : Traitement du signal voix en ToIP

Explications du synoptique : La bande voix qui est un signal électrique analogique utilisant une bande de fréquence de 300 à 3400 Hz, elle est d'abord échantillonné numériquement par un convertisseur puis codé sur 8 bits, puis compressé par les fameux codecs (il s'agit de processeurs DSP) selon une certaine norme de compression variable selon les codecs utilisés, puis ensuite on peut éventuellement supprimer les pauses de silences observés lors d'une conversation, pour être ensuite habillé RTP,UDP et enfin en IP. Une fois que la voix est transformée en paquets IP, ces petits paquets IP identifiés et numérotés peuvent transités sur n'importe quel réseau IP (ADSL, Ethernet, Satellite, routeurs, switchs, PC, Wifi, etc.)

2.4.2.2. Les trois étapes d'une communication en ToIP

Première étape : la phase de mise en correspondance permet aux applications respectives d'échanger les paramètres nécessaires à leur exécution.

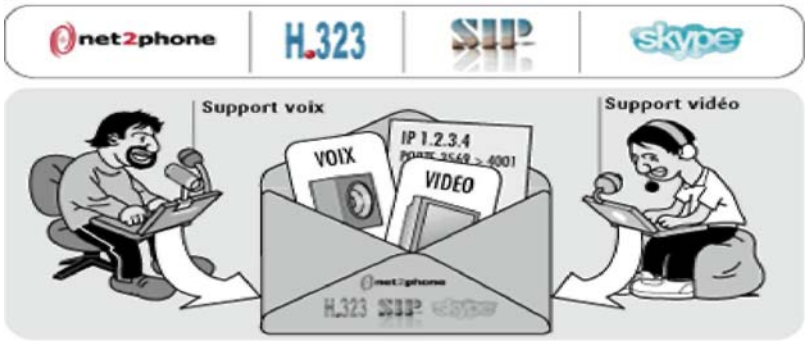

Figure 2.16 : Phase de mise en correspondance

Deuxième étape : l'établissement de la communication permettra aux applications respectives de transmettre les données au format numérique (voix, vidéo, etc.) au travers du réseau informatique.

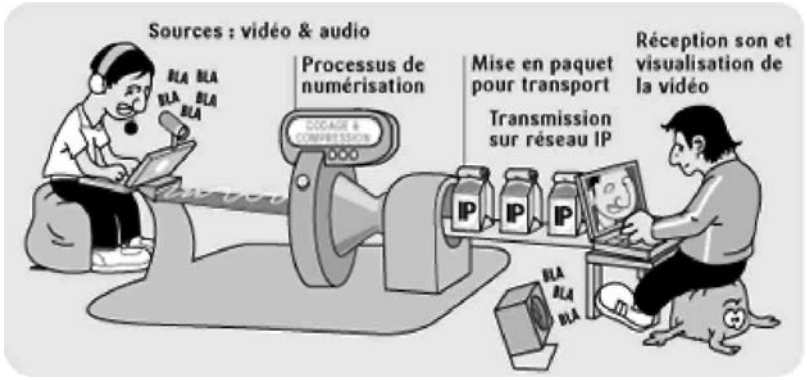

Figure 2.17 : Phase d'établissement de la communication

Troisième étape : une étroite collaboration entre RTP et RTCP permet, tout au long de la communication, aux applications respectives d'échanger des rapports sur la qualité des données reçues et de valoriser ces rapports afin d'améliorer la communication.

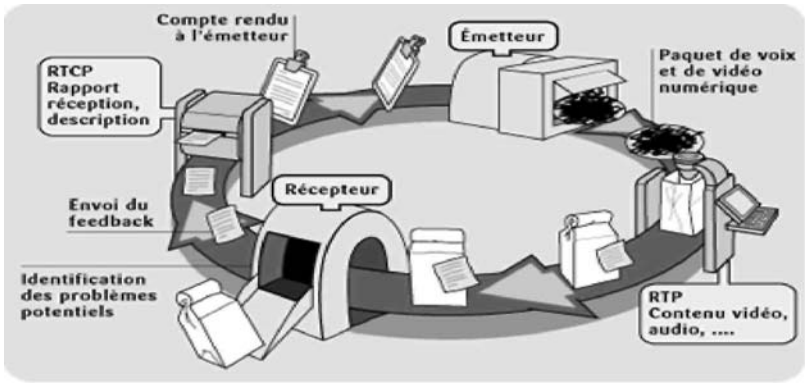

Figure 2.18 : Phase d'amélioration de la communication

2.4.3. Différence entre VoIP et ToIP

L'encapsulation de la voix sur IP peut être implémentée en deux points différents de l'infrastructure du réseau voix de l'entreprise :

- Limitée au réseau d'interconnexion privé des sites (technologie de transport) et contournant le réseau téléphonique commuté de l'opérateur voix : ***la Voix sur IP***
- Etendue de bout en bout du réseau de desserte jusqu'aux postes de travail et intégrant des services voix/données aux utilisateurs et aux exploitants : ***la Téléphonie sur IP***

A l'inverse de la Téléphonie sur IP, la Voix sur IP ne modifie pas le réseau voix existant : la conversion du signal voix en trames IP se fait au niveau du PABX, passerelle entre le monde de l'IP et de la téléphonie. Ni les réseaux voix, ni les terminaux téléphoniques existants ne sont modifiés.

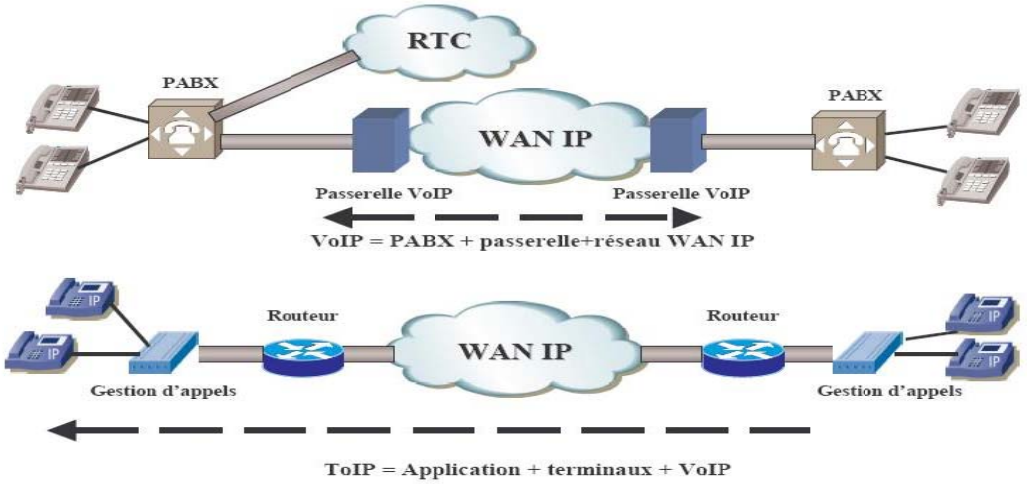

Figure 2.19 : Infrastructure du réseau IP d'entreprise (VoIP et ToIP)

La figure 2.20 suivante représente un exemple d'interconnexion de réseaux hétérogènes en VoIP.

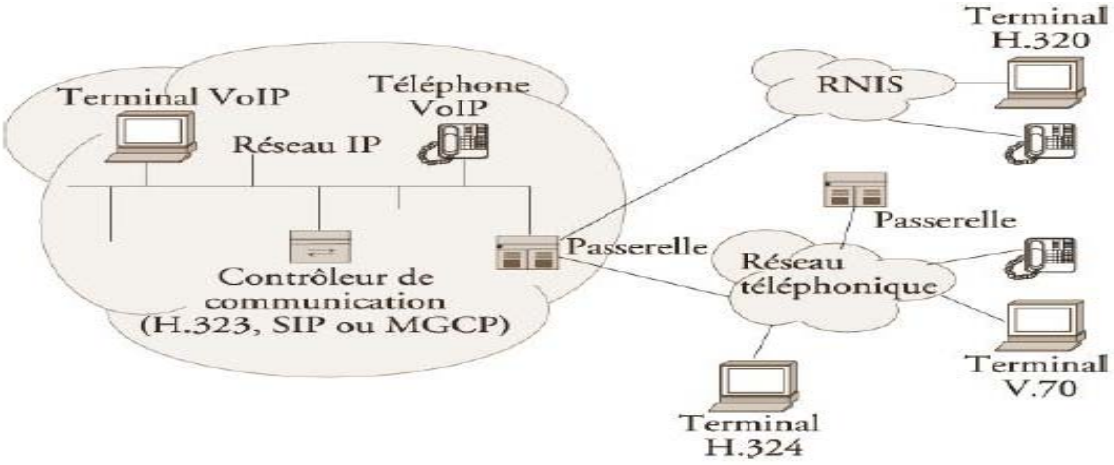

Figure 2.20 : Topologie d'un réseau IP

2.4.4. Les différents éléments pouvant composer un réseau

Les équipements dépendent de l'architecture choisie. Les éléments principaux que l'on peut retrouver sont les suivants :

- *Le PABX-IP (IPBX)*, c'est lui qui assure la commutation des appels et leurs autorisations, il peut servir aussi de routeur ou de switch dans certains modèles, ainsi que de serveur DHCP.
- *Le serveur de communications* (exemple : Call Manager de Cisco), il gère les autorisations d'appels entre les terminaux IP ou softphones et les différentes signalisations du réseau.
- *La passerelle (gateway)*, c'est un élément de routage équipé de cartes d'interfaces analogiques et/ou numériques pour s'interconnecter avec soit d'autres PABX (en QSIG, RNIS ou E&M), soit des opérateurs de télécommunications local, national ou international.
- *Le routeur*, il assure la commutation des paquets d'un réseau vers un autre réseau.
- *Le switch*, il assure la distribution et commutation de dizaines de port Ethernet à 10/100 voire 1000 Mbits/s.
- *Le gatekeeper*, il effectue les translations d'adresses (identifiant H323 et @ IP du référencement du terminal) et gère la bande passante et les droits d'accès. C'est le point de passage obligé pour tous les équipements de sa zone d'action.
- *Le MCU*, est un élément optionnel et gère les conférences audio vidéo.
- *L'IP-PHONE*, c'est un terminal téléphonique fonctionnant sur le réseau LAN IP à 10/100 avec une norme soit propriétaire, soit SIP, soit H.323.
- *Le SOFTPHONE*, c'est un logiciel qui assure toutes les fonctions téléphoniques et qui utilise la carte son et le micro du PC de l'utilisateur, et aussi la carte Ethernet du PC.
- *Autres équipements* : pour compléter la solution, on trouve fréquemment des équipements complémentaires comme une console d'administration pour le superviseur du réseau, un *serveur de taxation*, un *pare-feu* (indispensable), un *serveur de messagerie* vocale ou messagerie unifiée, etc.

Voici un exemple de communication en utilisant la téléphonie sur IP (figure 2.21) :

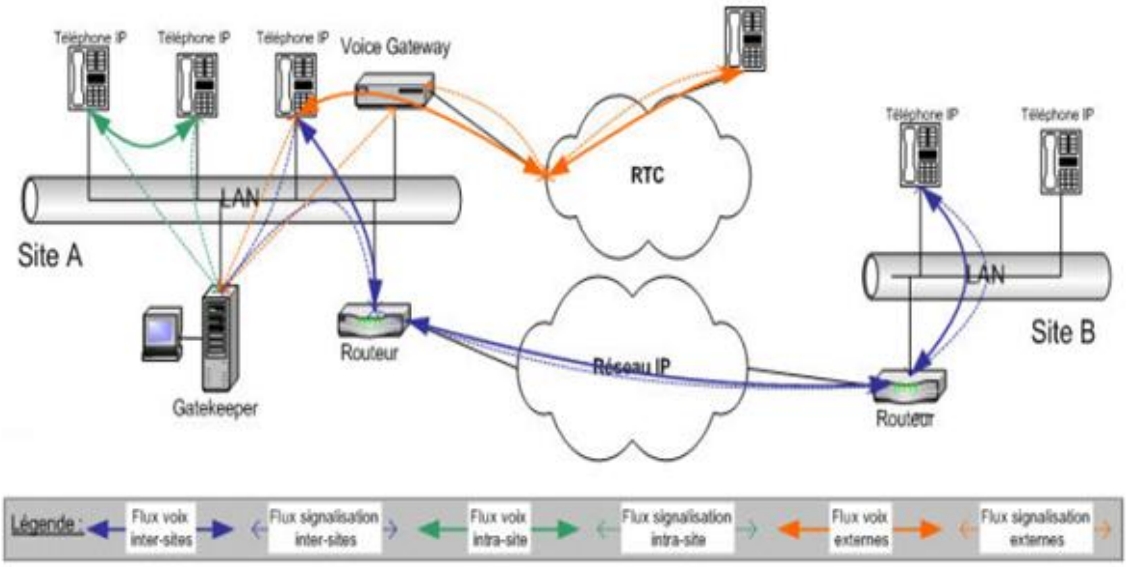

Figure 2.21 : Schéma illustratif de la communication ToIP

2.4.5. Avantages

La VoIP offre de nombreuses nouvelles possibilités aux opérateurs et utilisateurs qui bénéficient d'un réseau basé sur IP. Les avantages les plus marqués sont les suivants :

- Réduction des coûts
- Standards ouverts et interopérabilité multi-fournisseurs
- Un réseau voix, vidéo et données (triple play)
- Un service PABX distribué ou centralisé
- Evolution vers un réseau de téléphonie sur IP
- Intégration des services vidéo

2.4.6. Sécurité de la ToIP

2.4.6.1. Pourquoi la sécurité de la téléphonie

La nouvelle convergence dans les réseaux, introduit de nouveaux avantages mais aussi des vulnérabilités au niveau des échanges, aussi bien de données que de la voix. La sécurité est devenue un aspect prioritaire pour les services d'une manière générale. Dans la mesure où une grande partie de l'économie mondiale utilise ces infrastructures, se pose la question de sécuriser et de protéger cette infrastructure. Ainsi, les Hackers sont capables de compromettre et de contrôler différents aspects de la téléphonie, aussi bien les informations de signalisation que les paquets de voix.

2.4.6.2. Vulnérabilités et contraintes de la ToIP

Etant un service de téléphonie, la ToIP encourt également (au moins) les menaces suivantes :

- Ecoute des conversations
- Traçage des déplacements d'un usager
- Modification des données
- Rejeu (il implique la capture passive des données de signalisation ou de parole relative à l'appel et leur retransmission ultérieure en vue de produire un effet non autorisé).
- Accéder aux services avec l'identité d'autrui
- Scam (équivalent du spam en e-mail, appel inconnu non sollicité)
- Obstruction (consiste pour un pirate à empêcher un appel entrant ou sortant d'aboutir)
- Dégradation (consiste pour un pirate à diminuer la qualité de la conversation, en introduisant des signaux parasites, de la latence, etc.)
- Attaques en déni de service (consiste à émettre vers l'IPBX de l'entreprise une quantité importante de flux IP de manière à le saturer)
- Les virus et les vers informatiques (vulnérabilité particulière des softphones)

2.4.6.3. Définitions relatives à la sécurité

2.4.6.3.1. La confidentialité

La confidentialité est la propriété qui assure que seuls les utilisateurs habilités, dans des conditions prédéfinies, ont accès aux informations.

2.4.6.3.2. L'intégrité

L'intégrité est une propriété qui assure qu'une information n'est modifiée que dans des conditions prédéfinies, selon des contraintes précises. On parle de contraintes d'intégrité pour désigner l'ensemble des assertions qui définissent la cohérence d'un système d'information.

2.4.6.3.3. La fonction de hachage

Une fonction de hachage (parfois appelée fonction de condensation) est une fonction permettant d'obtenir un condensé (appelé aussi haché**)** d'un texte, c'est-à-dire une suite de caractères assez courte représentant le texte qu'il condense. La fonction de hachage doit être telle qu'elle associe un et un seul haché à un texte en clair (cela signifie que la moindre modification du document entraîne la modification de son haché). D'autre part, il doit s'agir d'une fonction à sens unique afin qu'il soit impossible de retrouver le message original à partir du condensé. Ainsi, le haché représente en quelque sorte l'empreinte digitale du document.

2.4.6.3.4. Le scellement de données

L'utilisation d'une fonction de hachage permet de vérifier que l'empreinte correspond bien au message reçu, mais rien ne prouve que le message ait bien été envoyé par celui que l'on croit être l'expéditeur.

Ainsi, pour garantir l'authentification du message, il suffit à l'expéditeur de signer le condensé à l'aide de sa clé privée (le haché signé est appelé sceau**)** et d'envoyer le sceau au destinataire. A la réception du message, il suffit au destinataire de déchiffrer le sceau avec la clé publique de l'expéditeur, puis de comparer le haché obtenu avec la fonction de hachage au haché reçu en pièce jointe. Ce mécanisme de création de sceau est appelé scellement.

2.4.6.3.5. L'authentification

L'authentification est la propriété qui assure que seules les entités autorisées ont accès au système, pour garantir l'identité du producteur d'un message, analogue à la signature numérique. Elle protège de l'usurpation d'identité.

La signature numérique, encore appelée signature électronique, est un procédé permettant de garantir l'authenticité de l'expéditeur d'un message en utilisant par exemple le principe de hachage.

2.4.6.3.6. La non répudiation

La non répudiation est la propriété qui assure que l'auteur d'un acte ne peut ensuite dénier l'avoir effectué. Elle peut constituer également un engagement contractuel du signataire qui ne peut plus par la suite revenir en arrière. La signature électronique permet également d'assurer la non répudiation.

2.4.6.3.7. Le Déni de Service

Les attaques par «Denial of Service» ("Déni de service", souvent abrégé en DoS) consistent à paralyser temporairement des serveurs afin qu'ils ne puissent être utilisés et consultés. Le but d'une telle attaque n'est pas de récupérer ou d'altérer des données, mais de nuire à la réputation de sociétés ayant une présence sur Internet en l'empêchant de fonctionner.

2.4.6.4. Sécurité et VoIP

Plusieurs protocoles permettent d'assurer la sécurité en VoIP. Dans le cas de H.323, H.235 définit les recommandations de sécurité où TLS (Transport Layer Security) est utilisé pour sécuriser le canal de signalisation. Le trafic de voix se sert du transport RTP pour des communications de bout en bout entre des points terminaux. Les trafics de signalisation et de voix peuvent être sécurisés respectivement par TLS, et DTLS (Datagram Transport Loyer Security) ou SRTP (Secure Real-

time Transport Protocol). IPSec permet de sécuriser tout le trafic IP, et donc aussi bien le trafic de signalisation et le trafic de voix. Pour SIP, le protocole SDP (RFC2327) est utilisé pour acheminer les paramètres de sécurité nécessaires au chiffrement, comme l'indication de l'algorithme de chiffrement ainsi que la clé.

2.4.6.4.1. La sécurité avec TLS

Le protocole TLS (RFC2246), qui est une norme de l'IETF, est le protocole le plus largement déployé pour sécuriser le trafic des applications du réseau IP nécessitant un transport fiable comme le trafic Web (HTTP) et la messagerie électronique (POP). Il s'insère entre la couche application et la couche réseau, pour sécuriser le trafic TCP tout en étant indépendant du protocole d'application offrant ainsi un canal sécurisé transparent. Ainsi TLS ne fonctionne pas au-dessus de UDP. En effet, la non fiabilité du transport a une influence directe sur la conception de TLS. TLS ne permet pas un déchiffrement indépendant. C'est à dire, nous ne pouvons pas déchiffrer le record N si le record N-l est perdu. De l'autre côté, les messages d'établissement de session (ou Handshake) sont supposés être bien délivrés de la source vers la destination et vice versa sinon la phase de handshake est interrompue. Ces deux points résument les faiblesses dans TLS.

L'implication évidente pour la téléphonie sur IP est la sécurisation du trafic de signalisation basé sur TCP. Mais la signalisation et les flots de médias qui utilisent UDP comme protocole de transport ne le seront pas.

2.4.6.4.2. La sécurité avec DTLS

Au cours de ces dernières années un nombre croissant de protocoles applicatifs, comme le SIP, RTP, le MGCP, et une variété de protocoles ont été conçues pour employer le transport sur UDP.

La solution de sécurité a été de concevoir un protocole de sécurité de canal générique qui suffira pour les applications les plus utilisées sur le transport datagramme UDP tel que TLS a fait pour TCP. Malgré les nombreux soucis, cette solution serait un projet de conception difficile, DTLS est sorti pour qu'il puisse fonctionner correctement au dessus du transport en datagramme. C'est une version modifiée de TLS qui prend comme référence le protocole IPSec. Celte approche a deux avantages principaux : d'abord, DTLS est quasiment semblable à TLS, l'infrastructure préexistante de ce dernier peut être réutilisée, en second lieu, puisque DTLS fournit une interface familière à une couche générique de sécurité, il est facile d'adapter les protocoles pour son emploi. Un des principaux buts dans la conception de DTLS est d'avoir des spécifications et caractéristiques de sécurité avoisinant ceux de TLS. En conséquence, DTLS n'offre aucune amélioration à TLS.

2.4.6.4.3. Le protocole SRTP

SRTP (RFC3711) est un nouveau mécanisme de sécurité considéré pour sécuriser les réseaux de VoIP, en assurant la sécurisation des paquets RTP qui transportent les échantillons de parole des usagers. Le but de SRTP est d'assurer la confidentialité des champs utiles des paquets RTP et RTCP, l'intégrité de tout le paquet RTP et RTCP avec protection contre le rejeu ainsi que l'authentification de l'origine des messages. Ces services de sécurité sont optionnels et mutuellement indépendants. Seule la protection de l'intégrité des paquets RTCP est obligatoire pour éviter la perturbation du flux RTP. SRTP utilise le cryptage symétrique (utilisation de clé privée), une fonction de hachage universelle pour l'authentification des messages et un numéro de séquence implicite pour le séquencement basé sur le numéro de séquence des en-têtes du paquet RTP. SRTP est implémenté entre l'application RTP et la couche transport. Il intercepte le paquet RTP à l'entrée et envoie le paquet SRTP équivalent en sortie coté émetteur, l'opération inverse est effectué du coté récepteur. Cependant, il n'assure pas la sécurisation des paquets de signalisation de la voix et requiert un protocole séparé pour la génération de clés de session.

2.4.6.4.4. Le protocole IPSec (IP Security)

Le protocole IPSec est une suite de protocoles désignés pour sécuriser les communications au niveau de la couche réseau. La suite de protocoles est constamment en évolution depuis 1995. IPSec propose deux protocoles de sécurité du trafic IP : Authentification Header (AH) (RFC 2402) et Encapsulating Security Payload (ESP) (RFC 2406). Les services de sécurité offerts par IPSec sont:

L'authentification des extrémités: cette authentification mutuelle permet à chacun de s'assurer de l'identité de son interlocuteur. Rappelons tout de même qu'IPSec est un protocole de niveau 3 et qu'il ne fournit donc qu'une authentification des machines mettant en œuvre le protocole plutôt que des personnes utilisant réellement la machine.

La confidentialité des données échangées: IPSec permet, si on le désire, de chiffrer les paquets IP pour éviter que quiconque puisse les lire. IPSec offre une protection contre les écoutes et analyses de trafic; il permet de chiffrer les adresses IP réelles de la source et de la destination, ainsi que tout l'en-tête IP correspondant.

L'intégrité des données échangées: IPSec permet de s'assurer qu'aucun paquet n'a subi aucune modification durant son trajet par utilisation d'une fonction unidirectionnelle de hachage.

La protection contre le rejeu: IPSec permet de se prémunir contre les attaques consistant à capturer un ou plusieurs paquets dans le but de les envoyer à nouveau (sans pour autant les avoir déchiffrés) pour bénéficier des mêmes avantages que l'envoyeur initial.

2.4.7. ToIP et NGN

2.4.7.1. Objectif

La commutation téléphonique évolue. Une nouvelle génération d'architectures de réseaux apparaît permettant d'offrir de nouveaux services émergents mixant voix et données : les NGN pour les opérateurs, et les réseaux de téléphonie sur IP pour les entreprises.

2.4.7.2. NGN et la Téléphonie IP

Les NGN sont un réseau de transport en mode paquet permettant la convergence des réseaux Voix/données et Fixe/Mobile ; ces réseaux permettront de fournir des services multimédia accessibles depuis différents réseaux d'accès.

Le protocole Internet (IP) est la base de ce modèle de réseau. Dans les NGN, l'IP peut être l'intégrateur universel des services voix, données et vidéo, aussi bien pour le fixe que pour le mobile. Le transport de tous ces services s'effectue sur un réseau IP commun. Il ne s'agit pas de l'internet public actuel bien qu'il s'y connectera. On peut envisager aujourd'hui que l'internet devienne une application ou un des services dont la fourniture est assurée par les NGN en faisant appel à la technologie IP.

Les NGN sont basés sur une évolution progressive vers le « tout IP » et sont modélisés en couches indépendantes dialoguant via des interfaces ouvertes et normalisées :

- La couche « ***Accès*** », qui permet l'accès de l'utilisateur aux services via des supports de transmission et de collecte divers : câble, cuivre, fibre optique, boucle locale radio, xDSL (ou x Digital Subscriber Line), réseaux mobiles.
- La couche « ***Transport*** », qui gère l'acheminement du trafic vers sa destination. En bordure du réseau de transport, des « média gateways » et des « signalling gateways » gèrent respectivement la conversion des flux de données et de signalisation aux interfaces avec les autres ensembles réseaux ou les réseaux tiers interconnectés.

- La couche « ***Contrôle*** », qui se compose de serveurs dits « softswitch » gérant d'une part les mécanismes de contrôle d'appel (pilotage de la couche transport, gestion des adresses), et d'autre part l'accès aux services (profils d'abonnés, accès aux plates-formes de services à valeur ajoutée).
- La couche « ***Services*** », qui regroupe les plates-formes d'exécution de services et de diffusion de contenus. Elle communique avec la couche contrôle du cœur de réseau via des interfaces ouvertes et normalisées, indépendantes de la nature du réseau d'accès utilisé. Les services et contenus eux-mêmes sont par ailleurs développés avec des langages convergents et unifiés.

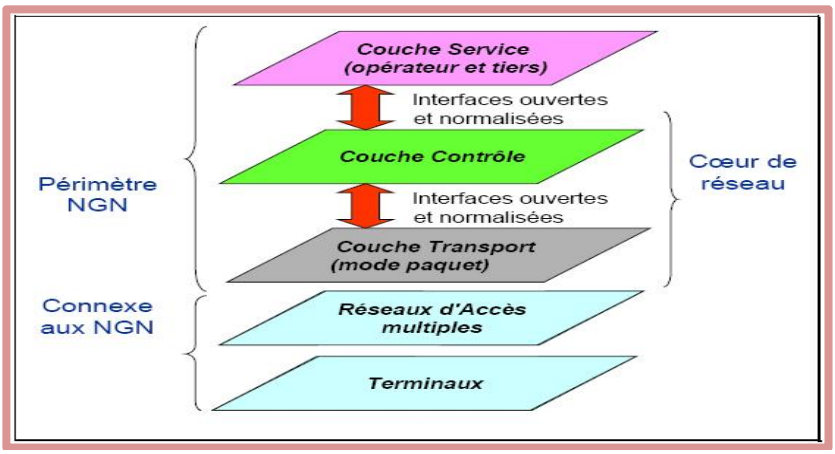

Figure 2.22 : Architecture NGN

Toutefois, la technologie IP offre de multiples autres opportunités de fonctionnalités téléphoniques de nouvelle génération. De plus, l'activation et l'annulation de ces services sont simplifiées par la mise en place d'interfaces Web.

Parmi les services de nouvelle génération disponibles, on retiendra notamment: l'accès universel à son carnet d'adresses personnel unique, l'accès universel au répertoire d'entreprise, l'intégration avec les répertoires mondiaux (par exemple: DNS, LDAP, H.350), l'obtention d'informations sur le client avec lequel la conversation est en cours, l'obtention des cours de la bourse, une application Web qui renvoie des informations au format XML, une application intelligence artificielle (une Interactive Voice Response (IVR) ou une Customer Response Application (CRA), comme par exemple une "secrétaire virtuelle"), le répondeur vidéo, la messagerie vidéo, l'enregistrement de la conversation audio/vidéo, la vidéo d'attente, la redirection vers la

messagerie électronique, la redirection vers un site Web, la redirection vers n'importe quel poste (IP ou pas), la mobilité de poste locale (logiciel sur PC, téléphone IP fixe ou mobile) au sein de l'entreprise, la mobilité de poste globale (logiciel sur PC, téléphone IP fixe ou mobile) à travers le monde, l'appel vers une conférence (afin d'y participer), le routage intelligent d'un appel (ex: prise en compte de données acquises durant la communication, l'état d'une file d'attente, l'heure actuelle, etc.), la liberté du lieu de prestations de service de helpdesk (par exemple: un centre d'appels peut être géographiquement délocalisé tant que l'accès au réseau IP est présent), la maintenance centralisée d'infrastructures décentralisées, la sélection de la passerelle la plus adéquate, etc.

2.5. Conclusion

La VoIP ouvre des horizons très riches en facilitant l'ajout d'applications à valeur ajoutée. En voici quelques exemples : la visiophonie, la convergence fixe - sans fil - mobile et le travail coopératif. Les terminaux téléphoniques IP n'en sont qu'à leurs débuts. Ils doivent encore évoluer pour : offrir un niveau de qualité de service au moins équivalent à celui de la voix traditionnelle, offrir de nouvelles interfaces utilisateur, des capacités de traitement multimédia, et des fonctionnalités de confort ; converger vers des protocoles standardisés permettant l'interopérabilité en environnement multi-constructeurs.

A l'origine, la voix sur IP était une technologie exclusivement réservée aux opérateurs de télécommunications. Cette technique de transport leur a permis de remplacer des liaisons TRUNK entre les équipements de leurs réseaux PSTN, permanentes et onéreuses, par des connexions dynamiques et moins coûteuses grâce au protocole IP.

Aujourd'hui, grâce à l'avènement de l'accès Internet large bande, l'intérêt pour ces technologies gagne également le public. Le consommateur professionnel ou privé tend à utiliser le réseau Internet comme infrastructure alternative d'accès aux services de téléphonie.

Au delà d'une réduction substantielle de ses coûts de communication, le consommateur peut ainsi avoir accès à une multitude de services complémentaires: la vidéo, le chat, le partage de contenus (par exemple les photos) ou d'applications, la messagerie instantanée, la messagerie unifiée (boîte vocale, comptes e-mail, fax, SMS, MMS,.., réunis au sein d'une seule "boîte" accessible de partout), la personnalisation de services (ringtones, déviation, musique/vidéo d'attente,...), l'utilisation mobile (comme pour le GSM) grâce aux technologies d'accès réseau sans fil (WiFi et WiMAX)…

On peut ainsi vraisemblablement penser que le protocole IP deviendra un jour un standard unique permettant l'interopérabilité des réseaux mondialisés. C'est pourquoi l'intégration de la voix sur IP n'est qu'une étape vers EoIP : Everything over IP.

Malgré tout ceci, des risques potentiels subsistent encore : vulnérabilité particulière des softphones, flux important de paquets IP qui peuvent provoquer l'encombrement de l'IPBX, et les risque d'écoute surtout pour les VoIP sur WiFi. Concrètement, l'opérateur protège contre trois types d'attaques : l'écoute téléphonique, la fraude d'identité et les portes dérobées.

Que faire alors ? Sécuriser un réseau ToIP / VoIP, c'est assurer deux choses, dans cet ordre : la disponibilité de la téléphonie et la confidentialité des conversations. ***La première mesure de sécurité, c'est donc la mise en place d'une politique de QoS.***

La qualité de service n'a pas été réellement prise au sérieux lors de la conception initiale des réseaux IP. Le protocole IP, comme la majorité des autres technologies de transport de données en mode paquet, a été construit et optimisé pour transporter des fichiers de données, et non pas la voix ou la vidéo.

Il est probable que dans l'avenir, la recherche se consacre principalement à dépasser la qualité vocale des réseaux téléphoniques actuels (en utilisant des codeurs bande élargie en voix sur IP, ou encore de codeurs stéréo ou spatialisés), et à améliorer la qualité de la voix sur IP sur réseaux sans fil en collaborant plus étroitement avec les mécanismes de qualité de service des couches physiques de l'UMTS ou du WiFi.

En résumé, les principaux paramètres qui caractérisent la QoS d'un réseau de transport en mode paquet sont : l'écho, la capacité (bande passante), le délai de transmission de bout en bout (latence), la variation de ce délai (gigue), le taux de perte de paquets et le taux de déséquencement de paquets.

Une solution qui pourra nous aider à avoir une bonne qualité de service est l'amélioration au niveau du passage des paquets dans le réseau IP : routage dynamique avec gestion de files d'attente, et des techniques de contrôle de flux et contrôle de congestion. ***Pour le routage, nous allons utiliser l'algorithme de Dijkstra pour trouver le chemin le plus court à prendre par les paquets en passant d'un routeur à un autre, avec une technique de routage basé sur les contraintes.***

CHAPITRE 3 ROUTAGE IP ET OPTIMISATION

La téléphonie sur IP utilise le transfert de paquets et le routage dynamique pour la transmission des données sur le réseau Internet.

3.1. Commutation et routage [4] [6] [8] [23] [24] [25] [28] [29] [30] [31] [32]

3.1.1. Commutation

Plusieurs techniques de transfert de données existent : la commutation de circuits, le transfert de messages, le transfert de paquets, la commutation de trames et de cellules. Le mot transfert indique que commutation et routage sont possibles tandis que commutation seule insiste sur le fait que seul un mode commuté existe.

3.1.1.1. Commutation de circuits

Dans ce cas, un circuit matérialisé est construit entre l'émetteur et le récepteur. Ce circuit n'appartient qu'aux deux entités qui communiquent. Le circuit doit être établi avant que les informations puissent transiter. Il dure jusqu'au moment où l'un des deux abonnés interrompt la communication. Même si les deux correspondants n'ont pas de données à transmettre pendant un certain temps, la liaison ne peut pas être utilisée par d'autres. Le réseau à commutation de circuits est "*orienté connexion*" (connectionful). D'où l'idée de concentrer sur une même liaison plusieurs communications pour que le taux d'utilisation des liaisons augmente. Si de nombreuses communications utilisent une même liaison, une file d'attente va se former. Il va falloir prévoir des zones de mémoire pour retenir les messages en attendant que la liaison soit disponible.

3.1.1.2. Transfert de paquets

Pour accroître la vitesse de transmission et rendre beaucoup plus simples les reprises sur erreur, la technique de transfert de paquets a été mise au point.

Contrairement à la commutation de circuit, celle par paquets ne réserve pas le chemin physique entre les terminaux: une fois la transmission d'un paquet effectuée, le chemin est immédiatement disponible pour la transmission d'un autre paquet. Ce type de réseau est "*orienté sans connexion*" (connectionless), le chemin physique entre les deux participants à une communication étant partagé avec d'autres communications devant emprunter le même parcours.

Le paquet est une suite d'informations binaires ne pouvant pas dépasser une taille fixée à l'avance.

Ainsi, les messages des utilisateurs sont découpés en paquets ce qui facilite la transmission. Les paquets sont envoyés indépendamment les uns des autres. Les nœuds les prennent en compte pour les émettre au fur et à mesure de leur arrivée. Les paquets de plusieurs messages peuvent être multiplexes temporellement sur une même liaison.

Internet est un exemple de réseau à transfert de paquets et plus précisément un réseau à routage de paquets. Ces paquets traversent un (ou plusieurs) sous-réseaux pour atteindre leur destination, sauf bien sûr si l'émetteur se trouve dans le même sous-réseau que le récepteur. Les paquets sont routés dans les passerelles qui sont utilisées comme nœuds d'interconnexion. Ces passerelles permettent de passer d'un réseau à un autre réseau. De façon plus précise, elles transfèrent des paquets d'une entrée vers une sortie, en déterminant pour chaque paquet la meilleure route à suivre. Internet est donc un réseau routé. En d'autres termes, dans un réseau routé, chaque paquet suit sa propre route qui est à chaque instant optimisée.

Réseau à commutation de circuits *Réseau à commutation de paquets*

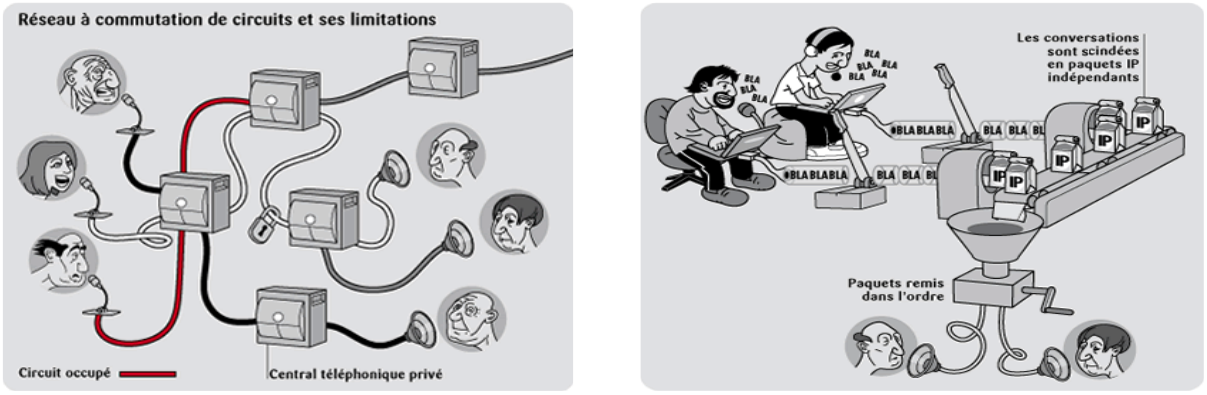

Figure 3.01 : Différence entre commutation de circuits et commutation de paquets

Il est possible de comparer la commutation de circuits et le transfert de paquets en regardant ce qui se passe sur les supports physiques.

La figure suivante décrit le cheminement de paquets dans un réseau à commutation de circuits.

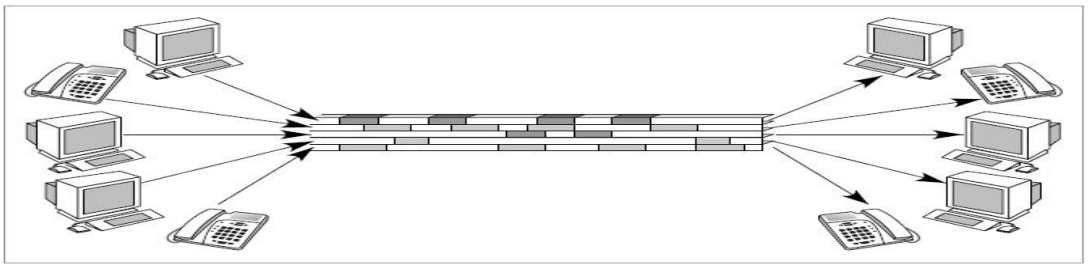

Figure 3.02 : Cheminement des paquets dans un réseau à commutation de circuits.

3.1.2. Routage

Une fois 1'adressage connu, le routage permet de trouver le meilleur chemin pour la transmission des paquets vers la destination. Comme il a été dit, la recherche du chemin s'appuie sur des algorithmes de routage. Deux grandes catégories d'algorithmes existent, les algorithmes non adaptatifs ou statiques et les algorithmes adaptatifs ou dynamiques. Dans la suite, nous nous intéressons à cette dernière catégorie qui est utilisée dans le routage Internet.

Routage adaptatif

Le routage dynamique utilise deux modes de services pour transmettre les données telles que: le *mode circuit virtuel* et le *mode datagramme*. L'Internet qui pratique le transfert de paquet utilise le mode datagramme. Notre étude s'oriente particulièrement au mode datagramme. Dans ce cas, tous les paquets portent l'adresse du destinataire («B» ou «C» dans notre exemple) et un numéro de séquence. Les paquets peuvent suivre des routes différentes puisque, pour chaque paquet, le nœud sélectionne la "bonne" route. Ainsi ces paquets peuvent arriver dans un ordre quelconque chez le destinataire parce qu'ils peuvent suivre des sauts différents.

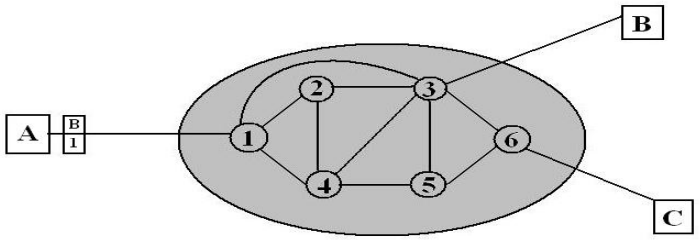

Figure 3.03 : Exemple d'un réseau basé sur les datagrammes

Ce type de réseau est robuste (des pannes sont toujours possibles) et flexible (la congestion menace toujours). A chaque nœud rencontré, il faut calculer la route vers le destinataire pour savoir vers quel voisin le paquet doit être envoyé.

Notons que les algorithmes adaptatifs modifient les décisions de routage pour traduire les variations de topologie et de trafic réel. Ils sont basés sur un échange d'informations (de tables de routage) entre les différents nœuds.

Tout d'abord, le réseau utilisant le routage dynamique doit réagir promptement au moindre changement c'est-à-dire qu'à chaque fois que la topologie d'un réseau change en raison, par exemple, de l'extension, d'une reconfiguration ou d'une panne sur le réseau, la représentation du réseau doit également être modifiée. Ainsi, la table de routage doit être mise à jour pour les nouvelles informations.

La réussite du routage dynamique dépend de deux fonctions de base des routeurs :

- la gestion d'une table de routage,
- la distribution opportune des informations aux autres routeurs sous la forme de mises à jour du routage.

Le routage dynamique s'appuie sur un protocole de routage pour partager les informations entre les routeurs. Un protocole de routage décrit les éléments suivants :

- diffusion périodique sur le réseau des informations de routage,
- échange des informations de routage,
- mise à jour de la table de routage.

En un mot, le protocole de routage gère la table de routage afin qu'elle contienne à tout moment la route optimale.

Dans un routage dynamique, la stratégie dépend du trafic qui est mesuré en chaque nœud. La gestion de ces différentes informations est effectuée de différentes façons et à différents endroits :

- centralisé,
- distribué.

3.1.2.1. Routage centralisé

Le routage centralisé est caractérisé par l'existence d'un centre qui prend les décisions quant à la définition d'une nouvelle table et de son envoi aux autres stations du réseau.

Périodiquement, chaque nœud envoie des informations vers le nœud central (exemples : la liste de ses voisins qui fonctionnent, les longueurs actuelles des files d'attente, etc.). Ce nœud central collecte toutes ces informations, et ensuite, en se fondant sur cette connaissance globale du réseau, il calcule les routes optimales d'un routeur à un routeur en utilisant un ***algorithme de plus court chemin***. En général, c'est ***l'algorithme de DIJKSTRA*** qui est le plus utilisé. Cet algorithme nécessite pour son exécution une connaissance globale du réseau.

Ainsi, la métrique utilisée pour calculer le chemin le plus court pour cet algorithme est le coût défini par avance.

Algorithme de DIJKSTRA

Cet algorithme permet le calcul de chemins optimaux d'un sommet à tous les autres :

Soit G = (X,U) le graphe représentant le réseau,

X = (1,2, ..., N) l'ensemble des nœuds du graphe,

U = (1,2, ..., M) l'ensemble des arcs du graphe,

v : U → R^+ étendu par ∀ (i, j) ∈ X^2 - U, v(i, j) = + ∞

d(i) définit la valeur minimale d'un chemin de 1 à i

v : valuation des arcs du graphe (des liaisons du réseau)

S : un sommet du graphe fixé comme source

S : = 1; d(l) : = 0; Pour tout i ≠ 0 faire d(i) : = v(1, i);

Tant que S ≠ X faire

début

choisir y ∈ X-S avec d(y) minimum

S : = S ∪ {y}

pour tout i ∈ X-S faire

d(i): = MIN (d(i), d(y) + v(y,i));

fin

L'organigramme suivant résume cet algorithme de *DIJKSTRA* :

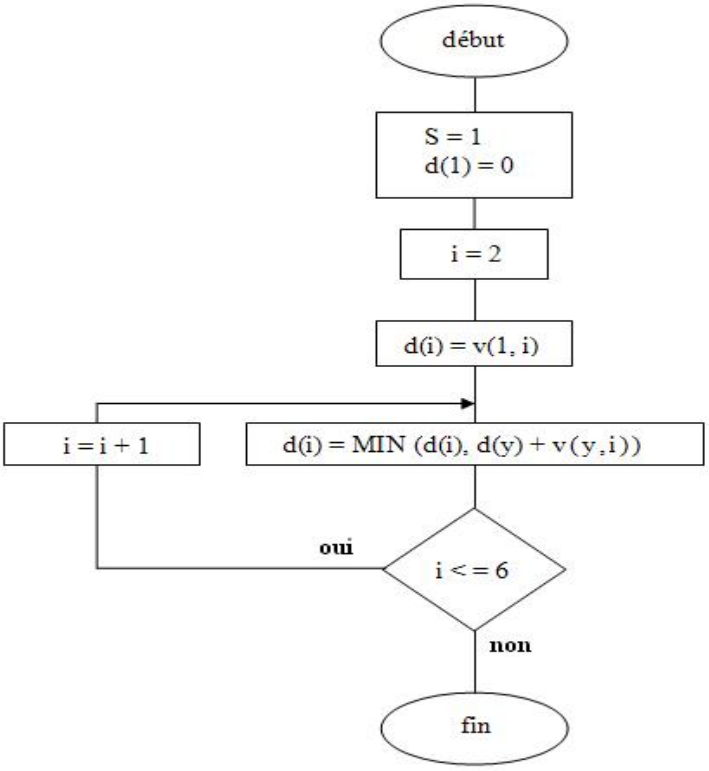

Figure 3.04 : Algorithme de DIJKSTRA

Exemple :

Soit le réseau (graphe) défini par la figure 3.05 :

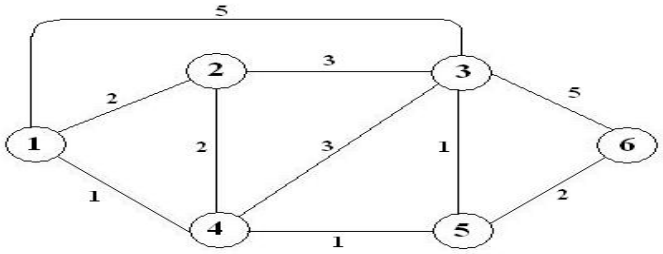

Figure 3.05 : Réseau initial (3raphe)

Nombre maximal de sauts	*Routeurs à traverser*	*d(2)*	*d(3)*	*d(4)*	*d(5)*	*d(6)*
Initial	{1}	2	5	1	∞	∞
1	{1,4}	2	4	1	2	∞
2	{1,2,4}	2	4	1	2	∞
3	{1,2,4,5}	2	3	1	2	4
4	{1,2,3,4,5}	2	3	1	2	4
5	{1,2,3,4,5,6}	2	3	1	2	4

Tableau 3.01 : Résultat de l'algorithme de DIJKSTRA

Dans le tableau 3.01, nous avons fixé le nœud 1 comme source. Le nombre maximal de sauts à effectuer est indiqué par la première colonne. La deuxième colonne montre les routeurs à traverser. Les colonnes d(i) indiquent le coût minimal entre la source et la destination. Les coûts obtenus sont les coûts minimaux c'est à dire que le coût de la liaison directe est considéré au lieu du coût quand il y a des sauts à effectuer. Les valeurs sont infinies quand il n'y a pas de liaison directe possible. Cet algorithme nécessite pour son exécution une connaissance globale du réseau, il sera donc utilisé pour des techniques implantées de façon centralisée.

3.1.2.2. Routage distribué

Pour l'algorithme de routage distribué, chaque nœud échange périodiquement avec ses voisins des informations explicites de routage. Chaque nœud tient à jour une table de routage, table qui comporte une seule entrée par nœud. Cette entrée contient deux parties : la ligne de sortie préférentielle à utiliser pour cette destination, et une estimation du temps et de la distance. L'unité utilisée peut être le nombre de sauts, le temps de transit en millisecondes, le nombre estimé de paquets dans les files d'attente pour ce chemin, ou d'autres caractéristiques du même genre.

Le nœud est supposé connaître la « distance » qui le sépare de chacun de ses voisins. Si l'unité est le saut, cette distance est exactement d'un saut. Si l'unité est la longueur de la file d'attente, le nœud examine simplement chaque file d'attente. Si l'unité est le délai, le nœud peut mesurer celui-ci directement à l'aide d'un paquet test que le récepteur se contente seulement de dater et de renvoyer le plus rapidement possible.

A titre d'exemple, supposons que l'unité utilisée soit le délai et que le nœud connaisse le délai de transfert vers chaque voisin. Toutes les T millisecondes, chaque nœud envoie à chacun de ses voisins la liste des délais estimés vers chaque destination. Il reçoit en retour une liste similaire de chacun de ses voisins. C'est l'algorithme de Ford-Fulkerson qui explique ce procédé.

3.1.2.3. Routage mixte

Notons qu'il existe aussi le routage mixte obtenu par la combinaison des deux routages cités précédemment où le routage est distribué et chaque nœud utilise des données globales de routage mises à jour de temps en temps.

3.2. Optimisation du temps de latence [10] [13] [16] [24] [31] [32]

3.2.1. Enoncé du problème

Le terme « VoIP » est en général utilisé pour décrire des communications « point à point ». Pour la diffusion de son ou de vidéos sur IP en multipoints, on parlera plutôt de streaming pour une simple diffusion, comme les radios Web par exemple. Le terme multipoints sera réservé à des visioconférences dont le nombre de participants est plus grand que deux.

La voix ou le son sur IP peut se faire en mode Unicast, broadcast ou Multicast sur les réseaux, c'est-à-dire en mode « point à point », en mode « une émission et plusieurs réceptions » (comme un émetteur TV par exemple) et en mode « une émission pour plusieurs réceptions » (mais le signal n'est routé que s'il y a des récepteurs) comme les radios Web. Le protocole H.323 ne fonctionne qu'en mode Unicast.

Le transport de communication sur IP est très dépendant du délai de latence d'un réseau. Ce délai influe beaucoup sur la qualité psycho-acoustique d'une conversation. Avec l'avènement des réseaux 100 Mbps et ADSL, les temps de latence deviennent acceptables pour une utilisation quotidienne de la voix sur IP. À l'inverse, les connexions par liaison satellite souffrent d'un temps de latence souvent trop important pour prendre en charge les applications de voix sur IP. En

moyenne, le temps de latence sur ce type de liaison est estimé entre 400 et 800 ms. Une connexion filaire (fibre optique ou cuivre) bénéficie d'un temps de latence de 60 à 200 ms. Plus que la latence, c'est la gigue qui pénalise la voix sur IP. En effet, s'il y a des fluctuations du signal en amplitude et fréquence il faudra un mécanisme de remise en ordre des paquets afin de restituer le message vocal, processus qui se traduira par des blancs et des attentes.

Bref, tous les problèmes reposent sur la qualité de service. Plusieurs paramètres entrent en jeu dans la définition de cette qualité de service. Ces critères sont divers et variés, mais on peut néanmoins les catégoriser aisément selon la qualité sonore, la qualité de l'image, le synchronisme ou bien le temps réel.

Face à ces paramètres, deux problèmes génériques peuvent survenir lors du transport des données multimédia : les retards de transmission et les défauts de transmission.

Le tableau suivant nous montre de façon synthétique la sensibilité des données multimédia face à ces problèmes :

	Sensibilité aux retards de transmission	***Sensibilité aux défauts de transmission***
Données	Faible	Forte
Voix	Forte	Faible
Image fixe	Faible	Forte
Vidéo	Forte	Moyenne

Tableau 3.02: Sensibilité des données multimédia

On s'aperçoit davantage que la vidéo qu'on aura sur Internet sera très sensible aux retards de transmission et aux défauts de celle-ci.

La qualité d'un service est une notion subjective. Selon le type d'un service envisagé, la qualité pourra résider dans le ***débit*** (téléchargement ou diffusion vidéo), le ***délai*** (pour les applications interactives ou la téléphonie), la ***disponibilité*** (accès à un service partagé) ou encore le ***taux de pertes de paquets*** (pertes sans influence pour de la voix ou de la vidéo, mais critiques pour le téléchargement).

Dans le réseau, les informations sont transmises sous la forme de paquets, petites unités de transmission transmises de routeur en routeur jusqu'à la destination. Tous nos traitements vont donc s'opérer sur ces paquets.

La mise en place de la qualité de service nécessite en premier lieu la reconnaissance des différents services. Celle-ci peut-se faire sur la base de nombreux critères :

- La source et la destination du paquet ;
- Le protocole utilisé (UDP, TCP, ICMP ...) ;
- Les ports de source et de destination dans le cas des protocoles TCP et UDP ;
- La date et l'heure ;
- La congestion des réseaux ;
- La validité du routage (gestion des pannes dans un routage en cas de routes multiples par exemple) ;
- La bande passante consommée ;
- Les temps de latence.

3.2.2. Solutions

En fonction de ces critères, différentes stratégies peuvent ensuite être appliquées pour assurer une bonne qualité de service :

3.2.2.1. Choix des routes

Lorsque plusieurs routes vers une destination sont disponibles, le choix d'une des routes peut se faire pour garantir la qualité de services. Par exemple, une route proposant délai faible mais débit faible sera utilisée pour les applications interactives, tandis qu'une route acceptant un meilleur débit au prix d'un délai plus long sera préférée pour les applications moins sensibles au délai (Streaming, téléchargement …).

3.2.2.2. Mise en forme du trafic

Mettre en forme un trafic (Traffic shaping) signifie prendre des dispositions pour s'assurer que le trafic ne dépasse jamais certaines valeurs prédéterminées. Pratiquement, cette contrainte s'applique en délayant certains paquets pour forcer un certain trafic, selon divers algorithmes.

Le contrôle du trafic peut-être utile pour limiter l'engorgement et assurer une latence correcte. Par ailleurs, des limitations de débits séparément aux trafics permettent en contrepartie de leur assurer en permanence un débit minimum, ce qui peut être particulièrement intéressant pour un fournisseur d'accès par exemple, souhaitant garantir une certaine valeur du débit à ses clients.

3.2.2.3. Ingénierie de trafic

L'ingénierie de trafic (Traffic Engineering) correspond à l'assignation des flux de trafics sur une topologie physique selon différents critères. Les procédures de contrôle offertes par les protocoles de routage mis en œuvre pour internet ne sont plus adéquates pour les nouveaux réseaux émergents.

En effet, les IGP basés sur le plus court chemin, par exemple, contribuent significativement aux problèmes de congestion dans les systèmes autonomes dans internet.

Les algorithmes SPF optimisent généralement les chemins en se basant sur la topologie et non sur la disponibilité de ressources ou sur les caractéristiques de trafic.

En conséquence, des problèmes de congestion apparaissent fréquemment lorsque :

- les chemins les plus courts pour de multiples flux de trafics convergent vers des liens ou interfaces de routeurs spécifiques ;
- un flux particulier de trafic est routé à travers un lien dont la bande passante est insuffisante.

L'ingénierie de trafic se penche sur ce problème en essayant d'optimiser l'utilisation des ressources du réseau. Ceci en vue de partager la charge entre les liens (load balancing), de rediriger le trafic vers des liaisons peu ou pas utilisées ou encore de prévoir un contrôle précis du re-routage de trafic en cas d'incident sur le chemin primaire.

Le *traffic engineering* améliore ainsi la QoS des flux des trafics par la minimisation des pertes de paquets et des délais de transmission, la maximisation des débits utiles (*throughput*) et l'introduction de la notion de priorité. En outre, il permet aux opérateurs de mieux répondre aux attentes de leurs clients en leur offrant plus d'options de raccordement.

3.2.2.4. Calcul du chemin

Afin de pouvoir gérer les critères de bande passante et de priorité introduits par l'ingénierie de trafic, des modifications ont été apportées au niveau de l'algorithme de calcul du plus court chemin sur lequel se base le choix des routes dans OSPF et IS-IS.

3.2.2.5. Ordonnancement

L'ordonnancement désigne l'ensemble des méthodes visant à modifier l'ordre de départ des paquets, en complément de la règle du « premier arrivé, premier servi ».

Une de ses applications les plus courantes consistera ainsi à donner priorité à certains types de trafic, de façon sommaire en ne laissant passer du trafic de faible priorité que s'il n'y a plus de trafic de forte priorité, ou de façon plus fine avec des algorithmes de Round-Robin pondérés, visant à faire passer des paquets des différentes connexions tour à tour, en laissant plus de temps aux paquets prioritaires.

Une autre application, le Fair Queuing consiste à séparer nettement les connexions, et à leur attribuer successivement et équitablement une possibilité de faire passer leurs paquets : cela permet de s'assurer qu'aucune application, même très demandeuse de débit, n'en écrasera d'autres.

3.2.2.6. Matériel

La qualité de service se réalise au niveau de la couche 3 du modèle OSI. Elle doit donc être configurée sur les routeurs ou la passerelle reliée à Internet.

3.2.2.7. Techniques d'amélioration de routage

Dans le domaine de routage, il faut avoir une bonne stratégie pour aboutir à un ***résultat fiable*** et une ***bonne qualité de service***. Cependant, des techniques sont mises en place pour gérer la circulation de trafic à l'intérieur du réseau à savoir le ***contrôle de flux***, et ***contrôle de congestion***. Les techniques de contrôle de flux et de congestion sont capitales dans le monde des réseaux. Les réseaux à transfert de paquet sont comme les autoroutes : s'il y a trop de paquets (voitures), plus personne ne peut avancer. Il faut donc contrôler le réseau et les flux qui y circulent. Le contrôle de flux est préventif : il doit limiter les flots d'informations à la capacité de transport du support physique. Le contrôle de congestion a pour objet d'éviter la congestion des nœuds et de résoudre les embouteillages lorsqu'ils sont effectifs.

Les deux termes, contrôle de flux et contrôle de congestion, peuvent être définis de manière plus précise. En effet, le contrôle des flux est un accord entre deux entités, la source et la destination, pour limiter le débit de transmission du service en considérant les ressources disponibles dans le réseau. Le contrôle de congestion représente l'ensemble des actions entreprises pour éviter et éliminer les congestions causées par manque de ressources. Ainsi, selon ces définitions, le contrôle de flux peut être considéré comme un cas spécial du contrôle de congestion.

De très nombreuses méthodes ont été testées, elles sont valides dans des cas spécifiques. Le contrôle s'effectue par une contrainte sur le nombre de paquets qui circulent dans le réseau. Cette limitation s'exerce soit sur le nombre de paquets en transit, d'une entrée à une sortie ou sur l'ensemble du réseau, soit sur le nombre de paquets ayant le droit d'entrer à l'intérieur du réseau

par unité de temps. A ces contrôles peuvent s'ajouter des techniques d'allocation des ressources qui garantissent qu'il n'y aura pas de congestion.

Il est clair que le contrôle de flux influence de façon prépondérante le comportement de l'algorithme de routage utilisé, et vice-versa. En effet, il existe une forte corrélation entre le routage et le contrôle de flux, le contrôle de congestion doit être jugé en tenant compte des objectifs du réseau : charge maximum avec temps de réponse minimum, plus précisément la bonne performance du réseau.

L'***algorithme de Dijkstra*** n'est pas encore implémenté partout. Pour le moment, le réseau Internet utilise l'algorithme RIP pour déterminer le chemin à suivre pour transmettre des données entre deux serveurs. Mais Dijkstra commence à s'implanter là aussi et à remplacer progressivement ce protocole pour la bonne et simple raison qu'il est le plus rapide à proposer une solution au problème des plus courts chemins.

Une application des plus courantes de l'algorithme de Dijkstra est le protocole OSPF qui permet un routage internet très efficace des informations en cherchant le parcours le plus efficace.

Comme nous avons vu précédemment, les caractéristiques des protocoles permettant d'acheminer les paquets d'un émetteur vers un récepteur varient considérablement. Il peut être décidé, par exemple, que tous les paquets passent par un même chemin. À l'inverse, chaque paquet peut être livré à lui-même dans le réseau.

Dans ce cas, le paquet porte simplement l'adresse du destinataire et choisit sa route à chaque nœud jouant le rôle de carrefour. Comme plusieurs chemins permettent d'atteindre le destinataire, à l'entrée du nœud, le paquet examine les directions acceptables et choisit en fonction des embouteillages qu'il aperçoit.

Cette situation ne va pas sans inconvénients. Le message d'un utilisateur étant souvent décomposé en plusieurs paquets, ce que l'on nomme un *flot,* ces derniers peuvent arriver d'une façon désordonnée. De plus, les nœuds de transfert deviennent complexes, puisqu'ils doivent décider dans quelle direction envoyer les paquets.

En contrepartie, la souplesse augmente. Si une route se trouve coupée ou fortement congestionnée, le paquet peut prendre une autre direction. Dans le cas d'un *routage fixe,* une coupure peut avoir des conséquences fâcheuses, puisqu'il faut ouvrir une nouvelle route, ce qui peut se révéler difficile. En dépit de ce manque de flexibilité, le routage fixe reste une solution simple et performante.

Pour résoudre les problèmes de congestion potentiels, il faut être capable de gérer les flots de paquets en leur interdisant d'entrer dans le réseau ou en les faisant passer par des chemins détournés lorsque les nœuds les plus utilisés sont saturés.

3.3. Conclusion

Un des problèmes principaux de la technologie de ToIP est de contrôler correctement l'écho et le délai tout en minimisant les dégradations causées par les pertes de paquets, et de préserver une bonne interactivité de la conversation. Heureusement, avec la technologie actuelle, il est possible de contrôler ces problèmes.

Pour le contrôle de l'écho, deux types de dispositifs sont couramment utilisés : les suppresseurs d'écho et les annuleurs d'écho. Dans notre cas, on utilisera des annuleurs d'écho.

De même, le délai peut être contrôlé grâce à la mise au point d'une série d'algorithmes de gestion de files d'attente pour assurer un partage équitable de la capacité. Ces algorithmes permettent aux réseaux IP d'offrir le même type de qualité de service que les réseaux de type ATM.

Le contrôle de la gigue est important principalement pour les applications temps réel qui nécessitent l'utilisation de mémoires tampons afin de restituer de manière régulière une information qui arrive de manière irrégulière.

Le problème de perte de paquets est très lié au problème de la bande passante, ainsi qu'à l'utilisation appropriée d'algorithmes de contrôle de la congestion en bordure et dans le cœur du réseau.

Ainsi, la solution proposée pour améliorer cette qualité de service est de modéliser le réseau IP à traverser par les paquets IP sous forme de graphe afin de chercher le chemin le plus court entre les routeurs en tenant compte des files d'attente et des congestions.

Enfin, le routage final dévie le trafic des chemins les plus courts calculés par l'algorithme Dijkstra vers des chemins plus longs moins chargés. Pour pouvoir le faire, les nœuds doivent avoir une connaissance de la capacité et de la charge des différents liens du réseau. Ces informations seront distribuées en utilisant les protocoles de routage à états de liens.

Les seuls protocoles à état de liens pouvant supporter l'ingénierie de trafic sont OSPF et IS-IS auxquels sont rajoutées quelques extensions pour l'ingénierie de trafic.

On suggère donc l'utilisation d' « algorithmes de routage basés sur les contraintes ».

CHAPITRE 4 SIMULATION SOUS C++

Le travail de simulation a pour objectif le développement d'un logiciel de routage IP. Ce logiciel a été réalisé à l'aide de Visual C++, pourtant il peut fonctionner sur toutes les plates-formes Windows.

4.1. Présentation du logiciel [18] [29] [33] [34]

4.1.1. But du logiciel

D'après tout ce qu'on a vu dans les chapitres précédents, on a bien remarqué que la qualité de service (QoS) est très importante pour la téléphonie sur IP pour éviter la dégradation de la conversation téléphonique qui peut avoir plusieurs causes : délai trop long de l'acheminement des paquets, buffer de gigue, perte de paquets...

Actuellement, en dépit de la concurrence entre les différents fournisseurs de service Internet, les clients ont la chance de choisir l'opérateur qui puisse leur donner le plus de satisfaction pour leurs besoins, plus précisément une bonne qualité de service. Par conséquent, les opérateurs doivent améliorer de plus en plus la performance de leurs réseaux pour être toujours parmi les meilleurs au niveau du marché concurrentiel.

C'est la raison pour laquelle, nous avons développé un logiciel qui a pour but d'évaluer la performance du réseau, d'estimer la qualité de service (QoS) du côté des utilisateurs. Cette solution permet à la fois d'optimiser le temps de latence (recherche du plus court chemin) et de réduire les gigues et pertes de paquets (gestion des files d'attente avec contrôle de congestion).

Ainsi, la solution prise s'applique au niveau du réseau Internet ou WAN IP, c'est-à-dire dans le passage des paquets dans les différents routeurs du réseau IP.

Considérons le réseau de téléphonie sur IP représenté par la figure 4.01 ci-dessous. Par exemple, si l'IP-Phone A veut appeler l'IP-Phone D, les paquets IP (signaux de parole) doivent traverser le réseau Internet ou un réseau étendu WAN où il existe plusieurs routeurs.

Différentes hypothèses doivent être définies d'avance :

- Le réseau internet est constitué de plusieurs routeurs ;
- Le passage d'un paquet dans chaque routeur prend un temps de traitement ;
- Chaque traitement de paquet (compression, mise en paquet,…) prend un délai ;
- Les terminaux IP Phones les plus récents possèdent des algorithmes d'annulation d'écho…

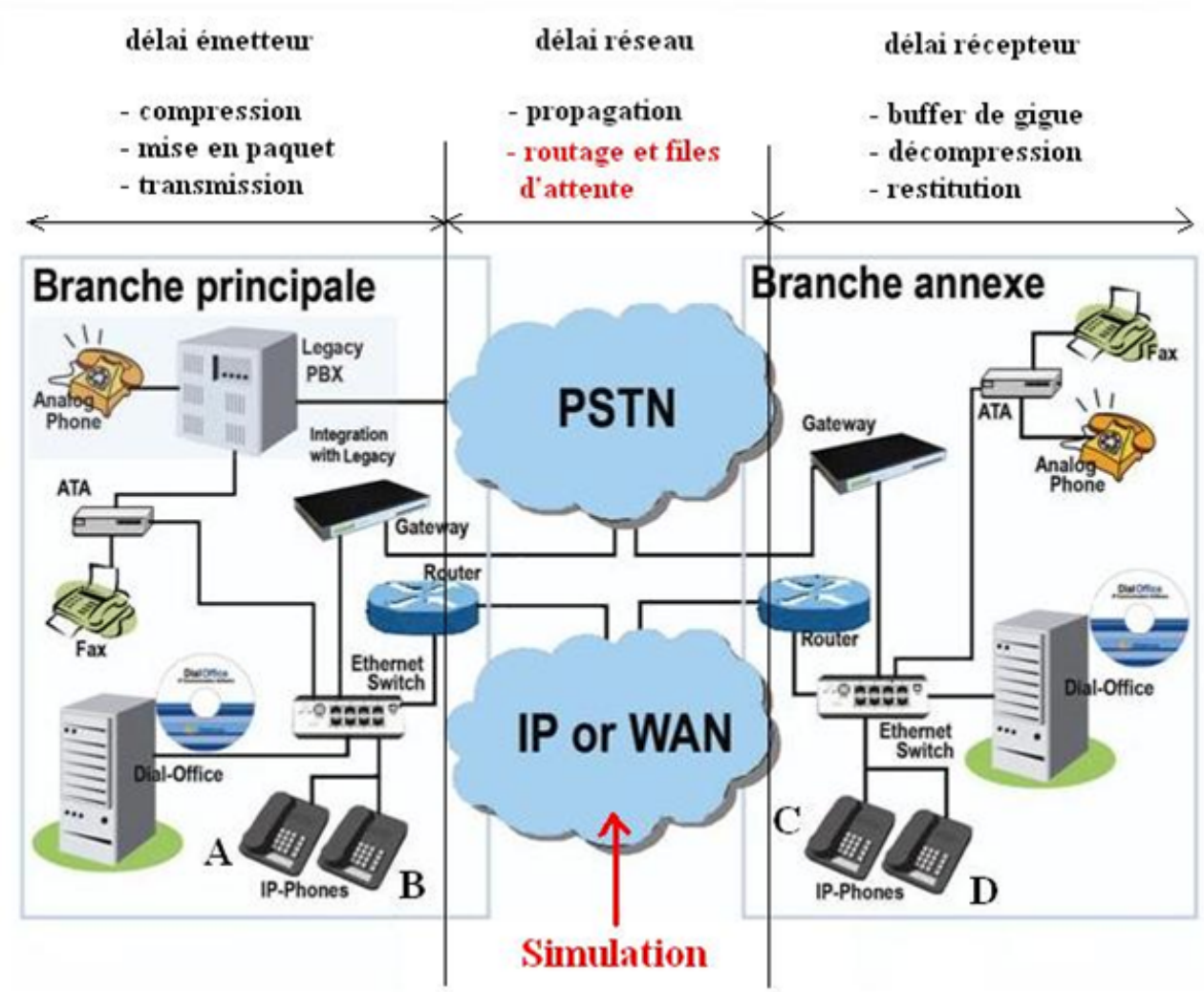

Figure 4.01 : Paramètres pour l'optimisation

4.1.2. Spécifications du logiciel

On a effectué une *simulation de routage IP*. Comme nous l'avons vu au chapitre précédent, tout ce phénomène se passe dans le routeur, et est observé au cours du routage du paquet.

Connaissant le temps de traitement de chaque codec utilisé, le temps de réponse de chaque routeur du réseau et le délai de traversée de chaque lien, nous pouvons déduire le délai total d'acheminement d'un paquet de la source vers la destination, c'est-à-dire d'un terminal à un autre.

En fait, le délai à calculer est le temps pris par le paquet depuis l'IP Phone appelant vers l'IP Phone appelé. Et à chaque IP Phone correspond un nœud ou sommet (routeur) qui le relie à Internet.

4.1.3. Modèles utilisés

4.1.3.1. Gestion de files d'attente

Bien évidemment, la réussite du travail dépend de la stratégie de routage choisie. Un élément qui peut jouer en sa faveur est la gestion de files d'attente.

Ainsi, on a choisi une approche de modélisation simple de ce problème, en utilisant le réseau de file d'attente M/M/1 (voir annexe 2). Le calcul se base donc sur l'équation :

$$T_i = 1 \,/\, (\mu C_i - \lambda_i) \tag{4.01}$$

où T_i : temps de réponse du système

μC_i : taux de service en paquet par seconde

λ_i : taux d'arrivée de paquets par seconde

Puis, on s'est servi de l'algorithme de DIJKSTRA pour le calcul du plus court chemin qui nous permet de connaître les différents nœuds intermédiaires empruntés par le paquet. De plus, nous avons combiné les deux techniques de gestion des files d'attente : la technique de gestion FIFO et la technique de gestion à priorité absolue. Pour cela, le paquet qui arrive le premier sera traité le premier, mais en tenant compte de la priorité éventuelle du paquet, c'est-à-dire que le paquet coloré (prioritaire), interrompt le service en cours pour exécuter son propre service. Le service reprend avec son ordre normal lorsque le paquet prioritaire a terminé son service.

4.1.3.2. Modèle de trafic

Considérons que la loi du trafic voix obéit à un processus de Poisson. Alors la probabilité d'observer k arrivées pendant un intervalle de temps t est donnée par :

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \tag{4.02}$$

La propriété la plus importante du processus de Poisson, peut s'interpréter de la façon suivante: une arrivée poissonienne signifie que la probabilité pour qu'un client arrive pendant dt est à peu près égale λdt.

La deuxième propriété essentielle du processus de Poisson relie le processus d'arrivée poissonnien aux variables aléatoires, mesurant ainsi le temps d'inter arrivée (exponentielles). Cette propriété s'interprète de la façon suivante : des arrivées poissoniennes signifient que les inters arrivés sont de types exponentiels.

En voix sur IP, la loi la plus utilisée est celle d'Erlang B. La loi d'Erlang B calcule la probabilité de rejet d'une demande de ressource à raison de ressources non disponibles. La probabilité de blocage sera alors égale à :

$$P_n = \frac{\frac{A^n}{n!}}{\sum_0^n \frac{A^i}{i!}} = E_1(A,n) \quad (4.03)$$

où : n : nombre de ressources disponibles,

A : trafic offert

P_n : probabilité que les n ressources soient occupés

E_1 : première formule d'Erlang qui est fonction de A et n

4.1.3.3. Capacité nécessaire

La bande passante nécessaire peut se calculer à partir du nombre de circuits et le débit par appel :

$$B\text{ande} = \text{Débit}_{\text{appel}} \times N\text{ombre}_{\text{circuit}} \quad (4.04)$$

4.1.3.3.1. Débit par appel

Le débit par appel peut être calculé en tenant compte des éléments suivant :

- les codecs audio utilisés au niveau de la couche application ;
- les différentes encapsulations aux niveaux des différentes couches (transport, réseau) ;
- les protocoles au niveau de la couche liaison.

4.1.3.3.1.1. Codecs audio

Les codecs les plus utilisés pour la compression/décompression de la voix sur IP sont :

- G.711 offrant un débit de 64 Kbit/s
- G.723 offrant deux débits de 6.4 Kbit/s et 5.3 Kbit/s
- G.726 offrant un débit de 32 Kbit/s
- G.728 offrant un débit de 16 Kbit/s
- G.729 offrant un débit de 8 Kbit/s

Selon le débit généré par le codec et tenant compte des différentes possibilités des cycles de transmission, nous obtenons la taille des données audio. Ces données audio passeront à des encapsulations au niveau des différentes couches : commençant par la couche transport jusqu'à la couche liaison de données.

4.1.3.3.1.2. Encapsulations aux niveaux transport et réseau

Les données audio de la couche application sont affectées au niveau de la couche transport, d'un en-tête RTP ayant une taille minimale de 12 octets, puis d'un en-tête UDP avec 8 octets enfin la mise en paquet au niveau de la couche réseau ajoute 20 octets pour l'en-tête IP (le champ Options n'est pas considéré). La figure suivante illustre le principe de la mise en paquet.

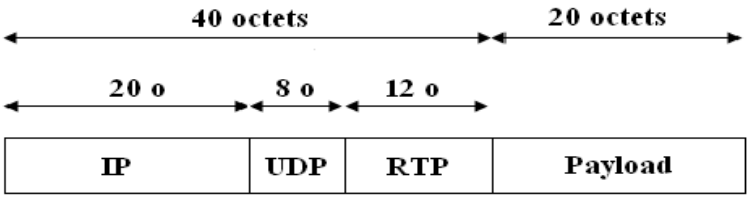

Figure 4.02 : Encapsulation RTP/UDP/IP

4.1.3.3.1.3. Protocoles utilisés au niveau de la couche liaison

L'encapsulation doit tenir compte des différents protocoles au niveau de la couche liaison : Ethernet (LAN), HDLC (High level Data Link Control), Frame Relay, PPP, ATM

La signification des différents champs n'est pas aussi importante que la taille des données qu'ajoute chaque protocole. Le débit généré sur le support physique change suivant la variation des différents paramètres cités ci-dessus.

Pour chaque site, nous supposons qu'il y a un choix uniforme entre les différents utilisateurs des différents paramètres : codec, cycle de transmission, protocole de couche liaison.

La formule de calcul du *débit par appel* est la suivante :

$$Débit_{apl} = [(Débit_{codec} \times cycle_{trans}) + entête_{RTP.UDP.IP} + entête_{protocoleliaison} + enqueue_{protocoleliaison}] \times 8/cycle_{trans} \quad (4.05)$$

où : $Débit_{apl}$: débit par appel en kbps

$Débit_{codec}$: débit généré par le codec en kbps

$cycle_{trans}$: cycle de transmission de paquet en ms

$entête_{RTP.UDP.IP}$: taille de l'entête RTP/UDP/IP à ajouter en octets

$entête_{protocoleliaison}$: taille de l'entête du protocole de couche liaison en octets

$enqueue_{protocoleliaison}$: taille de l'en-queue du protocole de couche liaison en octets

4.1.3.3.2. Nombre de circuits

L'algorithme de la formule d'Erlang inverse permet de déterminer le nombre de circuits à mettre en œuvre pour supporter un trafic donné avec une probabilité de blocage fixe.

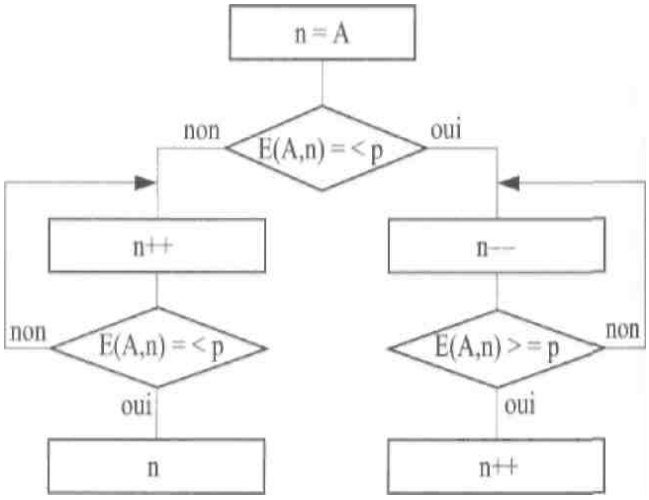

Figure 4.03 : Algorithme d'Erlang inverse

avec : A : trafic offert,

n : nombre de circuit,

E : formule d'Erlang,

p : probabilité de blocage fixé par l'opérateur.

4.1.4. Fonctionnement de l'ensemble

Par hypothèse, on a négligé les temps induits par les activités internes du routeur devant le temps de service moyen du serveur. Le serveur en question est représenté par les lignes de sortie du routeur, et le routeur représente le buffer qui est la file d'attente des paquets.

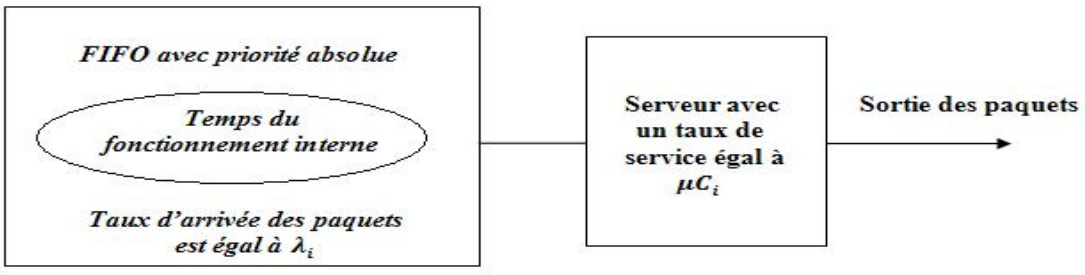

Figure 4.04 : Fonctionnement de l'ensemble du système

Le routage des paquets est résumé par l'organigramme suivant :

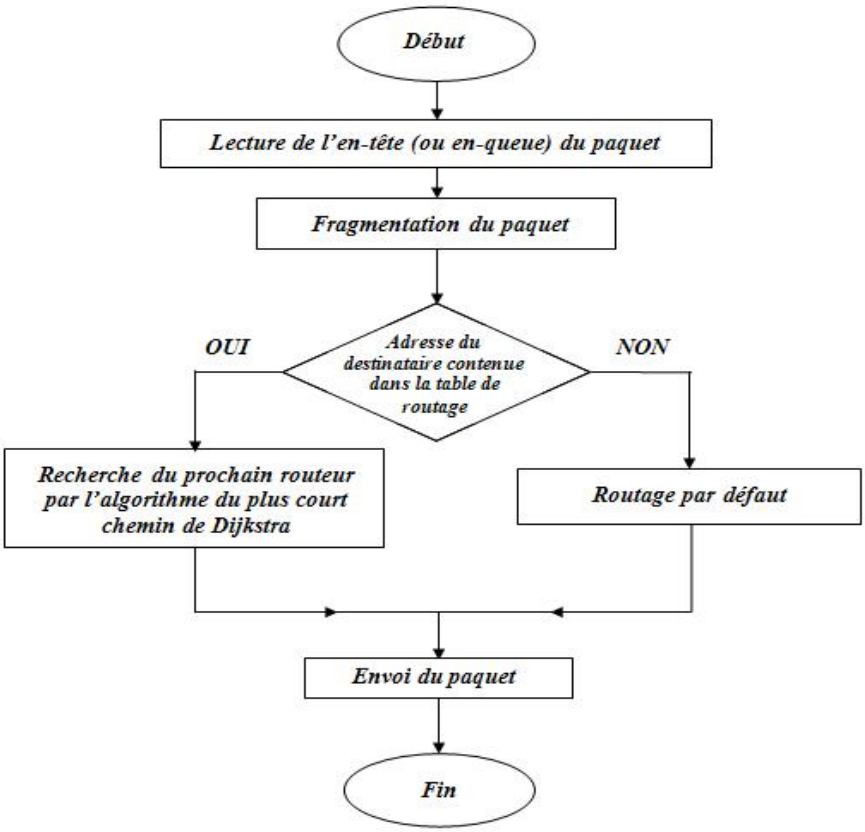

Figure 4.05 : Organigramme simplifié du routage IP

4.1.5. Algorithme final utilisé

L'algorithme utilisé en simulation permet le routage basé sur les contraintes.

Cet algorithme est défini comme suit :

- ignorer les liens qui offrent une bande passante inférieure à celle requise ;
- ignorer les liens qui ne correspondent pas à l'affinité signalée ;
- exécuter l'algorithme Dijkstra sur la topologie restante en tenant compte des métriques de l'IGP (OSPF ou IS-IS) ;
- si plusieurs chemins subsistent, choisir celui avec le minimum de bande passante totale disponible sur tous les liens constituant le chemin ;
- s'il y a encore plusieurs chemins, sélectionner celui avec le minimum de sauts ;
- s'il existe toujours plusieurs routes, en choisir une aléatoirement.

4.2. Mode d'utilisation du logiciel [29] [30] [31] [32] [35]

Considérons le réseau de la figure 4.06 suivant :

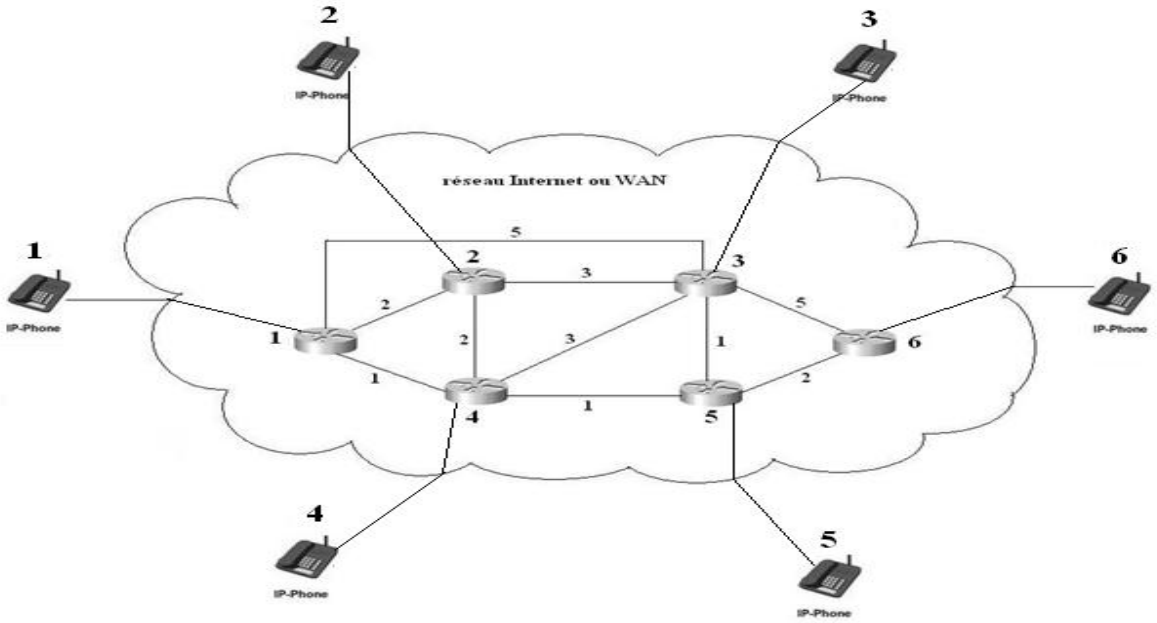

Figure 4.06 : Réseau de 6 IP Phones (6 LAN)

4.2.1. Les données du réseau

Nombre de nœuds ou sommets (routeurs à traverser dans le réseau ou ordre) : 6

Nombre d'arcs (liens possibles entre les différents routeurs) : 10

4.2.1.1. Les nœuds du réseau

Nœud n°1 : Routeur 1 (correspond au réseau 192.168.1.x)

Nœud n°2 : Routeur 2 (correspond au réseau 192.168.2.x)

Nœud n°3 : Routeur 3 (correspond au réseau 192.168.3.x)

Nœud n°4 : Routeur 4 (correspond au réseau 192.168.4.x)

Nœud n°5 : Routeur 5 (correspond au réseau 192.168.5.x)

Nœud n°6 : Routeur 6 (correspond au réseau 192.168.6.x)

4.2.1.2. Les arcs du réseau

[Coût] : distance

[Capacité maximale] : flots

Arc 1 :	Routeur 1	→	Routeur 2	coût : 2	capacité maximale : 10
Arc 2 :	Routeur 1	→	Routeur 3	coût : 5	capacité maximale : 10
Arc 3 :	Routeur 1	→	Routeur 4	coût : 1	capacité maximale : 10
Arc 4 :	Routeur 2	→	Routeur 3	coût : 3	capacité maximale : 10
Arc 5 :	Routeur 2	→	Routeur 4	coût : 2	capacité maximale : 10
Arc 6 :	Routeur 3	→	Routeur 4	coût : 3	capacité maximale : 10
Arc 7 :	Routeur 3	→	Routeur 5	coût : 1	capacité maximale : 10
Arc 8 :	Routeur 3	→	Routeur 6	coût : 5	capacité maximale : 10
Arc 9 :	Routeur 4	→	Routeur 5	coût : 1	capacité maximale : 10
Arc 10 :	Routeur 5	→	Routeur 6	coût : 2	capacité maximale : 10

4.2.2. Simulation

Supposons que l'IP Phone 1 (192.168.1.1) veut appeler l'IP Phone 6 (192.168.6.1), pour éviter le temps de latence trop long et les pertes de paquets, les paquets doivent prendre le chemin le plus court depuis le routeur 1 (source) jusqu'au routeur 6 (destination finale).

Au lancement du logiciel apparaît une première fenêtre (voir figure 4.07) :

Figure 4.07 : Fenêtre d'accueil

Pour commencer la simulation, on clique sur le bouton *Commencer* ; et une nouvelle fenêtre s'ouvre.

C'est dans cette fenêtre qu'on entre le nombre de routeurs à traverser dans le réseau (on parle aussi d'ordre ou nombre de nœuds) et le nombre de liens possibles dans le réseau.

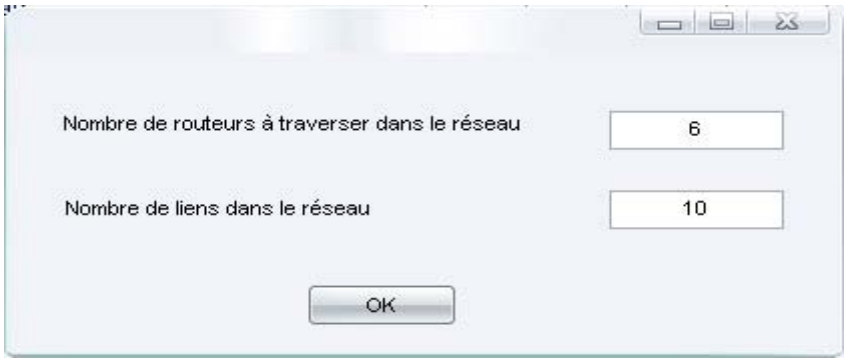

Figure 4.08 : Dimension du réseau

Ensuite, lorsqu'on valide (bouton *OK*) le nombre de routeurs et le nombre de liens dans le réseau, une nouvelle fenêtre s'ouvre pour paramétrer les routeurs.

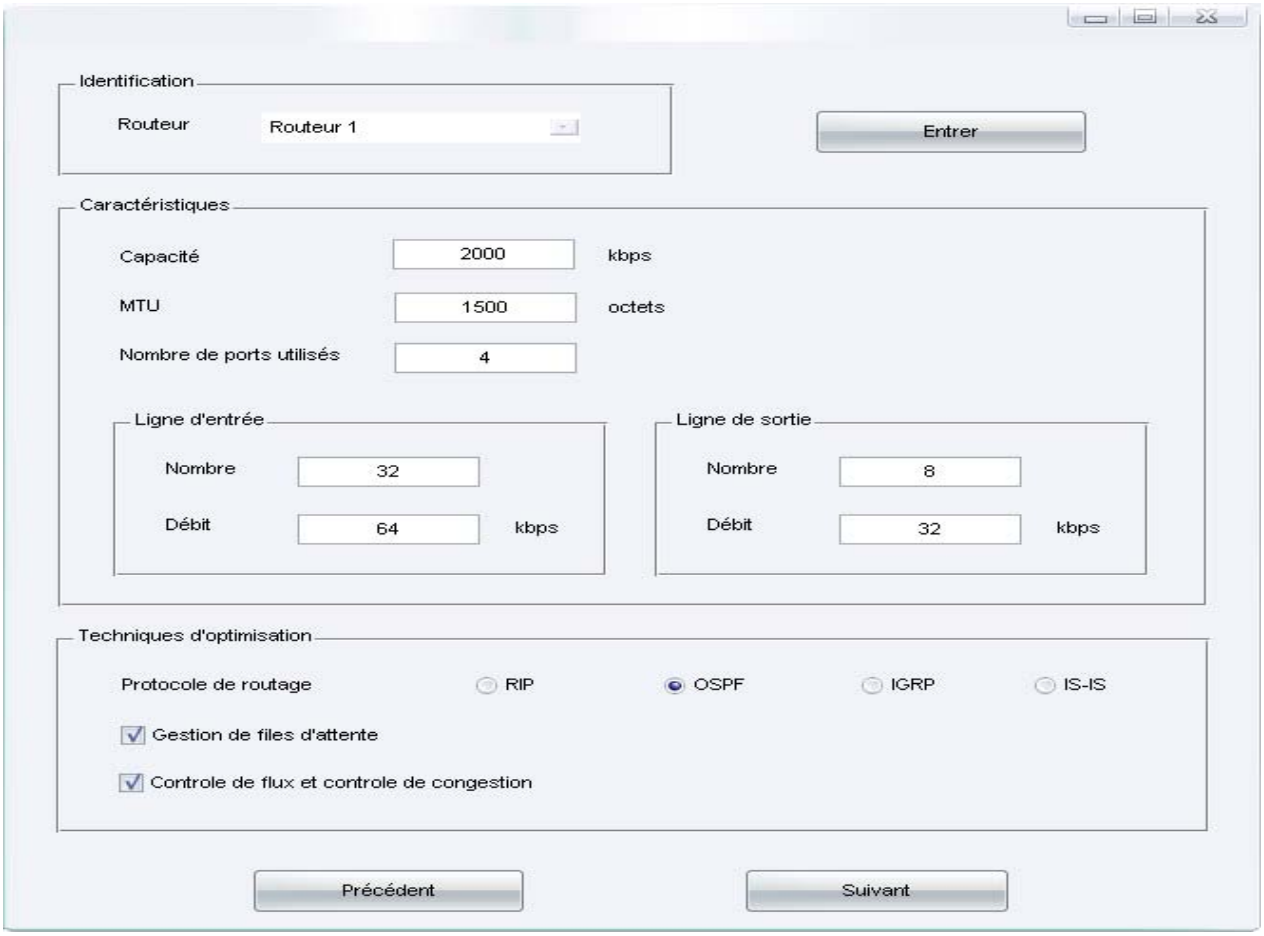

Figure 4.09 : Paramètres d'optimisation de la QoS

Cette fenêtre correspond à la configuration des routeurs du réseau pour l'optimisation, c'est-à-dire paramétrer les routeurs pour qu'ils gèrent les files d'attente avec des contrôles de flux et de congestion. De même, pour choisir le protocole de routage à utiliser.

Puis, lorsqu'on termine les paramétrages, on clique sur *Suivant* pour passer à la fenêtre de routage où on saisit toutes les données du réseau.

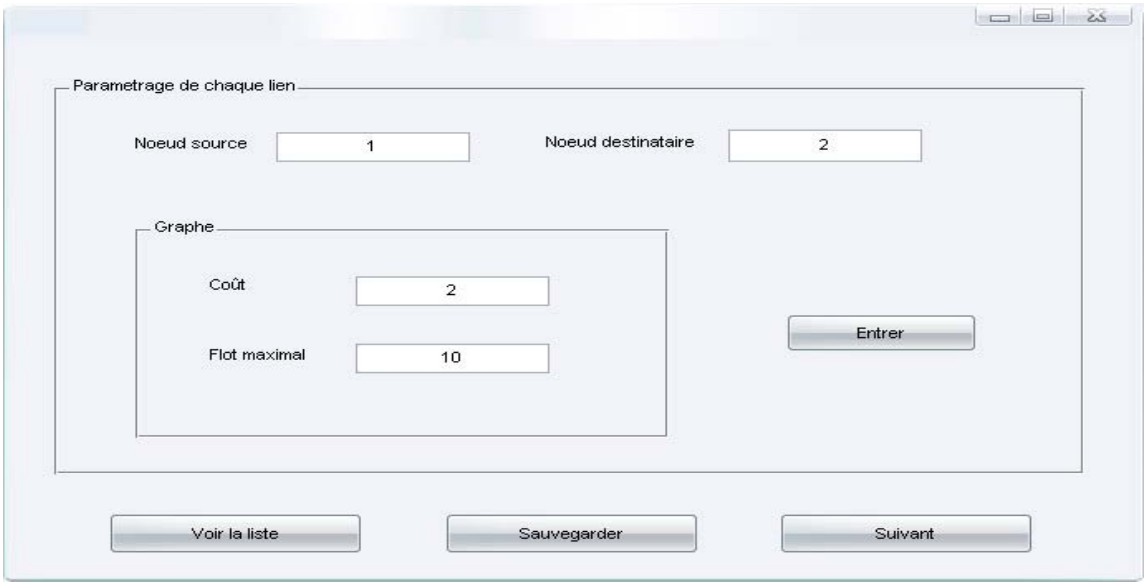

Figure 4.10 : Données du réseau

Nœud source, *Nœud destinataire* et *Coût* correspondent aux paramètres de chaque lien du réseau. Après chaque saisie (2 nœuds avec leur coût ou distance), on clique sur le bouton *Entrer* et toutes les données entrées sont enregistrées dans un fichier texte. Et il suffit de les vérifier en cliquant sur le bouton *Voir la liste*.

Flot maximal correspond à la capacité maximale qu'un lien physique peut supporter.

Les données saisies sont enregistrées dans un fichier texte.

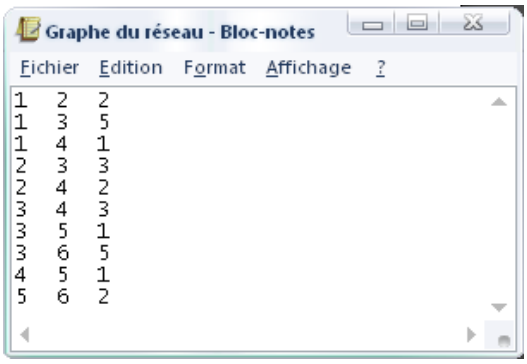

Figure 4.11 : Liste des données saisies

Les données saisies sont disposées en 3 colonnes :

1ère colonne : nœud source ; 2ème colonne : nœud destinataire ; et 3ème colonne : poids (coût)

Après avoir vérifié toutes les valeurs entrées, on clique sur *Suivant* pour arriver à la fenêtre d'appel.

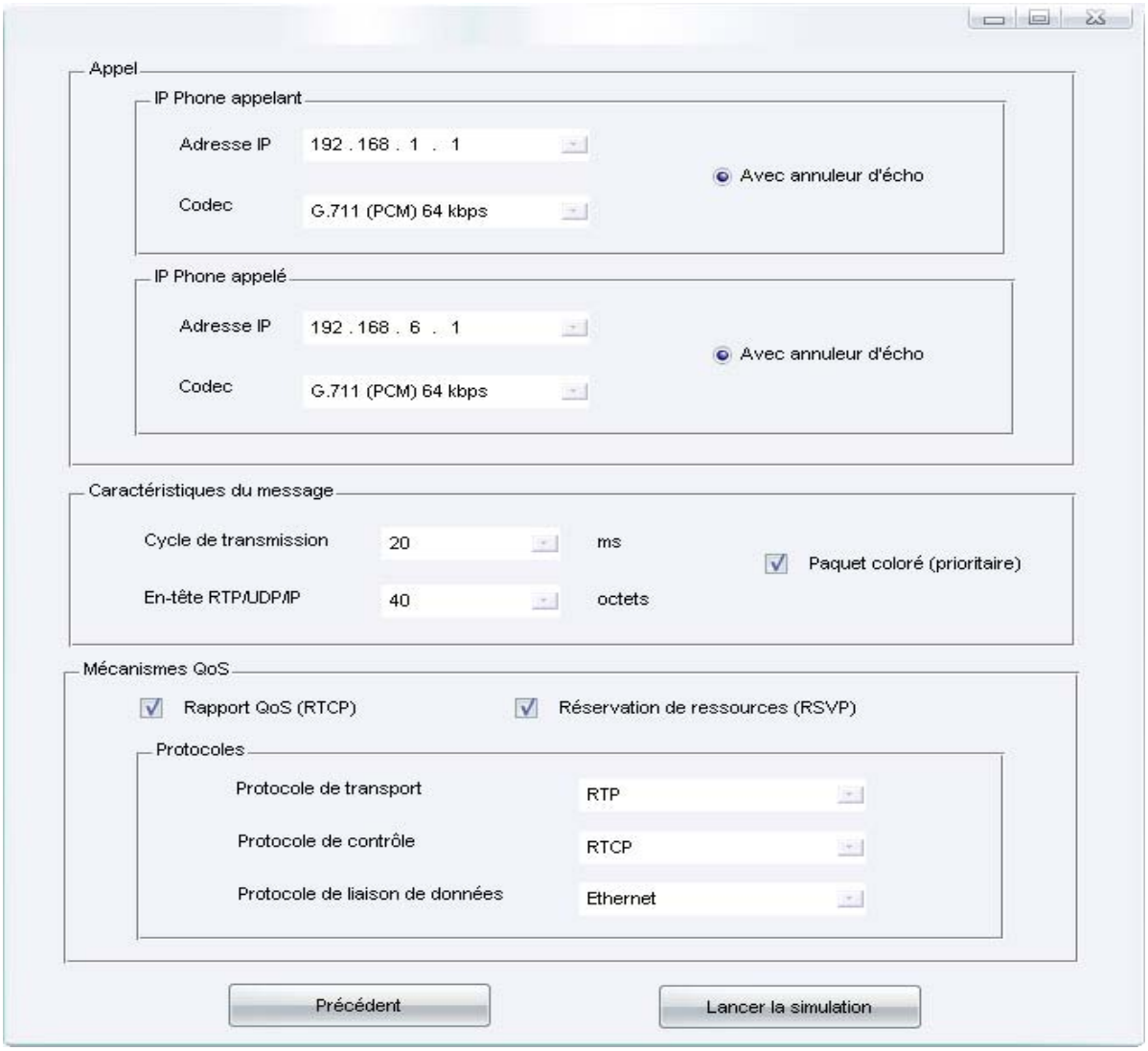

Figure 4.12 : Paramétrage de l'appel

Dans la zone Appel : on entre dans *IP Phone Appelant* et *IP Phone Appelé* les adresses IP des téléphones qui veulent se communiquer et le type de codec utilisé. Ici, on considère que le terminal utilise un annuleur d'écho, cela signifie que le délai induit par les échos n'est pas considéré.

Caractéristiques du message permet de définir le cycle de transmission du paquet (intervalle d'émission entre deux paquets) à émettre et la longueur de l'entête RTP/UDP/IP ainsi que de préciser si le paquet est coloré, c'est-à-dire que si le paquet possède une certaine priorité par rapport aux autres.

Ensuite, si tous les paramétrages sont terminés, on clique sur le bouton *Lancer la simulation*.

Figure 4.13 : Résultat de l'appel l'IP Phone 1 vers l'IP Phone 6

Enfin, pour voir la signification des valeurs affichées, on clique sur le bouton *Interprétation des résultats* et une boîte de dialogue s'ouvre pour l'indiquer.

Les champs *Source* et *Destination finale* correspondent aux terminaux IP Phones appelant et appelé entrés dans la fenêtre précédente.

Le *Chemin à suivre* affiche les différents nœuds (routeurs) que le paquet a traversés depuis sa source vers sa destination finale dans le réseau. C'est le plus court chemin en considérant les poids (coût ou distance) entrés entre les nœuds.

Le *Délai d'acheminement* est le temps de traversée des paquets depuis le premier routeur traversé jusqu'au dernier routeur en considérant plusieurs contraintes dans les différents liens du réseau : saturation du réseau (file d'attente), distance, accessibilité des routeurs, etc.

Le *Temps de traitement de la voix* varie selon le codec utilisé.

Le *Temps de latence* correspond à la somme de tous les délais considérés.

C’est ce temps de latence qui nous intéresse le plus dans cette simulation. On a considéré les délais de traitement à l’émetteur et au récepteur (autre que les traitements au niveau des codecs) sont négligeables par rapport au délai de traversée du réseau IP. C’est pourquoi on n’a considéré que le temps de passage dans les routeurs et les temps passés en file d’attente.

Gigue de phase et *Taux de perte* varient selon les contraintes du réseau.

Capacité réservée correspond à la bande passante occupée par le paquet pendant son passage dans le réseau.

Indice de dégradation montre le taux de dégradation de tout le signal de parole émis et reçu au destinataire (conversation).

Enfin, pour voir la signification des résultats :

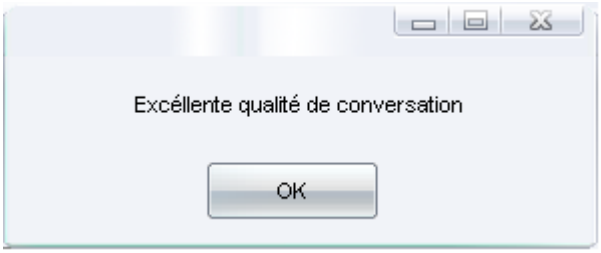

Figure 4.14 : Interprétation des résultats (délai < 150 ms)

4.3. Conclusion

On peut déduire que la technique de gestion de files d’attente est une technique qui s’avère efficace pour l’évaluation, voire l’amélioration de la performance du système, et de la qualité de service du côté des utilisateurs. Comme on a dit précédemment, la plus grande contrainte de la téléphonie sur IP est le délai de bouche à oreille, qui est déjà dans la plupart des cas dans la limite de ce qui est considéré comme acceptable par un utilisateur normal. Mais une fois que l’écho et le délai sont maintenus dans les limites correctes par une architecture réseau bien étudiée, la voix sur IP permet d’obtenir une qualité vocale supérieure à celle des réseaux téléphoniques actuels.

Ainsi, en combinant plusieurs techniques : recherche du plus court chemin par l’algorithme de Dijkstra, routage IP basé sur les contraintes, gestion de files d’attente, contrôle de congestion, contrôle de flux, réservation de ressources, nous avons pu optimiser le temps de latence et d’avoir une bonne qualité de service du côté des utilisateurs.

CONCLUSION GENERALE

Actuellement, il est évident que la téléphonie sur IP va continuer de se développer dans les prochaines années. Le marché de la ToIP est très jeune mais se développe à une vitesse fulgurante. C'est aujourd'hui que les entreprises doivent investir dans la ToIP si elles veulent y jouer un rôle majeur. La ToIP ouvre la voie de la convergence voix/données et celle de l'explosion de nouveaux services tels que les CTI.

La ToIP est une bonne solution en matière d'intégration, de fiabilité, d'évolutivité et de coût. Elle fera partie intégrante des Intranets d'entreprises dans les années à venir et apparaîtra aussi dans la téléphonie publique pour permettre des communications à bas coût. Bientôt nous téléphonerons tous sur IP...

Enfin, on peut ainsi vraisemblablement penser que le protocole IP deviendra un jour un standard unique permettant l'interopérabilité des réseaux mondialisés. C'est pourquoi l'intégration de la voix sur IP n'est qu'une étape vers EoIP : Everything over IP.

Toutefois, Madagascar semble encore être en retard de quelques années en matière de ToIP. En effet, la ToIP n'est pas encore exploitable dans le monde des entreprises malgaches. Le problème de licence au niveau de l'Office Malagasy d'Etudes de Régulation des Télécommunications (OMERT) en est la première raison, mais également à cause des tarifs des équipements VoIP onéreux et des faibles débits de connexion Internet. Pour l'utilisation à titre personnel, des offres de téléphonie gratuits existent sur Internet, néanmoins, la qualité est très mauvaise vu le débit Internet grand public qui tourne au mieux de 100 Kbps. Il est pourtant important de connaître que sur un réseau LAN, la qualité est bonne.

A travers ce mémoire, nous avons pu décrire les technologies VoIP et ToIP en partant des notions indispensables à ce propos. Et nous avons vu que la qualité des services offerts n'est pas encore satisfaisante. Le principal facteur de cette qualité insatisfaisante est le temps de latence. C'est pourquoi, nous avons essayé de proposer une solution pour pallier ce problème. Ensuite nous sommes amenés à étudier le problème de routage IP avec gestion de files d'attente. Et pour l'amélioration de routage, nous avons exploité deux types de techniques : la technique de contrôle de flux et la technique de contrôle de congestion. C'est ainsi que notions théoriques des graphes sont prises comme support afin de modéliser les réseaux IP.

Malgré les études faites sur la QoS, la ToIP encourt des menaces comme l'écoute des conversations, obstruction, attaques en déni de service… Des améliorations et des études plus approfondies sur la sécurité devront encore avoir lieu pour compléter ce qui a été traité dans cet ouvrage.

ANNEXE 1 NOTION DE THEORIE DES GRAPHES

Un graphe est constitué d'un ensemble de points appelés *sommets* ou *nœuds* et d'un ensemble de couples ordonnés appelés *arêtes* ou *arcs* si ceux-ci sont orientés.

A1.1. Graphes : concepts orientés

Un graphe $G = [X, U, \varphi]$ est déterminé par la donnée :

- d'un ensemble X dont les éléments sont appelés des *sommets* ou des *nœuds*. Si N = card(x) est le nombre de sommets, on dit que le graphe G est *d'ordre N*. On supposera que les sommets sont numérotés $i = 1,2,..., N$;
- d'un ensemble U dont les éléments $u \in U$ sont des couples ordonnés de sommets appelés des *arcs*. Si u = (i,j) est un arc de *G, i* est *l'extrémité initiale* de *u* et *j l'extrémité terminale* de *u*. On notera *card(u) = M*.
- φ est une fonction d'incidence qui à chaque arc *e* de *U* associe un couple *(x,y)* d'éléments de *X* non nécessairement distincts, *x* est l'origine de *e* et *y* est le but de *e*.

Graphiquement, les sommets peuvent être représentés par des points et l'arc $u = (i,j)$ sera représenté par une flèche joignant les deux points *i* et *j* (*j* étant la pointe de la flèche).

Exemple :

Soit G le graphe orienté défini par $X = \{a, b, c\}$, $E = \{e_1,e_2,e_3,e_4,e_5,e_6\}$

$\varphi(e_1) = \{a, b\}$, $\varphi(e_2) = \{b, c\}$, $\varphi(e_3) = \{a, b\}$, $\varphi(e_4) = \{c, b\}$, $\varphi(e_5) = \{c, c\}$, $\varphi(e_6) = \{a, c\}$

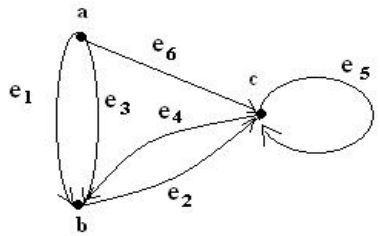

Figure A1.01 : Graphe orienté

Le sommet *a* est l'origine de e_1 et le sommet *b* est le but de e_1.

Les arcs e_1 et e_3 sont parallèles, l'arc e_5 est une boucle mais les arcs e_2 et e_4 ne sont pas parallèles.

Notation :

Un graphe orienté *G* est noté par $G = (V, E)$ dont *V* représente l'ensemble des sommets et *E* l'ensemble des arcs de *G*.

A1.2. Graphes : concepts non orientés

Dans l'étude de certaines propriétés des graphes, il arrive que l'orientation des arcs, c'est-à-dire la distinction entre extrémité initiale et extrémité terminale, ne joue aucun rôle. On s'intéresse simplement à l'existence ou à la non-existence d'un (ou de plusieurs) arc(s) entre deux sommets (sans préciser l'ordre de ces sommets).

A tout arc (i,j) et à tout couple ordonné (i,j), on associe le couple *non ordonné* (i,j), qui est appelé *l'arête* (i,j). On peut dire de façon équivalente qu'une arête est un arc sur lequel on a «oublié» l'orientation.

Graphiquement, l'arête (i,j) sera représentée par un *segment* (sans flèche) joignant les deux points représentatifs des sommets *i* et *j*.

Dans l'étude des propriétés « non orientées » d'un graphe G = [X, U, φ], il suffira donc de considérer l'ensemble U comme un ensemble *d'arêtes* c'est-à-dire une famille finie de parties à deux éléments (distincts ou non) de X.

φ est une fonction d'incidence qui à chaque arête de U associe une paire d'éléments de X non nécessairement distincts.

Exemple :

Soit G défini par X = {a, b, c, d, f, g}, E = {e_1,e_2,e_3,e_4,e_5,e_6,e_7,e_8}

$\varphi(e_1) = \{a, b\},\ \varphi(e_2) = \{b, c\},\ \varphi(e_3) = \{c\},\ \varphi(e_4) = \{c, d\},\ \varphi(e_5) = \{b, d\},\ \varphi(e_6) = \{d, g\},\ \varphi(e_7) = \{b, g\}, \varphi(e_8) = \{b, c\}$

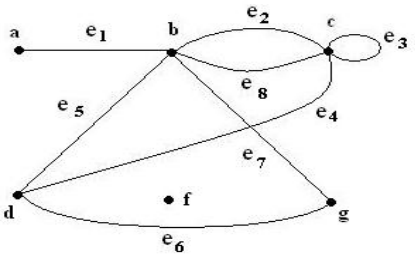

Figure A1.02: Graphe non-orienté

On dit les sommets *a* et *b* sont adjacents, que l'arête e_4 est incidente au sommet *c* et au sommet *d.*

Les arêtes e_2 et e_8 sont parallèles, l'arête e_3 est une boucle, le sommet *f* est « isolé » et le sommet *a* est « pendant ».

Remarques :

- Un graphe non orienté n'est qu'un graphe orienté symétrique ; si un arc relie le sommet *a* au sommet *b,* un autre arc relie le sommet *b* au sommet *a* : on ne trace alors qu'un trait entre *a* et *b* que l'on appelle une « arête ».
- Un *multigraphe* est un graphe pour lequel il peut exister plusieurs arêtes entre deux sommets *i* et *j* donnés.
- Un graphe est dit *simple* s'il est sans boucle et ne comporte jamais plus d'une arête entre deux sommets quelconques.
- Deux arcs (deux arêtes) sont dits *adjacents* s'ils ont au moins une extrémité commune.

A1.3. Degré d'un graphe

Dans le cas d'un graphe orienté, le « degré sortant » d'un sommet *x* est le nombre d'arcs qui partent de *x*, noté $d^+(x)$, et son « degré entrant », noté $d^-(x)$, est le nombre d'arcs arrivant au sommet *x*.

On a la relation :

$$\sum d^+ = \sum d^- = |A| \text{ (nombre d'arcs)} \qquad \text{(A1.01)}$$

Dans le cas d'un graphe non orienté, le degré est le nombre d'arêtes rattachées au sommet x.

On a la relation :

$$\sum d = 2|A| \qquad \text{(A1.02)}$$

A1.4. Chemin ou chaîne

Un « chemin » est une succession d'arcs parcourus dans le même sens. Le nombre d'arcs parcourus s'appelle la longueur du chemin.

On parle de « chaîne » si l'on ne tient pas compte de la direction des arcs c'est-à-dire dans les graphes non orientés.

Si un chemin revient à son point de départ, on parle de « circuit » dans un graphe orienté, ou de « cycle » dans un graphe non orienté.

L'adjectif « élémentaire » s'applique quand le chemin ou le cycle parcourt des sommets distincts.

La « distance » entre deux sommets est la longueur du plus court chemin entre ces deux sommets.

Enfin le « diamètre » d'un graphe est la plus grande distance séparant deux sommets de ce graphe.

A1.5. Connexité

On dit qu'un graphe non orienté est « connexe » si deux éléments quelconques de ce graphe sont reliés par au moins une chaîne.

On appelle « composante connexe » d'un graphe non orienté un sous-ensemble maximal de sommets tels qu'il existe une chaîne entre deux sommets quelconques.

Un graphe orienté est « fortement connexe » si, quels que soient deux sommets *x* et *y*, il existe un chemin reliant *x* à *y* et un reliant *y* à *x*.

A1.6. Types de graphes

A1.6.1. Arbres

Ce sont des graphes non orientés connexes sans cycle ou encore des graphes non orientés connexes dont le nombre de sommets est égal au nombre d'arêtes plus 1.

Remarques :

Une arborescence est un graphe orienté possédant un sommet privilégié, la « racine », tel qu'il existe un et un seul chemin depuis la racine à tout autre sommet.

On peut orienter les arêtes d'un arbre à partir de n'importe quel sommet de façon à obtenir une arborescence.

A1.6.2. Graphe pondéré

Quand les arêtes représentent un coût (en argent, en temps, en distance...), on leur attribue un nombre ce qui donne un graphe pondéré.

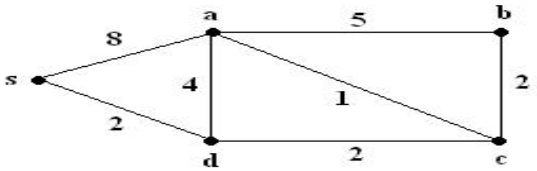

Figure A1.03 : Graphe pondéré

A1.7. Distance dans un graphe non orienté

On définit la distance *d(x, y)* entre deux sommets *x*, *y* d'un graphe non orienté par :

- *d(x,y)* = 0 si x = *y***.**
- *d(x, y)* longueur d'une plus courte chaîne reliant *x* à *y* s'il existe
- *d(x, y)* = ∞ sinon

ANNEXE 2 NOTION SUR LA THEORIE DE FILES D'ATTENTE

A2.1. Techniques de gestion de files d'attente

Les réseaux fonctionnant en mode paquet posent un problème particulier pour la qualité de service car chaque nœud du réseau peut recevoir et analyser chaque paquet de données immédiatement, mais doit attendre, pour pouvoir le transmettre au nœud suivant, la disponibilité de capacité sur la ligne de transmission appropriée. Le délai entre la réception et la réémission d'un paquet est par conséquent variable (cette variation de délai s'appelle la gigue) : le délai peut être très grand lorsque le réseau est congestionné ou quand un paquet particulièrement long est déjà en cours de transmission sur l'interface choisie, ou bien très court si le paquet à transmettre trouve toujours de la capacité disponible immédiatement le long de son chemin de transmission.

Un moyen simple de réduire le délai et la gigue pour un flux de données particulier serait de lui donner priorité sur tous les autres, mais ceci n'est pas une solution acceptable, car les réseaux de données sont conçus en général pour traiter équitablement tous les utilisateurs du réseau. Cette notion d' « équité » peut être interprétée de nombreuses manières différentes, mais l'idée directrice est que les flux de données doivent recevoir le même niveau de service, ou bien si le réseau implémente une notion de priorité, un service d'autant meilleur que la priorité est haute, tous les flux d'un même niveau de priorité devant être de la même manière. Selon l'interprétation exacte de la notion d' « équité » dans le réseau, il est possible d'utiliser plusieurs techniques de gestion de files d'attente dans les nœuds du réseau. Un effort de recherche très important a été fait pour concevoir des politiques d'ordonnancement qui permettent de minimiser les délais de transmission tout en donnant à chaque flux une part équitable de la capacité. Sur chaque file d'attente de sortie correspondant à une ligne de transmission, la politique d'ordonnancement décide de l'ordre de la transmission des paquets ou éventuellement de leur destruction et vise pour chaque flux de données à se rapprocher le plus possible de certains objectifs de vitesse de transmission, de délai, de gigue et de taux de perte de paquets.

Il existe plusieurs technologies, qui peuvent être regroupées en deux catégories.

Dans la première on trouve des algorithmes qui ne visent qu'à ordonner les paquets dans les files de sortie :

- La technique FIFO (*First In, First Out*, c'est-à-dire « Premier Entré, Premier Sorti »), parfois appelée FSFS (*First Come, First Served*, c'est-à-dire « Premier Arrivé, Premier Servi »). Comme son nom l'indique, cette politique se borne à émettre les paquets sur les liens de transmission dans l'ordre dans lequel ils ont été reçus ;

- La technique CBQ (*Class Based Queuing* parfois aussi appelée *Custom Queuing*) qui permet de donner aux paquets un traitement différencié selon leur catégorie ;
- Les techniques de partage équitable dites *Fair Queuing* et *Weighted Fair Queuing.*

Les techniques de la seconde catégorie définissent des politiques de destruction sélective de paquet dans les files d'attente, et peuvent être combinées avec les techniques de la première catégorie :

- Le débordement simple ;
- La technique RED (*Random Early Detection*), WRED (*Weighted Random Early Detection*) et ARED (Adaptive Random Early Detection), que l'on pourrait traduire par respectivement par " Détection Préventive Aléatoire", « Détection Préventive Adaptative Aléatoire »et « Détection Préventive Aléatoire à Coefficient Variables » ;
- La technique BLUE ;
- L'approche IntServ basée sur la réservation statique de ressources (flux par flux) et DiffServ permettant de fournir des garanties par classe et AQM conçu pour la gestion des files d'attente au niveau des routeurs.

A2.2. Différents paramètres à étudier

En général, les arrivées des paquets dans un nœud sont indépendantes entres elles. Ces arrivées sont définies par les processus tels que : Markovien, Poissonnien ou Exponentiel. Ainsi, les paquets placés dans les mémoires tampons du routeur vont être servis avec un processus de service selon l'ordre de leurs arrivées et suivant une discipline de service *DS* appliquée au routeur. Deux cas peuvent se présenter, s'il y a un serveur libre, alors le paquet est traité immédiatement, sinon il doit attendre jusqu'à ce qu'il soit traité, c'est le phénomène d'attente.

Une file d'attente est généralement caractérisée par six composantes :

- le processus d'arrivée,
- le processus de service,
- le nombre de serveurs,
- les disciplines de service,
- la capacité de la file,
- la population des usagers.

A2.3. Types de files d'attente

Il existe plusieurs types de files d'attente telles que : M/M/l, M/M/c, M/M/∞, M/M/c/c, M/M/c/m/K, M/M/c/c/K, M/M/l/K/K et la file M/M/c/K/K. Mais nous n'allons étudier que la file *M/M/1* qui est un standard des files d'attentes.

D'après la notation de Kendall, une file *M/M/1* est donc une file avec un processus de Markov en entrée désigné par le premier *M,* le deuxième *M* signifie que le processus de sortie est aussi un processus de Markov, le nombre 1 indique qu'il n'y a qu'un seul serveur, la discipline de service appliquée est le FIFO « First In, First Out » ou Premier Arrivé, Premier Servi, une capacité infinie et un nombre infini de clients qui peuvent entrer dans cette file.

Réseaux de files d'attente M/M/1

Pour arriver à modéliser des systèmes beaucoup plus complexes, une file d'attente simple n'est pas suffisante, il faut faire appel à des réseaux de files d'attente.

Un réseau de files d'attente est composé d'un ensemble de stations de service (modélisant les ressources, exemple un routeur de transfert de paquets) et d'un ensemble de clients (paquets). Les files d'attente devant chaque station contiennent les activités qui ont besoin d'accéder à la ressource. Ce système est caractérisé par les processus représentant l'arrivée des clients au réseau, les temps de service des clients aux stations, le cheminement des clients d'une station à l'autre, et les disciplines de service des clients à chaque station. Il est possible aussi de définir des limitations de la longueur des files, ce qui entraîne des pertes de clients arrivant à une station déjà pleine, ou le blocage d'une file ne pouvant débiter des clients vers une file saturée.

Les réseaux à transfert de paquets sont modélisés par des réseaux de files d'attente M/M/l : à l'état stationnaire, chaque nœud réagit comme une file d'attente M/M/l isolée. Il convient d'adapter les notations à ces réseaux. La longueur des paquets, en bits, suit une loi exponentielle de paramètre μ. La moyenne est de $1/\mu$ bit. La capacité de la $i^{ème}$ ligne de communication est notée C_i en bit/s. Le produit μC_i est le taux de service en paquets/s. Le taux d'arrivée pour la $i^{ème}$ ligne est λ_i paquets/s.

Pour la $i^{ème}$ ligne, T_i représente le temps d'attente et de transmission, sous la forme :

$$T_i = 1 / (\mu C_i - \lambda_i) \qquad \text{(A2.01)}$$

ANNEXE 3 CODE SOURCE DE LA SIMULATION

```
#include <windows.h>
#include <commctrl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>
#include "resource.h"
#include "dijkstra.h"

INT_PTR CALLBACK DlgProcReseau(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam);
INT_PTR CALLBACK DlgProcRouteurs(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam);
INT_PTR CALLBACK DlgProcParametrage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam);
INT_PTR CALLBACK DlgProcAppel(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam);
INT_PTR CALLBACK DlgProcSimulation(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam);

HINSTANCE hInstance;
int nbLiens, ordre, source, destination, Cout_total;
double tps_latence, tps_traitement,delai_acheminement;
char IP_Appelant[20], IP_Appelee[20], *chemin_suivi;
GRAPHE g;
CHEMIN * chemin;
double traitement_codec(int id_codec);
double capacite(double latence);
int indice_degradation(double latence);
int gigue(double latence);
char * perte(double latence);
char * interpretation(double latence);
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
{
	InitCommonControls();
	DialogBox(hInstance, MAKEINTRESOURCE(IDD_RESEAU), NULL, DlgProcReseau);
	return 0;
}

// Réseau IP
INT_PTR CALLBACK DlgProcReseau(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)
{
	FILE *fp;
	switch(Msg)
	{
	case WM_INITDIALOG:
		fp = fopen("Réseau_IP.txt", "w");
		fclose(fp);
		break;
	case WM_COMMAND:
		switch( LOWORD(wParam))
		{
		case IDB_OK:
			ordre = GetDlgItemInt(hWnd, IDE_NBROUTEURS, NULL, FALSE);
			nbLiens = GetDlgItemInt(hWnd, IDE_NBLIENS, NULL, FALSE);
			fp = fopen("Réseau_IP.txt", "a+");
			fprintf(fp, "Nombre de routeurs : %d\n", ordre);
			fprintf(fp, "Nombre de liens : %d\n", nbLiens);
			fprintf(fp, "\n\n");
			fclose(fp);
			SendMessage(hWnd, WM_CLOSE, 0, 0);
			DialogBox(hInstance, MAKEINTRESOURCE(IDD_ROUTEURS), NULL, DlgProcRouteurs);
			break;
		}
		break;
	case WM_CLOSE:
		EndDialog(hWnd, 0);
		break;
	default:
		return FALSE;
	}
	return TRUE;
}
```

```
// Graphe
INT_PTR CALLBACK DlgProcParametrage(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)
{
	int i;
	FILE *fpBuffer, *fpReseau;
	PROCESS_INFORMATION ProcessInformation;
	STARTUPINFO StartUpInfo;
	switch(Msg)
	{
	case WM_INITDIALOG:
		g.n_sommets = ordre;
		g.sommets_tbl = malloc(g.n_sommets * sizeof(SOMMET));
		g.n_aretes = nbLiens;
		g.aretes_tbl = malloc(g.n_aretes * sizeof(ARETE));
		fpBuffer = fopen("Buffer.txt", "w");
		fprintf(fpBuffer, "%d", 0);
		fclose(fpBuffer);
		return TRUE;
	case WM_COMMAND:
		switch( LOWORD(wParam))
		{
		case IDB_LIST:
			GetStartupInfo(&StartUpInfo);
			CreateProcess(
				0,
				"notepad.exe Réseau_IP.txt",
				NULL,
				NULL,
				FALSE,
				CREATE_NEW_CONSOLE,
				NULL,
				NULL,
				&StartUpInfo,
				&ProcessInformation);
			break;
		case IDB_ENTERPARAM:
			fpBuffer = fopen("Buffer.txt", "r+");
			fscanf(fpBuffer, "%d", &i);
			if(i < g.n_aretes)
			{	g.aretes_tbl[i].s1 = GetDlgItemInt(hWnd, IDE_NDSRC, NULL, FALSE) - 1;
				g.aretes_tbl[i].s2 = GetDlgItemInt(hWnd, IDE_NDDEST, NULL, FALSE) - 1;
				g.aretes_tbl[i].cout = GetDlgItemInt(hWnd, IDE_COUT, NULL, FALSE);
				fpReseau = fopen("Réseau_IP.txt", "a+");
				fprintf(fpReseau, "%d -> %d: %d\n", g.aretes_tbl[i].s1 + 1, g.aretes_tbl[i].s2 + 1,
				g.aretes_tbl[i].cout);
				fclose(fpReseau);
				SetDlgItemText(hWnd, IDE_NDSRC, NULL);
				SetDlgItemText(hWnd, IDE_NDDEST, NULL);
				SetDlgItemText(hWnd, IDE_COUT, NULL);
				i++;
			}
			else
			{	MessageBox(hWnd, "Nombre d'arètes en excès", "Erreur", MB_OK);
				SendMessage(hWnd, WM_COMMAND, IDB_NXT, 0);
			}
			fseek(fpBuffer, 0L, SEEK_SET);
			fprintf(fpBuffer, "%d", i);
			fclose(fpBuffer);
			break;
		case IDB_NXT:
			SendMessage(hWnd, WM_CLOSE, 0, 0);
			DialogBox(hInstance, MAKEINTRESOURCE(IDD_APPEL), NULL, DlgProcAppel);
			break;
		}
		break;
	case WM_CLOSE:
		EndDialog(hWnd, 0);
		break;
	default:
		return FALSE;
	}
	return TRUE;
}
```

```
// Appel
INT_PTR CALLBACK DlgProcAppel(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)
{
	int i, codec, source, destination, id_codec1, id_codec2;
	char buffer[100];
	char *Liste[6] = {"1. G.711 (PCM) 64 kbps",
		"2. G.726 (ADPCM) 32 kbps",
		"3. G.728 (CELP) 16 kbps",
		"4. G.729 (CS-ACELP) 8 kbps",
		"5. G.723.1 (MP-MLQ) 6.4 kbps",
		"6. G.723.1 (MP-MLQ) 5.3 kbps"};
	double codecs[] = {0.125, 0.125, 0.625, 15, 37.5, 37.5};
	switch(Msg)
	{
	case WM_INITDIALOG:
		SendDlgItemMessage(hWnd, IDC_CALLING, CB_ADDSTRING, 0, (LPARAM)" ");
		SendDlgItemMessage(hWnd, IDC_CALLED, CB_ADDSTRING, 0, (LPARAM)" ");
		for(i = 1; i <= ordre; i++)
		{
			sprintf(buffer, "192.168.%d.1", i);
			SendDlgItemMessage(hWnd, IDC_CALLING, CB_ADDSTRING, 0, (LPARAM)buffer);
			SendDlgItemMessage(hWnd, IDC_CALLED, CB_ADDSTRING, 0, (LPARAM)buffer);
		}
		SendDlgItemMessage(hWnd, IDC_CODEC1, CB_ADDSTRING, 0, (LPARAM)" ");
		SendDlgItemMessage(hWnd, IDC_CODEC2, CB_ADDSTRING, 0, (LPARAM)" ");
		for(i = 0; i < 6; i++)
		{
			SendDlgItemMessage(hWnd, IDC_CODEC1, CB_ADDSTRING, 0, (LPARAM)Liste[i]);
			SendDlgItemMessage(hWnd, IDC_CODEC2, CB_ADDSTRING, 0, (LPARAM)Liste[i]);
		}
		break;
	case WM_COMMAND:
		switch( LOWORD(wParam))
		case IDB_PREC:
		{
			SendMessage(hWnd, WM_CLOSE, 0, 0);
			DialogBox(hInstance, MAKEINTRESOURCE(IDD_PARAMETRAGE), NULL, DlgProcParametrage);
			break;
		case IDB_LAUNCH:
			source = (int)SendDlgItemMessage(hWnd, IDC_CALLING, CB_GETCURSEL, 0, 0) - 1;
			destination = (int)SendDlgItemMessage(hWnd, IDC_CALLED, CB_GETCURSEL, 0, 0) - 1;
			codec = (int)SendDlgItemMessage(hWnd, IDC_CODEC1, CB_GETCURSEL, 0, 0);
			chemin_suivi = malloc(100 * sizeof(char));
			strcpy(chemin_suivi, "");
			id_codec1 = (int)SendDlgItemMessage(hWnd, IDC_CODEC1, CB_GETCURSEL, 0, 0) - 1;
			id_codec2 = (int)SendDlgItemMessage(hWnd, IDC_CODEC2, CB_GETCURSEL, 0, 0) - 1;
			chemin = dijkstra(&g, source, destination);
			Cout_total = cout_total(&g, source, destination);
			delai_acheminement = Cout_total * 32;
			tps_traitement = traitement_codec(id_codec1) + traitement_codec(id_codec2);
			tps_latence = delai_acheminement + tps_traitement;
			for(i = chemin->n_sommets - 1; i > 0; i--)
			{
				sprintf(buffer, "%d ", chemin->sommets_tbl[i] + 1);
				strcat(chemin_suivi, buffer);
			}
			sprintf(buffer, "%d ", chemin->sommets_tbl[i] + 1);
			strcat(chemin_suivi, buffer);
			GetDlgItemText(hWnd, IDC_CALLING, IP_Appelant, 20);
			GetDlgItemText(hWnd, IDC_CALLED, IP_Appelee, 20);
			SendMessage(hWnd, WM_CLOSE, 0, 0);
			DialogBox(hInstance, MAKEINTRESOURCE(IDD_SIMULATION), NULL, DlgProcSimulation);
			break;
		}
		break;
	case WM_CLOSE:
		EndDialog(hWnd, 0);
		break;
	default:
		return FALSE;
	}
	return TRUE;
}
```

BIBLIOGRAPHIE

[1] L.E. Randriarijaona, *Réseaux TCP/IP*, Cours I4-TCO, Dép. Tél.- E.S.P.A, A.U. : 2006 - 2007

[2] http : //www.laissus.fr/cours/cours.html

[3] http : //www.frameip.com/tcpip/index.html

[4] G. Pujolle, *Les Réseaux*, Edition Eyrolles : Paris 2003

[5] http : //www.guill.net/reseaux/tcpip/index.html

[6] C. Servin, *Réseaux et Télécoms*, Dunod : Paris 2003

[7] http: //www.commentcamarche.net/internet/tcpip.php3

[8] J. Doyle, *Routing TCP/IP – Volume I*, CCIE 1919, Cisco Press, 2001

[9] http : //www.frameip.com/ipv6/index.html

[10] http://www.voipfr.com

[11] http : //www.frameip.com/tcp/index.html

[12] http : //www.frameip.com/udp/index.html

[13] Z.V. Andriamiasy, *Téléphonie avancée*, Cours I5-TCO, Dép. Tél.- E.S.P.A., A.U. : 2007-2008.

[14] T. H. Andriamialison, *La Voix sur IP*, Mémoire de fin d'études, Dép.Tél. - ESPA, A.U. : 2000-2001

[15] http : //www.frameip.com/voip/index.html

[16] http : //www.frameip.com/toip/index.html

[17] http: //www.telephone-par-internet.com

[18] O. Hersent, D. Gurle, J.P. Petit, *La Voix sur IP*, Dunod : Paris 2006

[19] A. Sulkin, *PBX Systems for IP Telephony*, McGraw-Hill Telecom : USA 2002

[20] ITU-T, Recommendation Y.1291, *An architectural framework for support of Quality of Service (QoS) in Packet networks*

[21] Cabinet Arcome, *Etude technique, économique et réglementaire de l'évolution vers les réseaux de nouvelle génération NGN*, Septembre 2002

[22] http : //www.awt.be/fiche/toip.pdf

[23] http: //www.httr.ups.tlse.fr/pedagogie/cours/commutation/commutation.htm

[24] http: //www.httr.ups.tlse.fr/pedagogie/cours/commutation/routage.htm

[25] N. A. H. Rakotomalala, *La visioconférence sur Internet*, Mémoire de fin d'études, Dép.Tél. - ESPA, A.U. : 2001-2002

[26] D. Kofman, *Synthèse sur l'évolution des réseaux de télécommunications*, ENST, Paris : 2001

[27] A. Tanenbaum, *Réseaux*, 4e édition, Pearson Education, Paris : 2003

[28] M.A. Rakotomalala, *Réseaux des télécommunications*, Cours I5-TCO, Dép. Tél.- E.S.P.A., A.U. : 2007-2008

[29] M. Gondran, M. Minoux, *Graphes et algorithme*, Dunod informatique, 1978

[30] J. Cohen, *Théorie des graphes et algorithmes*, UIT Bordeaux I, 2002

[31] http : //www.nimbustier.net/publications/djikstra/djikstra.html

[32] http : //www.nimbustier.net/publications/djikstra/graphes.html

[33] J.N. Rasamoelina, *Trafic et file d'attente*, Cours I5-TCO, Dép. Tél.- E.S.P.A., A.U. : 2007-2008

[34] E. Gelenbe, G. Pujolle, *Introduction aux réseaux de files d'attente*, Edition Eyrolles, Paris : 1990

[35] http : //cpp.developpez.com

PAGE DE RENSEIGNEMENTS

Nom : ANDRIANARISON

Prénoms : Maherizo Valinaina

Adresse de l'auteur : Lot 24 A 98 Est Tribunal
ANTSIRABE 110
Madagascar
Tél: 033 12 826 80 / 032 55 768 54 / 034 08 700 14
E-mail: andrianarison_maherizo@yahoo.fr

Titre du mémoire : « LA TELEPHONIE SUR IP : OPTIMISATION DU TEMPS DE LATENCE PAR AMELIORATION AU NIVEAU DU ROUTAGE IP »

Nombre de pages : 110

Nombre de figures : 61

Nombre de tableaux : 15

Mots clés : Dijkstra - files d'attente - graphes - latence - optimisation - QoS - routage - TCP/IP – ToIP - VoIP

Directeur de mémoire : M. ANDRIAMIASY Zidora

RESUME

La téléphonie est la plus pénétrante de toutes les technologies. Pourtant, le tarif plus élevé de la téléphonie traditionnelle limite encore son utilisation par le grand public. Ainsi, la réduction de coût a accéléré la convergence de réseaux de voix et de données. Cependant, la téléphonie sur Internet est encore loin de satisfaire aux exigences de qualité de service attendues pour ce type de service, même si de fortes améliorations sont prévisibles.

Le but de cet ouvrage est de fournir une synthèse sur les aspects théoriques des protocoles et des techniques mises en jeu par la téléphonie par paquet. Cet ouvrage propose une méthode pour l'optimisation du temps de latence en téléphonie sur IP par amélioration au niveau du routage des paquets.

ABSTRACT

Telephony is the most pervasive of all technologies. However, the higher tariff of traditional telephony still limits its use by the general public. Thus, cost reduction has fueled the convergence of data and voice networks. However, telephony over Internet is still far from satisfying the requirements for quality of service awaited for this type of service, even if strong improvements are foreseeable.

The target of this work is to provide a synthesis on the theoretical aspects of the protocols and the techniques concerned by telephony over packet. This work proposes a method for optimization of the latency time in telephony over IP by improvement on the level of packet routing.