République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

En vue de l'obtention du diplôme Master en Informatique

Option :

Réseaux et Systèmes Distribués

Mémoire de fin de cycle

Téléphonie Internet en mode P2P-SIP

Réalisé par :

M^{lle} MADI Nouria
M^{lle} BOUICHE Razika

Soutenu devant le jury composé de :

Président :	M^{r} MOUMEN Hamouma	M.A. Université de Bejaia
Examinateur :	M^{r} RABAH Hamid	M.A. Université de Bouira
Examinatrice :	M^{me} HALFOUNE Nadia	M.A. Université de Béjaia
Encadreur :	M^{r} AMAD Mourad	M.C. Université de Béjaia

Juin 2014

En tout premier lieu, nous remercions Allah le tout puissant, à la sagesse et au savoir infinis, " Gloire à Toi , Nous n'avons de savoir que ce que Tu nous as appris. Certes c'est Toi l'Omniscient, le Sage, le tout miséricordieux le très miséricordieux " (*Sourate al-Baqarah, verset 32*).

Nous tenons à exprimer nos sincères remerciements à notre promoteur : M^r AMAD Mourad qui nous a aidé et suivi notre travail pendant toute cette année.

Nous remercions aussi M^r MOUMEN Hamouma de nous faire l'honneur de présider le jury de notre soutenance. Nos remerciements s'adressent aussi aux membres de jury constitué de : M^r RABAH Hamid et M^{me} HALFOUNE Nadia pour avoir accepté la tache de juger ce travail.

Nous remercions vivement notre famille, en particulier nos parents, pour nous avoir toujours soutenu au cours de nos études. Qu'ils trouvent ici le fruit de leur patience et du soutien permanent qu'ils nous ont prodigué pour affronter tous les moments difficiles.

Notre sincère gratitude va vers tous les personnes ayant contribué à la réalisation de notre mémoire. Merci à tous ceux qui nous ont apporté leur aide, chacun à sa manière.

Un grand merci à tout(e)s nos ami(e)s sans exception.

Dédicaces Razika

Je dédie ce travail à :

À mes trés chers parents.

À mes frères .

À mes sœurs.

À tous mes amis (es).

À toute ma famille cousins et cousines.

À ma binôme Nouria et toute sa famille.

À toutes les personnes que j'ai connues, tous ceux que j'aime, tous ceux que m'aiment et à tout ceux qui m'ont aidé.

BOUICHE Razika

Dédicaces Nouria

Je dédie ce travail à :

À ma source de bonheur et la joie de mon cœur, mes chers parents. Leurs compréhensions, encouragements et conseils sont toujours présents.

À mes quatre frères Fatah, Djamel, Mourad, Rabah ansi que toutes leurs familles.

À mes deux sœurs Houria et Kahina.

À tous mes cousins, cousines, tantes et oncles.

À la mémoire de mes grands parents.

À ma binôme Razika et toute sa famille.

À tous mes amis Razika, Hizia, Khokha, Sabah, Hamanou, Hakim...

À tous ceux que j'aime et à tous ceux que m'aiment.

MADI Nouria

Résumé

LA téléphonie IP (*ToIP*) en mode P2P basée sur le protocole SIP, apporte beaucoup d'avantages grâce à sa fiabilité et sa facilité.

Le routage de données dans les réseaux Peer to Peer (*P2P*) est reconnu comme un domaine de recherche très actif, vu les spécificités de ce type de réseau en utilisant le protocole de signalisation SIP (*session iniation protocol*).

Notre contribution consiste en un routage pioritaire pour les appels téléphoniques d'urgence, qui garantit la transmission des requêtes dans un réseau décentralisé non structuré, où les noeuds sont allégés de toute décision concernant le routage de ces requêtes.

La simulation a montré que les résultats fournis par notre routage sont très compétitifs, et ceux en minimisant un seul paramètre de la qualité de service (l*e temps*).

Mots clés **: P2P, SIP, P2P-SIP, Routage prioritaire.**

Abstract

THE telephony over IP (*TOIP*) in mode Peer to Peer (*P2P*) based on SIP (session iniation protocol), brings many advantages thanks to its reliability and its easiness.

Data routing in P2P network is a very active area of research, given the specifities of this type of network in using the signaling protocol SIP.

Our contribution is a priority routing for emergency calls, witch guarantees the delivry of requests in a decentralized and unstructured network, where the nodes are relieved of any decision concerning the routing of these requests.

The simulation shows that the results provided by our routing are very competitive, and this by minimized one measure of the quality of service (*the time*).

Keywords **: P2P, SIP, P2P-SIP, Priority routing.**

Table des matières

Table des matieres iv

Liste des figures vi

Liste des Acronymes viii

Introduction générale 1

1 Réseaux Peer to Peer (*P2P*) 3
- 1.1 Introduction . . . 3
- 1.2 Classification des systèmes informatiques . . . 3
- 1.3 Historique de Peer to Peer . . . 4
- 1.4 Définition des réseaux Peer to Peer . . . 5
- 1.5 Caractéristiques . . . 5
  - 1.5.1 Décentralisation . . . 5
  - 1.5.2 Passage à l'échelle . . . 6
  - 1.5.3 Auto-organisation . . . 6
  - 1.5.4 Autonomie des noeuds . . . 6
  - 1.5.5 Hétérogénéité . . . 6
  - 1.5.6 Dynamique . . . 7
  - 1.5.7 Tolérance aux fautes . . . 7
  - 1.5.8 Anonymat . . . 7
  - 1.5.9 Scalabilité . . . 7
  - 1.5.10 Contrôle . . . 7
- 1.6 Applications des réseaux Peer to Peer . . . 8
  - 1.6.1 Partage de fichiers . . . 8
  - 1.6.2 Collaboration . . . 8
  - 1.6.3 Calcul distribué . . . 8
  - 1.6.4 Communication . . . 8
- 1.7 Architectures des réseaux Peer to Peer . . . 9
  - 1.7.1 Architecture centralisée . . . 9
  - 1.7.2 Architecture décentralisée . . . 12
  - 1.7.3 Architecture hybride . . . 14
- 1.8 Autre classification des réseaux Peer to Peer . . . 16
  - 1.8.1 Réseaux Peer to Peer structurés . . . 16
  - 1.8.2 Réseaux Peer to Peer non structurés . . . 16
- 1.9 Problèmes des réseaux Peer to Peer . . . 17

1.9.1 Sécurité . . . 17
1.9.2 Interopérabilité . . . 18
1.9.3 Bande passante . . . 18
1.9.4 Découverte des ressources . . . 18
1.10 Conclusion . . . 18

2 Protocole SIP (Session Initiation Protocol) 19
2.1 Introduction . . . 19
2.2 Définition de SIP . . . 19
2.3 Caractéristiques de SIP . . . 20
2.3.1 Simplicité . . . 20
2.3.2 Description générique de session . . . 21
2.3.3 Système d'adressage . . . 21
2.3.4 Multimédia . . . 21
2.4 Fonctionnalités de SIP . . . 21
2.5 Messages SIP . . . 22
2.5.1 Requêtes (*méthodes*) SIP . . . 23
2.5.2 Réponses SIP . . . 24
2.6 Architecture de SIP . . . 24
2.7 Adressage SIP . . . 26
2.8 Exemple d'établissement d'une session . . . 26
2.9 Problèmes de SIP . . . 27
2.9.1 Pares-feu (*Firewalls*) . . . 27
2.9.2 NATs (*Network Address Translation*) . . . 28
2.9.3 Complexité . . . 28
2.10 Conclusion . . . 28

3 État de l'art sur la téléphonie Internet en mode P2P-SIP 29
3.1 Introduction . . . 29
3.2 Généralités sur la téléphonie IP (*ToIP*) . . . 29
3.2.1 Téléphonie sur IP et le RTC . . . 30
3.2.2 Modéle de la ToIP . . . 31
3.2.2.1 Modèle de PC à PC . . . 31
3.2.2.2 Modèle de PC à téléphone . . . 31
3.2.2.3 Modèle de téléphone à téléphone . . . 32
3.2.3 Composants de la ToIP . . . 32
3.2.4 Avantages de la ToIP . . . 33
3.2.4.1 Convergence . . . 33
3.2.4.2 Optimisation des ressources . . . 33
3.2.4.3 Coût de transport quasiment nul . . . 33
3.2.4.4 Services spéciaux . . . 33
3.2.5 Problèmes de la ToIP . . . 34
3.2.5.1 Sécurité . . . 34
3.2.5.2 Disponibilité . . . 34
3.2.5.3 Gestion . . . 34
3.2.5.4 Contrôle . . . 34
3.2.5.5 Qualité de service . . . 34

3.3 Téléphonie IP basée sur le protocole SIP . . . 35
3.4 Téléphonie IP basée sur le P2P-SIP . . . 35
3.4.1 Fonctions de P2P-SIP . . . 36
3.4.1.1 Contrôle d'accès . . . 36
3.4.1.2 Bootstrap . . . 37
3.4.1.3 Routage . . . 37
3.4.1.4 Stockage . . . 37
3.4.1.5 Communication . . . 37
3.4.2 Différentes architectures P2P-SIP . . . 37
3.4.2.1 Architecture de Henning Schulzrinne . . . 38
3.4.2.2 Sosimple . . . 43
3.4.2.3 Skype . . . 46
3.4.2.4 Comparaison entre les architectures de P2P-SIP . . . 49
3.4.3 Problèmes de P2P-SIP . . . 50
3.4.3.1 Tolérance aux pannes . . . 50
3.4.3.2 Surcharge des supers noeuds . . . 50
3.4.3.3 Sécurité . . . 51
3.5 Conclusion . . . 51

4 Routage prioritaire en mode P2P-SIP **52**
4.1 Introduction . . . 52
4.2 Problématique . . . 52
4.3 Proposition d'une solution . . . 53
4.3.1 Principe de fonctionnement . . . 53
4.3.2 Caractéristique de la file d'attente . . . 54
4.3.3 Processus d'arrivé et servir des requêtes . . . 56
4.3.4 Routage prioritaire de requêtes entre les noeuds . . . 58
4.4 Conclusion . . . 62

5 Évaluation de performances **63**
5.1 Introduction . . . 63
5.2 Techniques d'évaluation . . . 63
5.2.1 Analytique . . . 63
5.2.2 Mesure . . . 64
5.2.3 Simulation . . . 64
5.3 Choix du MATLAB . . . 65
5.4 Paramèttres de simulation . . . 65
5.5 Étapes de réalisation du simulateur . . . 66
5.5.1 Initialisation des variables du simulation . . . 66
5.5.2 Déploiement des noeuds du réseau . . . 67
5.5.3 Création de l'échéancier . . . 67
5.5.4 Application de l'algorithme de routage . . . 68
5.5.5 Affichage des résultats . . . 68
5.6 Métriques de performance . . . 68
5.7 Résultats de la simulation . . . 69
5.8 Conclusion . . . 70

Conclusion générale et Perspectives **71**

Bibliographie **72**

Liste des figures

1.1 Classification des systèmes informatiques 4
1.2 Architecture P2P centralisée. 9
1.3 Exemple de fonctionnement de Napster. 11
1.4 Architecture P2P décentralisée 12
1.5 Exemple de fonctionnement de Gnutella. 14
1.6 Architecture P2P hybride. 15

2.1 Positionnement du protocole SIP parmi les protocoles Internet. 20
2.2 Modes de communication SIP. 22
2.3 Format des messages SIP. 23
2.4 Architecture de SIP. 25
2.5 Adressage SIP. 26
2.6 Exemple d'établissement d'une session avec le protocole SIP. 27

3.1 Téléphonie sur IP et le RTC. 30
3.2 Modèle de PC à PC. 31
3.3 Modèle de PC à téléphone. 32
3.4 Modèle de téléphone à téléphone. 32
3.5 Téléphonie IP basée sur le protocole SIP. 35
3.6 Architecture de P2P-SIP. 36
3.7 Fonctions de P2P-SIP. 36
3.8 Architecture réseau de Henning Schulzrinne. 38
3.9 Enregistrement d'un utilisateur. 39
3.10 Diagramme d'un noeud P2P-SIP. 40
3.11 Diagramme d'enregistrement d'un noeud. 41
3.12 Recherche d'un utilisateur. 42
3.13 Défaillance d'un super noeud dans la DHT. 43
3.14 Arrivé d'un noeud. 44
3.15 Localisation d'un nouveau noeud. 46
3.16 Architecture de Skype. 47
3.17 Arrivé d'un nouvel utilisateur. 48
3.18 Localisation d'un utilisateur. 49

4.1 Modélisation de réseau. 54
4.2 File d'attente M/M/1. 55
4.3 Exemple d'implémentation. 60

5.1 Principales fonctions du simulateur. 66

5.2 Déploiement du réseau. 67
5.3 Temps de réponse en fonction de nombre de requêtes envoyées. 69

Liste des tableaux

1.1 Comparison entre les architectures P2P. 17

3.1 Comparaison entre architectures de P2P-SIP. 50

5.1 Paramèttres de simulation. 65
5.2 Exemple d'échéancier d'un noeud. 68
5.3 Comparaison entre les temps de réponse des deux routages. 70

Liste des Acronymes

ACK	**ACQ**uittement .
CPU	**C**entral **P**rocessing **U**nit.
CRLF	**C**arriage **R**eturn **L**ine **F**eed.
CS	**C**lient **S**kype.
DHCP	**D**ynamic **H**ost **C**onfiguration **P**rotocol.
DNS	**D**omain **N**ame **S**erver.
DHT	**D**istributed **H**ash **T**able.
E-mail	**E**lectronic-mail.
FIFO	**F**irst **I**n **F**irst **O**ut.
HTTP	**H**yper **T**ext **T**ransport **P**rotocol.
ID	**ID**entificateur.
IETF	**I**nternet **E**ngineering **T**ask **F**orce.
IM	**I**nstant **M**essaging.
IP	**I**nternet **P**rotocol.
MCU	**M**ultipoint **C**ontrol **U**nit.
MMUSIC	**M**ultiparty **MU**ltimedia **S**ess**I**on **C**ontrol.
NAT	**N**etwork **A**dress **T**ranslation.
OSI	**O**pen **S**ystem **I**nterconnection.
PC	**P**ersonnal **C**omputer.
PDA	**P**ersonnal **D**igital **A**ssistant.
PR	**PR**iority.
P2P	**P**eer **T**o **P**eer.

QoS **Q**uality of **S**ervice .

RAM **R**andom **A**ccess **M**emory.
RFC **R**e **F**eren **C**e.
RTC **R**éseau **T**éléphonique **C**ommuté.
RTCP **R**éseau **T**éléphonique **C**ommuté **P**ublic .
RTP **R**eal-time **T**ransport **P**rotocol.

SCTP **S**tream **C**ontrol **T**transmission**P**rotocol.
SDP **SD**escription **P**rotocol.
SIP **S**ession **I**nitiation **P**rotocol.
SN **S**uper **N**oeud.
SMTP **S**imple **M**ail **T**ransport **P**rotocol.

TCP **T**ransmission **C**ontrol **P**rotocol.
TOIP **T**elephony **O**ver **I**nternet **P**rotocol.
TTL **T**ime **T**o **L**ive.

UDP **U**ser **D**atagram **P**rotocol.
VOIP **V**oice **O**ver **I**nternet **P**rotocol.

Introduction générale

L'ARRIVÉ de nouvelles technologies informatiques de communication prend son origine de nouvelles perspectives de simplification d'architecture et d'administration des équipements.

La téléphonie sur IP (*ToIP*) est une révolution technologique qui permet des communications en temps réel entre deux ou plusieurs pairs, elle constitue une avancée significative dans la convergence entre les réseaux de la téléphonie et ceux de données. Le fort succès que représente l'utilisation des emails ou du chat instantané entre pairs, rend l'exploitation du réseau avec la ToIP développée à grand échelle.

Le réseau Peer to Peer (*P2P*) est un réseau dynamique qui n'a pas du serveur central, dont tous les pairs jouent le même rôle. Les applications P2P existantes permettent de différencier trois grandes classes : Le calcul distribué, le travail partagé et l'échange de fichiers qui est avec son succès le plus populaire pour les applications de la VoIP.

La signalisation en utilisant le protocole SIP (*session initiation protocol*) est l'une des plus importantes fonctions dans les télécommunications, elle permet aux utilisateurs du réseau de communiquer entre eux pour établir et terminer des sessions multimédias telle que la VoIP.

P2P-SIP s'appuie sur la technologie Peer to Peer dans le but d'offrir un service SIP et distribué aux utilisateurs. Le travail avec le modèle client/serveur classique, utilisé par SIP délègue l'ensemble des fonctionnalités du protocole SIP aux pairs. L'utilisation de l'ensemble des ressources disponibles sur le réseau P2P plutôt qu'un serveur unique diminue le taux de panne du système.

Une des principales difficultés soulevées par les applications de la ToIP est de prendre en considération les paramètres de la qualité de service (*QoS*) dans une architecture P2P décentralisée non structurée (*Gnutella*).

Dans le cadre de cette problématique, nous avons proposé une solution qui prend en compte un seul paramètre de QoS (*le temps*). Cette solution basée sur la priorité de transmission des appels téléphoniques d'urgence.

Notre mémoire a été réalisé pour bien comprendre le fonctionnement de la ToIP en mode P2P-SIP. Il est organisé en cinq chapitres comme suit :

Le premier chapitre décrit les réseaux pair à pair, leurs caractéristiques, leurs applications, leurs architectures et leurs principaux problèmes.

Le deuxième chapitre décrit le protocole SIP, ses caractéristiques, ses différents messages, son adressage et un exemple d'établissement d'une session multimédia.

Le troisième chapitre décrit l'état de l'art de la téléphonie Internet en combinant les réseaux P2P et le protocole de signalisation SIP (*P2P-SIP*).

Le quatrième chapitre présente la solution proposée qui est le routage prioritaire en mode P2P-SIP.

Le cinquième chapitre présente l'évaluation de performances de la solution proposée.

Notre mémoire s'achève par une conclusion générale résumant les grands points abordés dans ce mémoire, ainsi que des perspectives pour des travaux futurs.

Chapitre 1

Réseaux Peer to Peer (*P2P*)

1.1 Introduction

Avec l'arrivé de l'Internet et son rôle actif croissant dans l'économie et la société, l'informatique réseau n'a eu de cesse de trouver des innovations pour exploiter les ressources qu'un réseau de cette ampleur contient d'où l'apparition de réseau Peer to Peer.

Les réseaux pair à pair ou (*P2P*) est un modèle alternatif de réseau fourni par l'architecture traditionnelle de client/serveur, ils emploient un modèle décentralisé dans lequel chaque machine, désignée sous le nom d'un pair joue le rôle d'un client et d'un serveur en même temps. C'est-à-dire, le pair peut lancer des demandes à d'autres pairs, et répond en même temps aux demandes entrantes d'autres pairs sur le réseau. Il diffère du modèle client/serveur où un client peut seulement envoyer des demandes à un serveur et alors attendre la réponse de ce dernier.

1.2 Classification des systèmes informatiques

Les systèmes informatiques peuvent être classés en deux grandes catégories présentées par la figure1.1 : Les systèmes centralisés et les systèmes distribués. Ces derniers peuvent être classés en deux modèles : Le modèle client/ serveur (*hiérarchique ou plat*) et le modèle Peer to Peer (*centralisé, décentralisé ou hybride*).

Dans la suite de chapitre, nous allons établir le modèle Peer to Peer en étudiant ses caractéristiques, ses architectures, ses applications et ses principaux problèmes.

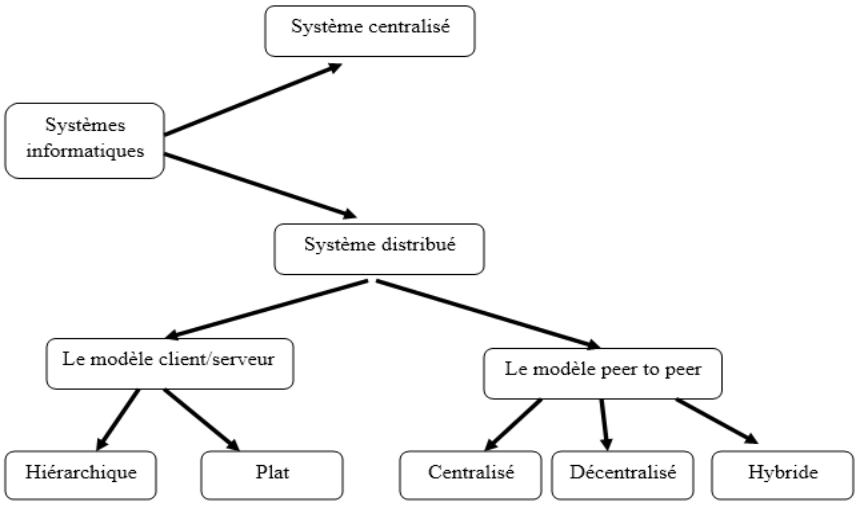


FIGURE 1.1 – Classification des systèmes informatiques

1.3 Historique de Peer to Peer

En 1999, Shawn Fanning, un étudiant de la Northeastern University de Boston agé de 19 ans, épaulé par Jordan Ritter et Sean Parker **[8]**, cherche à développer une méthode pour permettre l'échange de fichiers musicaux sur le réseau Internet sans passer par les moteurs de recherche, les sites web ou les forums de discussion en ligne. Shawn Fanning crée sa propre société sous le nom Napster avec l'aide de son oncle. La première version de Napster est mise à disposition au public en juin 1999 **[8]**.

Napster est basé sur un système d'échange direct de fichiers au format MP3 entre pairs grâce à un serveur central. L'application Napster va également faire des émules et contribuer aux échanges de fichiers non musicaux, qu'ils s'agissent de fichiers vidéos ou d'applications informatiques classiques. Seul le transfert de fichiers était décentralisé **[9]**.

La disparition de Napster donne néanmoins lieu durant l'année 2001 à l'apparition d'applications P2P nouvelles : Gnutella et KazaA. Ces nouveaux systèmes d'échange connaissent un succès immédiat par rapport à Napster, puisqu'à la différence de Napster, aucun serveur central n'est utilisé pour l'établissement des connexions entre utilisateurs. Seuls quelques serveurs annuaires sont utilisés pour initialiser les applications avec une première liste d'ordinateurs pairs **[9]**.

1.4 Définition des réseaux Peer to Peer

Le réseau Peer to Peer (*P2P*) est un réseau logique qui utilise un réseau physique **[10]**. Le terme Peer to Peer désigne une classe de systèmes et d'applications qui utilisent des ressources distribuées, où les entités communiquent entre elles sans utiliser un serveur central.

Les systèmes pair à pair permettant à plusieurs ordinateurs de communiquer via un réseau, de partager simplement des objets, des fichiers le plus souvent, mais également des flux multimédias continus (*streaming*), le calcul réparti, la téléphonie (*comme le skype*) sur Internet.

Un réseau Peer to Peer est dynamique, au sens que les entités le constituant peuvent aller et revenir et que sa topologie n'est pas stable.

La technologie Peer to Peer ne permet pas seulement le partage des ressources numériques (*texte, son, image*), mais elle permet également le partage des capacités de traitement de l'information et de l'espace de stockage (*CPU, RAM*), ainsi l'une des spécificités les plus marquantes des réseaux Peer to Peer est que l'échange ont lieu directement entre les utilisateurs qui peuvent être contributeurs et consommateurs au même temps.

1.5 Caractéristiques

Les systèmes pairs à pair sont caractérisés par **[1]** :

1.5.1 Décentralisation

Le fait que chaque noeud gère ses propres ressources, cela permet d'éviter la centralisation de contrôle. Un système P2P peut fonctionner sans avoir aucun besoin d'une administration centralisée, ce qui permet d'éviter les goulets d'étranglements et d'augmenter la résistance du système aux pannes et aux défaillances.

Cette décentralisation présente l'avantage de réduire, même de supprimer complètement tout risque de goulot d'étranglement, très fréquents dans les systèmes mettant en oeuvre le modèle client/serveur. Cependant une architecture complètement décentralisée peut être difficile de la mettre en place puisqu'il n'existe pas d'entité possédant une vue globale de tous les pairs du système, ni même de type de ressource qu'ils partagent.

1.5.2 Passage à l'échelle

Il s'agit de faire coopérer un grand nombre de noeuds pour partager leurs ressources, tout en maintenant une bonne performance du système. Cela signifie qu'un système P2P doit offrir des méthodes bien adaptées avec un environnement dans laquelle il y'a un grand volume de données à partager, un nombre important de messages à échanger entre un grand nombre de noeuds partageant leurs ressources via un réseau largement distribué.

1.5.3 Auto-organisation

Dans les réseaux pair a pair, l'environnement de ressources est complètement dynamique puisque les pairs peuvent apparaitre et disparaitre à n'importe quel moment, la connexion ou la déconnexion d'un noeud ne concerne que le serveur qui devra mettre à jour la liste des clients connectés.

Puisque les systèmes P2P sont souvent déployés sur l'Internet, la participation d'un nouveau noeud à un système P2P ne nécessite pas une infrastructure coûteuse. Il suffit d'avoir un point d'accès à l'Internet et de connaitre un autre noeud déjà connecté pour se rejoindre le système.

1.5.4 Autonomie des noeuds

Chaque noeud gère ses ressources d'une façon autonome, il décide quelle partie de ses données à partager. Il peut se connecter ou/et se déconnecter à n'importe quel moment. Il possède également l'autonomie de gérer sa puissance de calcul et sa capacité de stockage.

1.5.5 Hétérogénéité

À cause de l'autonomie de noeuds possédant des architectures matérielles et/ou logicielles hétérogènes, les systèmes P2P doivent posséder des techniques convenables pour résoudre les problèmes liés à l'hétérogénéité de ressources.

1.5.6 Dynamique

À cause de l'autonomie de noeuds, chaqu'un peut quitter le système à n'importe quel moment ce qui fait disparaitre ses ressources du système. Des nouvelles ressources peuvent être ajoutées au système lors de la connexion de nouveaux noeuds. Alors, à cause de l'instabilité des noeuds, les systèmes P2P doivent être capables de gérer un grand nombre de ressources fortement variables.

1.5.7 Tolérance aux fautes

C'est la capacité des systèmes pour continuer à fournir des services d'une manière régulière malgré la présence des fautes dans le software ou le hardware, ainsi que les arrivés et les départs fréquents des noeuds **[2]**.

1.5.8 Anonymat

Elle est définie comme étant le degré pour lequel les systèmes P2P tiennent compte des opérations non identifiées **[2]**.

1.5.9 Scalabilité

Dans les réseaux Peer to Peer, la qualité et la quantité des données disponibles augmentent au fûr et à mesure que le nombre d'utilisateurs augmente. La valeur du réseau augmente donc avec sa popularité **[3]**.

1.5.10 Contrôle

Les systèmes centralisés permettent plus facilement un contrôle sur les noeuds qui peuvent accéder aux informations fournies par le serveur. Il est en effet relativement aisé de contrôler le point unique d'accès à l'information, qui est la plupart du temps clairement identifiable.

1.6 Applications des réseaux Peer to Peer

Une application est dite pair à pair si elle met en relation des programmes de même nature sans intermédiaire. La grande caractéristique des applications P2P est que chaque participant joue à la fois le rôle de client et de serveur.

Les réseaux pair à pair sont utilisés dans différents domaines, le plus connu est le partage de fichiers à travers Internet. Cependant, diverses applications se sont développées à partir de ce modèle. On retrouve donc :

1.6.1 Partage de fichiers

Le partage de fichiers constitue l'application la plus répondue actuellement dans les réseaux pair à pair. Les internautes peuvent partager leurs fichiers ainsi télécharger les fichiers des autres via des logiciels comme Napster, Gnutella et KaZaA.

1.6.2 Collaboration

Elle concerne le travail collaboratif en temps réel pour la réalisation d'un ensemble de projets distribués, il permet aussi le partage des documents d'un projet, de conduire des réunions, assigner des tâches et de faciliter le travail en ligne **[1]**.

1.6.3 Calcul distribué

Consiste à utiliser les machines connectées à l'Internet pour faire des petites portions d'un grand calcul, en exploitant les ressources (*CPU, mémoire ..*) inutiles des PCs pour accroitre le potentiel réseau **[4]**.

1.6.4 Communication

Les réseaux Peer to Peer sont très adaptés pour toutes les applications nécessitant une communication entre pairs. Cette communication peut être audio (*Skype*) ou par simple message (*texte*).

1.7 Architectures des réseaux Peer to Peer

Il est possible de regrouper les architectures de Peer to Peer en trois grandes classes. Nous allons citer : Architecture centralisée, décentralisée et hybride.

1.7.1 Architecture centralisée

Dans cette architecture, il existe un unique serveur central, gérant l'identité des utilisateurs, l'indexation des partages, les recherches et la mise en relation des pairs. Le serveur donne à chaque client l'adresse des pairs possédant le fichier recherché, le client se connecte alors à ce pair pour échanger directement les données, les fichiers ne passent donc pas par le serveur.

Dans toute architecture centralisée, un dispositif exclusivement serveur se charge de mettre en relation directe tous les utilisateurs connectés. L'intérêt de cette technique réside dans l'indexation centralisée de tous les répertoires et intitulés de fichiers partagés par les pairs sur le réseau. En général, la mise à jour de cette base s'effectue en temps réel, dès qu'un nouvel utilisateur se connecte ou quitte le service **[3]**.

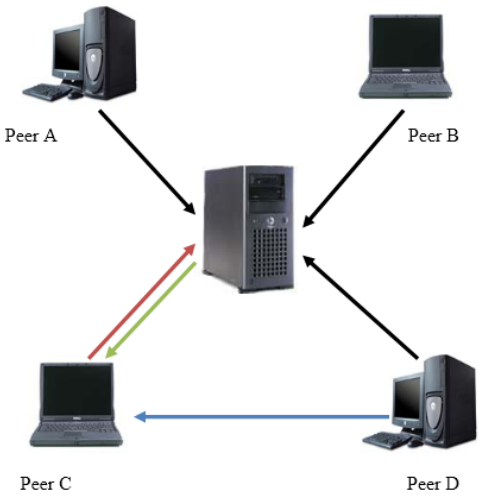
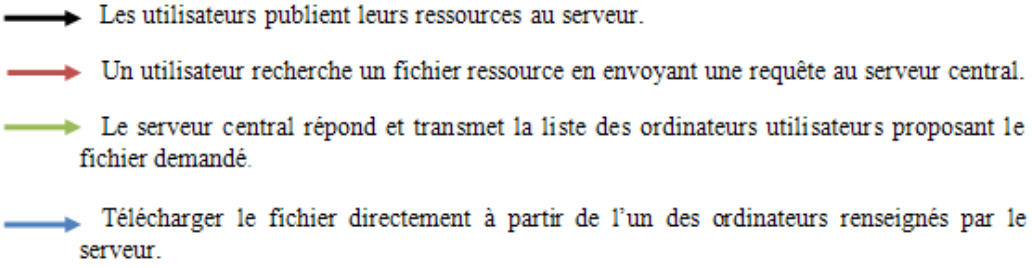


FIGURE 1.2 – Architecture P2P centralisée.



Avantages

- Simplicité : pas de soucis de connexion au bon serveur.
- Si un fichier est disponible sur le réseau, on est en mesure de le savoir systématiquement car la recherche de ressources est facile.
- Trafic réseau réduit : les pairs ne communiquent entre eux que s'ils ont quelque chose à échanger.

Inconvenients

- Une architecture P2P centralisée montre la vulnérabilité. elle a une seule porte d'entrée qui est le serveur central, il suffit de bloquer ce serveur pour déconnecter tous les utilisateurs et stopper le fonctionnement du réseau **[3]**.
- Une architecture centralisée ne garantie aucun anonymat car, chaque utilisateur peut connaitre l'adresse IP de la machine et le type de fichiers téléchargés des autres utilisateurs **[3]**.
- Sensible au partitionnement du réseau et aux attaques.

Exemple de Napster

Napster est l'un des plus anciens et plus célèbres des systèmes P2P hybrides dont lequel la recherche de services est centralisée **[10]**. Il est basé sur un système d'échange direct de fichiers musicaux entre pairs grâce à un serveur central. Seul le transfert de fichiers qui est décentralisé **[9]**. La figure1.3 représente l'architecture générale de Napster.

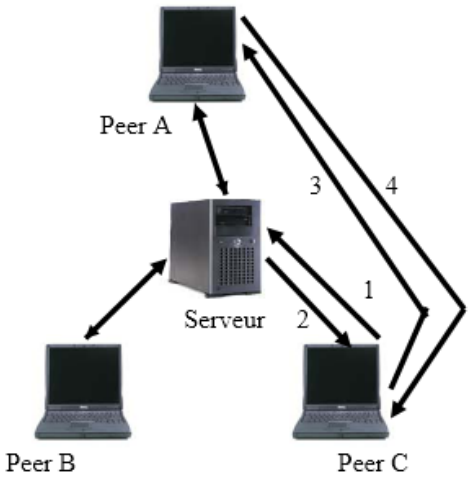


FIGURE 1.3 – Exemple de fonctionnement de Napster.

1. Demande du fichier au serveur.
2. Réponse contenant les pairs disposants ce fichier.
3. Demande directe de Peer C au Peer A pour télécharger le fichier.
4. Téléchargement de fichier à partir de Peer A.

1.7.2 Architecture décentralisée

Dans ce type d'architecture, l'ensemble des noeuds sont égaux et jouent le même rôle. Chaque pair gère donc les recherches et les partages, sans passer par un serveur central ou des supers pairs.

Dans cette architecture, le client diffuse une demande pour l'identification à tous les noeuds du réseau au quel il est connecté. Les noeuds recevant la demande l'envoyer à tous les autres pour répondre avec une identification **[5]**.

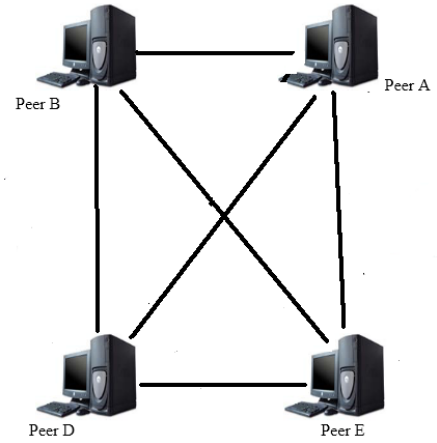


FIGURE 1.4 – Architecture P2P décentralisée

Le Peer A désire lancer une recherche d'information, cette demande va être transmise à tous les noeuds proches de A qui vont à leur tour la transmettre à leurs voisins.

A reçoit toutes les réponses correspondantes à sa demande, et un index de fichiers est créé en local sur A, alors l'utilisateur choisit les fichiers qu'il désire récupérer.

Avantages

L'architecture décentralisée propose plusieurs avantages **[6]** :

- La taille d'un réseau est infinie contrairement aux autres architectures dont le nombre de clients dépend du nombre et de la puissance des serveurs.
- L'utilisation d'un tel réseau est anonyme.
- Cette architecture est très tolérante aux fautes.
- Le réseaux est dynamique.

Inconvenients

- Le ralentissement des échanges de données entre pairs à cause de séries de broadcast qui sont diffusées sur le réseau **[3]**.
- Anonymat implique le risque de piratage **[7]**.

Exemple de Gnutella

Contrairement à Napster, Gnutella est un protocole de recherche P2P décentralisé, il a pris la suite en résolvant le problème de centralisation de Napster **[10]**. Il permet non seulement le transfert mais également la recherche de fichiers.

Comme il est illustré dans la figure 1.5, un pair se connecte sur le réseau, il commence par rechercher tous les pairs Gnutella, il transmet une trame d'identification (*PING*) à tous ces voisins, qui à son tour la transmettent á leurs voisins et ainsi de suite jusqu'à le pair 7. À chaque pair du réseau le TTL (*Time To Live*) qui représente le nombre de retransmissions est décrémenté et lorsqu'il s'annule, la retransmission est stoppée.

Pour éviter les boucles dans la transmission, la trame reçue est stockée pendant un court laps de temps. Lorsqu'un pair est identifié, il envoie à l'émetteur une trame de réponse (*PONG*) **[3]**.

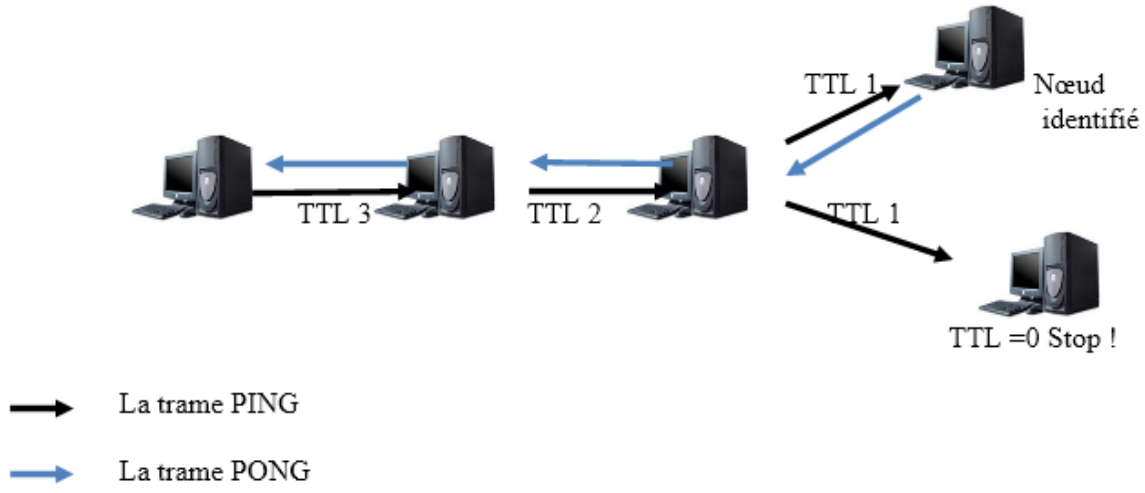


FIGURE 1.5 – Exemple de fonctionnement de Gnutella.

1.7.3 Architecture hybride

Cette architecture est la combinaison entre les deux architectures centralisée et décentralisée **[5]**. Les pairs sont reliés à des supers noeuds (*serveurs*) comme dans le modèle centralisé. Ces derniers sont organisés au sein d'un anneau faisant ainsi intervenir le principe de décentralisation. Un serveur qui contient un index sur l'ensemble de fichiers peut donc proposer à n'importe quel pair toutes les informations contenues sur le réseau (*exemple KazaA*).

Le modèle super noeud a pour but d'utiliser les avantages des deux types de réseaux (*centralisé et décentralisé*). En effet sa structure permet de diminuer le nombre de connexions sur chaque serveur, et ainsi d'éviter les problèmes de la bande passante **[3]**.

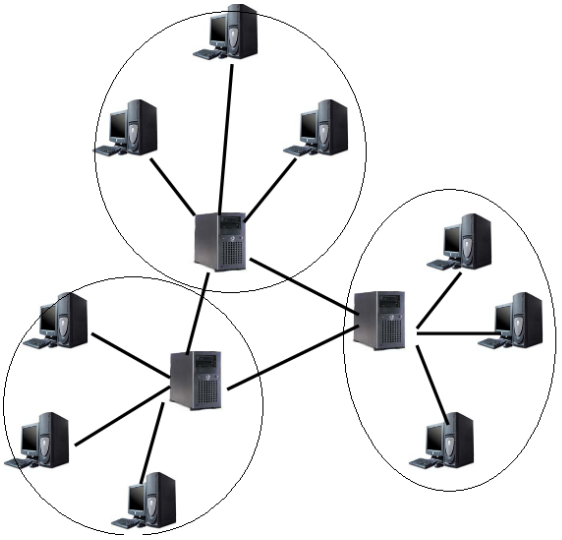


FIGURE 1.6 – Architecture P2P hybride.

Comme il est illusté dans la figure 1.6, un utilisateur lance une demande de recherche d'un pair. Cette dernière se fait par l'intermédiaire de l'ensemble des supers noeuds contenant l'ensemble des données des utilisateurs, il obtient alors une liste d'utilisateurs connectés et les ressources partagées correspondantes à la demande, il suffit alors de choisir le pair correspondant pour se connecter directement et entamer le transfert de ressources. Les supers noeuds sont donc pour acheminer des demandes lorsque les ressources trouvées.

Avantages

- La facilité d'administrer et de contrôler à cause de présence d'un serveur central.
- Éviter les recherches coûteuses sur le réseau.
- Tolérance aux fautes en recherchant régulièrement les pairs connectés.

Inconvenients

- Pas d'anonymat, car chaque pair est connu du serveur et des pairs sur lesquels il est relié.
- Certains pairs peuvent mentir sur leur débit pour ne pas être sollicités.
- Problème de disponibilité et de passage à l'échelle (*saturation de la bande passante et du nombre de noeuds*).

1.8 Autre classification des réseaux Peer to Peer

Les systèmes P2P peuvent être classifiés en non structuré et structuré.

1.8.1 Réseaux Peer to Peer structurés

Les P2P structurés sont basés sur la DHT (*Distributed Hash Table*), permettant de placer les nouveaux noeuds au sein du réseau. Chaque noeud reçoit une liste de voisins avec lesquels il pourra communiquer. Il s'agit ici de voisins logiques **[5]**. Un système P2P structuré garantit de trouver n'importe quel objet s'il existe.

1.8.2 Réseaux Peer to Peer non structurés

Les systèmes P2P non structurés n'impose pas de règles de connexion entre les pairs, ils supportent des opérations simples et efficaces d'arrivé et de départ des pairs.

Un système P2P est non structuré quand les liens entre les noeuds sont établis de façon arbitraire. La communication entre les noeuds est donc plus difficile à gérer **[5]** .

Le tableau 1.1 illustre les principales différences entre les différentes architectures de P2P.

Architecture P2P	**Avantages**	**Inconvénients**
Avec Serveur	-Recherche non directe -Charge minimale des pairs -Contrôle des pairs	-Existence d'un seul serveur -Confiance envers des serveurs inconnus
Réseaux non structurés	-Distribution complète -Recherche non directe	-Charge non ordonnée des pairs (*supers pairs*) -Localisation de nouvelles données (*chemin aléatoire*)
Réseaux structurés	-Distribution complète -Localisation de nouvelles données -Stockage des données	-Recherche directe

TABLE 1.1 – Comparison entre les architectures P2P.

1.9 Problèmes des réseaux Peer to Peer

Parmi les problèmes des réseaux Peer to Peer on cite **[7]** :

1.9.1 Sécurité

Dans l'architecture décentralisée des réseaux P2P, le téléchargement et le partage de fichiers se font directement entre machines, par conséquent, la sécurité sera touchée. Les fichiers téléchargés à partir d'autre systèmes peuvent être infectés par un virus qui rend les machines incapables d'authentifier les paramètres des autres machines avec lesquelles elles communiquent.

1.9.2 Interopérabilité

Les applications des réseaux P2P utilisent différentes technologies réseau avec les différentes plateformes. L'interopérabilité est difficile lorsque plusieurs plateformes puissent exister au sein de même réseau avec différents systèmes de sécurité.

1.9.3 Bande passante

Les problèmes de trafic et de saturation de réseau engendrés par le nombre d'utilisateurs du réseau P2P qui n'est pas limité.

1.9.4 Découverte des ressources

Comme il n y'a pas de serveurs dans les réseaux P2P, ainsi que les adresses IP ne sont pas toujours permanentes, l'utilisation d'un bon algorithme de découverte et de localisation des ressources est indispensable **[11]**.

1.10 Conclusion

Les réseaux Peer to Peer sont aujourd'hui de plus en plus présents dans le monde informatique, que ce soit pour les simples utilisateurs à travers les réseaux d'échanges de fichiers.

Nous avons vu dans ce chapitre comment le P2P sert à accéder et partager des fichiers en utilisant divers techniques (*premièrement il faut trouver le fichier*) et également en utilisant sur un même fichier des techniques de partage équitable (*assurer une résolution et un téléchargement rapide et fiable des fichiers dans le temps réel*).

Cependant, le pair à pair, ce n'est pas que l'échange de fichiers, mais il existe de nombreuses autres applications telles que la messagerie instantanée et la téléphonie sur IP qui sera détaillée dans la suite de ce rapport.

Chapitre 2

Protocole SIP (Session Initiation Protocol)

2.1 Introduction

La plupart des services multimédias sur IP demandent la création et la gestion d'une session, cette dernière est considérée comme un échange de données entre différents utilisateurs.

Plusieurs protocoles de signalisation ont été proposés pour l'établissement, le contrôle et la rupture d'une connexion en temps réel. Deux parmi les plus importants standards de la téléphonie IP sont : H.323 et SIP (*Session Initiation Protocole*) qui sera détaillé brièvement dans ce chapitre.

2.2 Définition de SIP

SIP a été défini à l'origine par le groupe MMUSIC (*Multiparty Multimedia Session Control*) de l'IETF (*Internet Engineering Task Force*), il est décrit par le RFC 3261 qui rend obsolète le RFC 2543, et est complété par le RFC 3265 en mars 1999 **[30]**.

SIP est un protocole de commande au niveau de la couche application, il peut établir, modifier et terminer des sessions multimédias telles que des communications téléphoniques sur Internet et les vidéoconférences.

SIP est un protocole fonctionnant sous TCP/IP (*Transmission Control Protocol/ Internet Protocol*), il se base plus précisément sur UDP (*User Datagram Protocol*) **[22]**.

SIP peut être utilisé avec d'autres protocoles de l'IETF pour construire une architecture multimédia complète qui vont inclure des protocoles tels que : Le protocole de transport de données en temps réel RTP (*Real Time Transport Protocol*) et le protocole de description de session SDP (*Session Description Protocol*) **[29]**.

SIP hérite certaines fonctionnalités de protocole HTTP (*Hyper Text Transport Protocol*) utilisé pour naviguer sur le WEB, et SMTP (*Simple Mail Transport Protocol*) utilisé pour transmettre des messages textuels **[14]**.

La figure 2.1 présente le positionnement de SIP dans la constellation des protocoles Internet **[13]**.

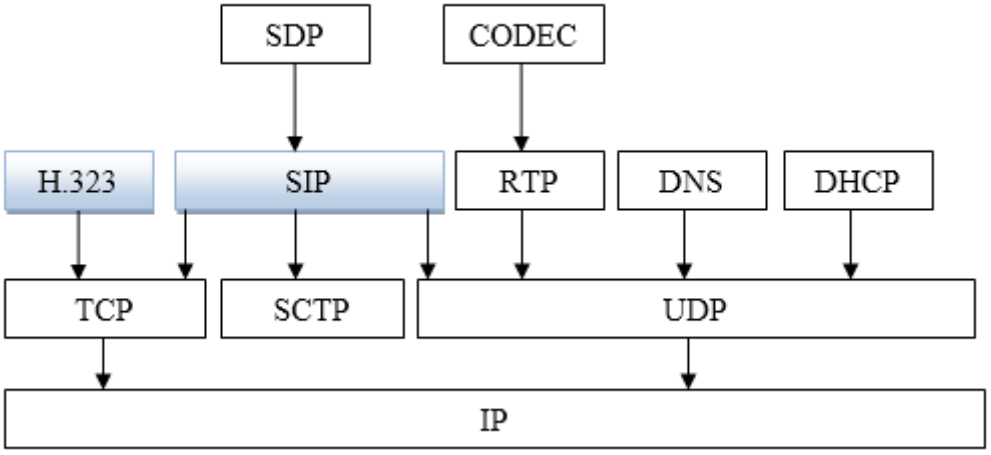


FIGURE 2.1 – Positionnement du protocole SIP parmi les protocoles Internet.

2.3 Caractéristiques de SIP

Les principales caractéristiques de SIP peuvent être résumées comme suit **[20]** :

2.3.1 Simplicité

SIP est un protocole léger qui nécessite peu de ressources physiques et peu de temps de développement.

2.3.2 Description générique de session

SIP sépare la signalisation des sessions de leurs descriptions, il peut être utilisé pour lancer et commander des nouvelles sessions.

2.3.3 Système d'adressage

L'adressage peut être un numéro de téléphone, une adresse IP ou une adresse Email. Les messages sont très semblables à ceux employés par HTTP.

2.3.4 Multimédia

SIP peut avoir des sessions médias multiples pendant un appel, il est indépendant de la transmission des données et de protocoles qui sont utilisés pour cet échange.

2.4 Fonctionnalités de SIP

Pour établir et terminer une communication multimédia, SIP propose cinq fonctionnalités :

- **Localisation d'un utilisateur :** déterminer (*localiser*) un utilisateur pour la communication.
- **Disponibilité d'un utilisateur :** contacter un utilisateur pour déterminer sa volonté d'établir une session.
- **Capacité d'un utilisateur :** échanger des informations sur les médias (*voix, vidéos*) pour permettre l'établissement d'une session.
- **Établissement de session :** est la demande d'ouverture des sessions médias existantes.
- **Gestion de session :** transférer, terminer et modifier les paramètres des sessions.

SIP peut être utilisé pour le contrôle de conférences multimédia, des appels téléphoniques sur IP et bien d'autres types de communication, qui peuvent être : Unicast (*point à point*) ou multicast (*diffusif ou combinatoire*) **[12]**.

- Point à Point : communication entre deux machines.

- Diffusif : plusieurs utilisateurs en multicast via une unité de contrôle MCU(*Multipoint Control Unit*).

- Combinatoire : plusieurs utilisateurs pleinement interconnectés en multicast via un réseau à maillage complet.

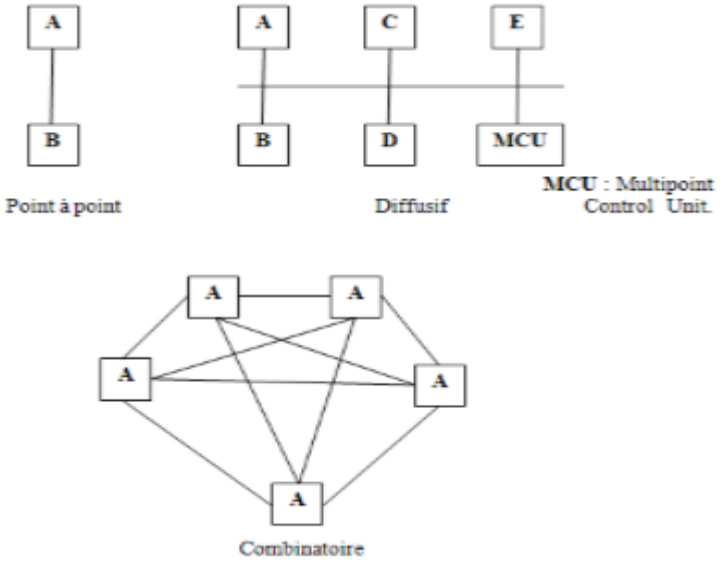


FIGURE 2.2 – Modes de communication SIP.

2.5 Messages SIP

Les messages SIP sont décrits dans la RFC 822, ils peuvent être à la fois des requêtes d'un client vers un serveur ou des réponses d'un serveur vers un client **[25]**.

Le format des requêtes et réponses est en effet similaire à celui utilisé dans le protocole HTTP, et les entêtes s'apparentent à celles utilisées dans le protocole SMTP **[38]**.

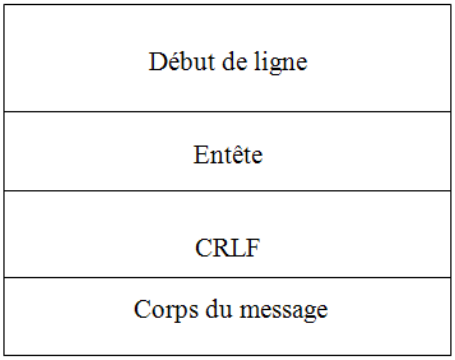


FIGURE 2.3 – Format des messages SIP.

- Début de ligne : il s'agit d'une requête ou d'une réponse.

- Entête : rassemble les entêtes du message (*entête général ou de requête ou d'entité*).

- CRLF (*Carriage Return Line Feed*) : assure la séparation de l'entête et du corps du message, ce qui permet d'optimiser le temps de traitement des messages.

- Corps du message : contient la description complète des paramètres de la session concernée.

2.5.1 Requêtes (*méthodes*) SIP

Le protocole SIP est constitué initialement de six requêtes :

- **REGISTER :** permet à un utilisateur d'enregistrer son adresse IP auprès du serveur auquel il est relié.

- **INVITE :** permet à un utilisateur de demander une nouvelle session.

- **ACK :** confirmer la réponse finale à une méthode INVITE.

- **OPTIONS :** obtenir les capacités de l'entité interrogée sur différentes informations.

- **BYE :** cette requête est utilisée pour terminer une session.

- **CANCEL :** annuler une session qui n'a pas encore été établie.

2.5.2 Réponses SIP

Les requêtes SIP sont acquittées par des réponses qui sont dans le même format que celles du protocole HTTP. Voici les plus importantes d'entre elles :

- **Classe 1xx :** informer que la requête a été reçue, et elle est en cours de traitement.
- **Classe 2xx :** succès, la requête reçue est comprise et acceptée.
- **Classe 3xx :** redirection, la session requiert d'autres traitements avant de pouvoir déterminer si elle peut être réalisée.
- **Classe 4xx :** erreur requête client.
- **Classe 5xx :** erreur serveur.
- **Classe 6xx :** échec global, la requête ne peut être traitée par aucun serveur.

2.6 Architecture de SIP

L'architecture SIP est composée de cinq types d'entités logiques, chaque entité a des fonctions spécifiques, elle participe à une communication SIP comme un client, un serveur ou les deux **[15]** :

- **Agent utilisateur (*User Agent*) :** désigne les agents que l'on retrouve dans les téléphones SIP (*ordinateurs, PDA*).
- **Serveur d'enregistrement (*Registar Server*) :** gère les requêtes REGISTER envoyées par les Users Agent pour signaler leur emplacement courant.
- **Serveur proxy (*Proxy Server*) :** permet d'acheminer uniquement les messages SIP pour établir, contrôler et terminer la session.
- **Serveur de redirection (*Redirect Server*) :** répondre à des requêtes SIP en donnant l'adresse IP d'un utilisateur ou d'un service de localisation.

- **Serveur de localisation (*Location Server*) :** contient la base de données de l'ensemble des utilisateurs, il est utilisé par un serveur proxy ou un serveur d'enregistrement.

La figure 2.4 illustre la communication entre deux Users Agent à travers SIP **[28]**.

- Un utilisateur A envoie un message REGISTER au serveur d'enregistrement qui l'achemine au serveur de localisation pour l'enregistrement (*messages 1, 2*).

- L'utilisateur B envoie une requête INVITE (*messages 3, 4, 5, 6 et 8*) à l'utilisateur A aprés avoir sa localisation en consultant le serveur de localisation (*message 7*).

- L'utilisateur A répond par OK (*messages 9, 10, 11*).

- L'utilisateur B envoie un ACK (*messages 12*) à l'utilisateur A.

- Échange de médias entre les deux utilisateurs (*messages 13*).

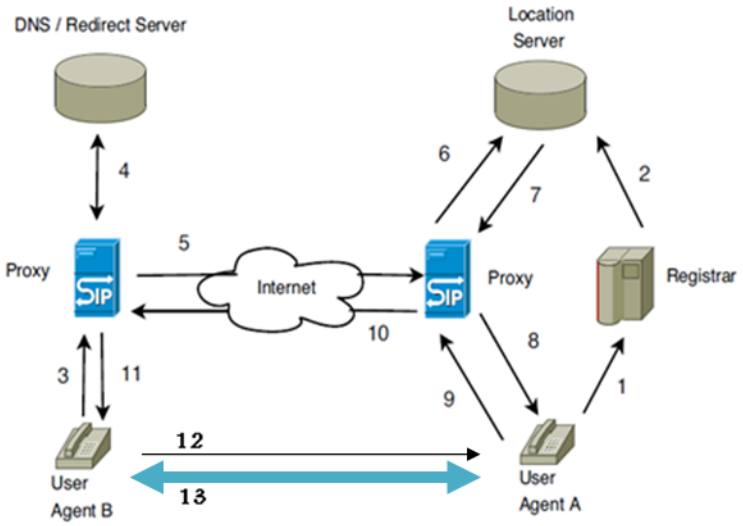


FIGURE 2.4 – Architecture de SIP.

2.7 Adressage SIP

SIP choisit l'Email comme adresse de la forme **[21]** :

« utilisateur@ nom domaine » pour contacter un utilisateur particulier.

L'utilisateur peut être représenté par le numéro de téléphone associé :
« numéro téléphone@ nom domaine ».

Le nom de domaine peut être décrit sous la forme d'une adresse IP numérique « utilisateur@ adresse IP ».

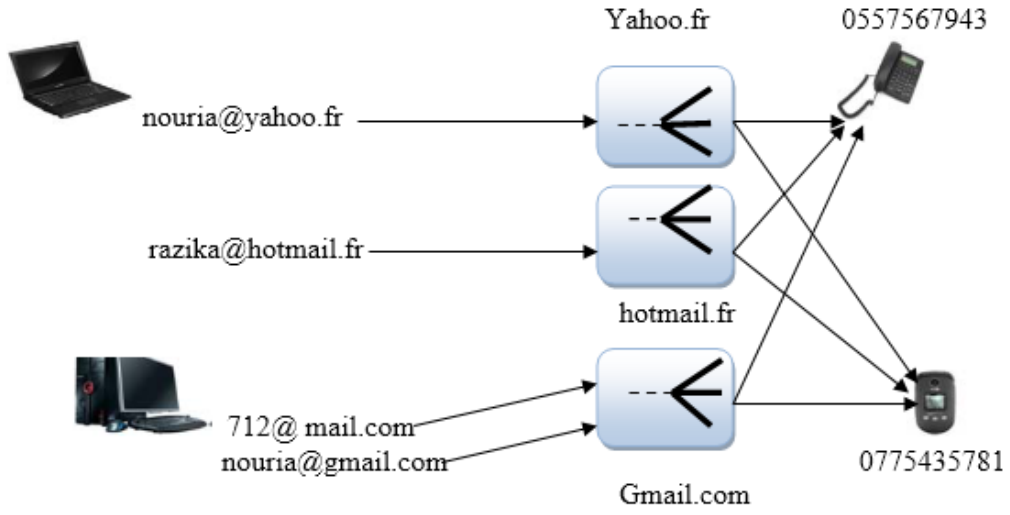


FIGURE 2.5 – Adressage SIP.

2.8 Exemple d'établissement d'une session

Dans l'exemple suivant représenté par la figure 2.6, l'appelant (*Nouria*) a une adresse nouria@yahoo.fr et l'appelé (*Razika*) a une adresse razika@hotmail.fr.

Nouria émet une méthode INVITE au Proxy Server. Ce dernier achemine la demande d'initiation de session à la destination (*Razika*).

La réponse 180 RINGING est retournée par Razika. Lorsque Razika accepte la session, elle envoie la réponse 200 OK qui achemine jusqu'à Nouria.

Nouria retourne une méthode ACK à Razika, la conversation entre les deux commence.

Le serveur proxy participe à l'acheminement de la signalisation entre UAs alors qu'ils établissent directement des canaux RTP pour le transport de la voix ou de la vidéo.

Lorsque Nouria raccroche, elle envoie une requête BYE au serveur proxy qui l'achemine à Razika pour terminer la session. Razika retourne la réponse 200 OK.

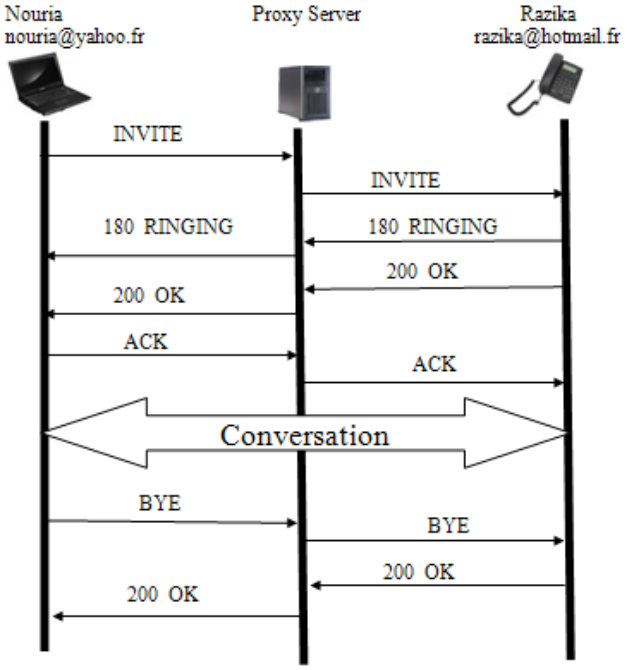


FIGURE 2.6 – Exemple d'établissement d'une session avec le protocole SIP.

2.9 Problèmes de SIP

Parmi les problèmes de SIP, nous citons **[24]** :

2.9.1 Pares-feu (*Firewalls*)

Un pare feu (firewall) est un équipement permettant d'assurer la sécurité d'un utilisateur, il permet de filtrer les paquets venant du réseau public. SIP, par son fonctionnement, permet la localisation des utilisateurs au sein d'un réseau et la négociation des paramètres de la session, pose des problèmes pour les flux multimédias qui traversent les firewalls. En effet, ces derniers permettant de rejeter tous les paquets qui ne proviennent pas ou qui ne

sont pas destinés à une adresse IP et un port défini, ils ne permettent pas la traversée d'un flux de données de paramètres inconnus à des protocoles comme SIP lors de l'établissement d'une session.

2.9.2 NATs (*Network Address Translation*)

NAT permet de faire correspondre les adresses IP internes. Puisque ces dernières sont employées pour la communication, la transition échouera si l'un des UAs est derrière un NAT, ce qui représente un problème pour le transit des flux multimédias. En effet, les informations utilisées pour la signalisation ou la communication sont incluses au niveau 4 et les couches supérieures du modèle OSI (*Open System Interconnexion*), tandis que les NATs travaillent sur la couche 3.

2.9.3 Complexité

Lorsque vous appelez un téléphone traditionnel, l'appelant peut recevoir une voix féminindisant que le numéro que vous avez appelé n'est pas disponible actuellement, donc l'appel n'est pas réussi. Dans SIP, cela signifie qu'il n y'a pas 200 OK alors, le SIP User Agent ne peut pas démarrer l'écoute de messages avant le 200 OK **[16]**.

2.10 Conclusion

À l'heure actuelle, SIP se présente comme un protocole de signalisation le plus adéquat aux applications de voix et vidéo sur IP. Sa simplicité relative par rapport aux autres standards, le rend de plus en plus populaire dans ce domaine.

Dans le chapitre suivant, nous présentons quelques architectures de la téléphonie IP en mode Peer to Peer en utilisant le protocole SIP (*P2P-SIP*).

Chapitre 3

État de l'art sur la téléphonie Internet en mode P2P-SIP

3.1 Introduction

La voix sur IP (*VoIP*) est devenue une technologie réelle dans le monde pour la communication.

La téléphonie IP basée sur le protocole SIP peut être traitée comme un système de P2P avec un ensemble de supers noeuds (*serveurs SIP*). Cependant, employer une architecture de P2P améliore la fiabilité du système.

Plusieurs travaux de recherche ont été fait ces derniers temps pour la définition d'un protocole P2P standard pour la voix sur IP. Les solutions proposées cherchent à approuver une solution P2P basée sur le protocole SIP.

Dans ce chapitre nous présentons les principales architectures P2P-SIP telles que l'architecture de Henning Schulzrinne, Skype et Sosimple.

3.2 Généralités sur la téléphonie IP (*ToIP*)

La plupart des téléphones sont connectés au réseau téléphonique classique, qui est un outil de communication à large échelle grâce a son efficacité, il est basé sur la technologie de commutation de circuits.

Les services de téléphonie sur IP doivent pouvoir accepter tout trafic provenant de ces réseaux avec l'utilisation du réseau RTCP (*Réseau Téléphonique Commuté Public ou RTC*).

La téléphonie sur IP (*ToIP*) est la transmission de la voix en utilisant le protocole IP, elle repose sur deux principes **[26]** :

- Découpage du flux voix numérisé en une suite de paquets pour diminuer la quantité d'informations à transmettre.

- Transit sur le réseau IP : les paquets sont réassemblés avant d'être transportés sur le réseau.

3.2.1 Téléphonie sur IP et le RTC

Le réseau téléphonique public RTPC à essentiellement pour objectif de transfert de la voix, il donne accès à de multiples fonctions **[17]**.

En effet outre le fait de pouvoir téléphoner, le RTC permet d'utiliser multiples services tels que la transmission et la réception de fax, ainsi que l'accès à Internet.

La figure 3.1 illustre la téléphonie sur IP et le RTC.

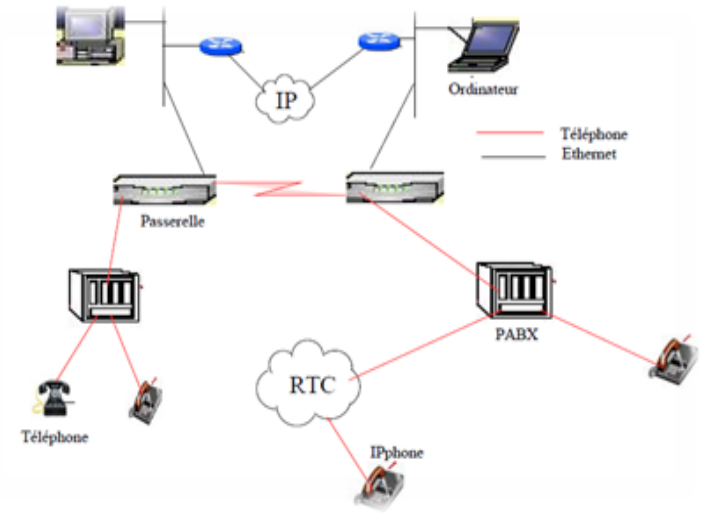


FIGURE 3.1 – Téléphonie sur IP et le RTC.

3.2.2 Modéle de la ToIP

Selon le type de terminal utilisé, il existe trois modèles différents de la ToIP : La ToIP de PC à PC, la ToIP de PC à téléphone et la ToIP de téléphone à téléphone **[23]**.

3.2.2.1 Modèle de PC à PC

Dans ce modèle (*voir la figure 3.2*), les deux correspondants utilisent un PC relié au réseau Internet. Chaque ordinateur est muni une carte de son, microphone et haut-parleur, il se connecte directement au réseau Internet par l'intermédiaire de modem ou d'un fournisseur d'accès à l'Internet. Cette technique nécessite d'avoir un logiciel de téléphonie compatible à chaque coté.

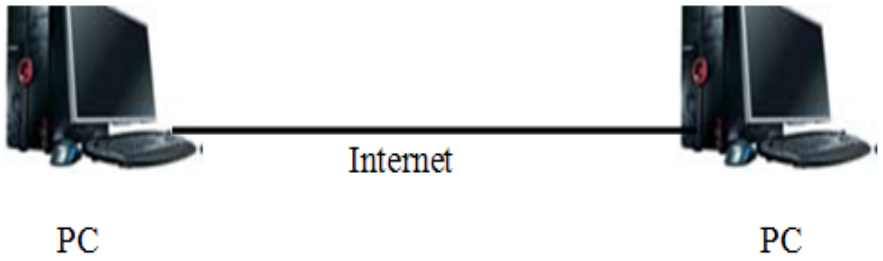


FIGURE 3.2 – Modèle de PC à PC.

3.2.2.2 Modèle de PC à téléphone

Dans ce modèle (*voir la figure 3.3*), l'un des correspondants utilise un PC relié au réseau Internet par un fournisseur d'accès à Internet et l'autre utilise un téléphone relié au réseau téléphonique commuté. Une passerelle (*Gateway*) est nécessaire entre les deux réseaux Internet et le RTC.

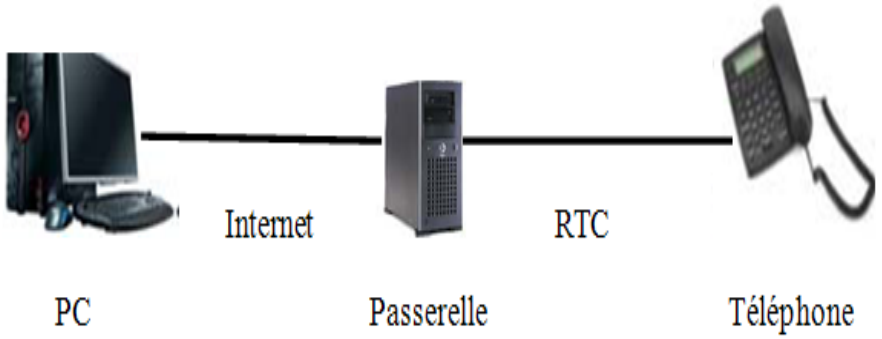


FIGURE 3.3 – Modèle de PC à téléphone.

3.2.2.3 Modèle de téléphone à téléphone

Dans ce modèle (*voir la figure 3.4*), les deux correspondants utilisent un téléphone, il utilise le réseau Internet pour communiquer entre les réseaux RTC. Une passerelle est nécessaire pour chaque coté entre les réseaux RTC et le réseau Internet pour la conversion de la voix.

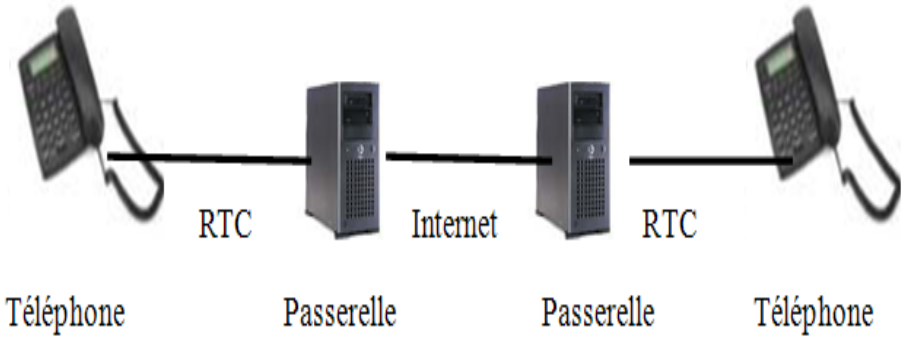


FIGURE 3.4 – Modèle de téléphone à téléphone.

3.2.3 Composants de la ToIP

L'établissement de la ToIP doit avoir les composants suivants :

- **Terminaux :** sont des ordinateurs installant des logiciels supportant la ToIP, des téléphones IP. Ils établissent et terminent des appels.

- **Serveurs :** sont des serveurs qui gèrent les réseaux de la ToIP.

• **Passerelles :** sont des ponts pour connecter entre les composants des réseaux de la ToIP et les composants d'autres réseaux.

3.2.4 Avantages de la ToIP

Plusieurs raisons expliquent le succès de la téléphonie sur IP **[38]** :

3.2.4.1 Convergence

Les flux de voix, de vidéo transitent sur le même réseau. Les communications deviennent plus riches, et sans avoir besoin de multiplier les canaux de transport. Les utilisateurs peuvent, par exemple, envoyer un compte rendu d'activité en même temps qu'ils téléphonent à leur correspondant.

3.2.4.2 Optimisation des ressources

Dans le réseau RTC, les ressources sont dédiées pour toute la durée de la communication, qu'elles soient utilisées ou non. Lors d'une conversation téléphonique, une seule personne qui parle en même temps. Les ressources sont donc globalement gaspillées. C'est pourquoi la réservation effectuée dans les réseaux RTC représente un coût supérieur à celui des réseaux IP.

3.2.4.3 Coût de transport quasiment nul

Le réseau permettant d'effectuer le transport est celui qui effectue tous les transports de données. Les opérateurs, qui obligent de maintenir au moins deux réseaux, celui de téléphonie et celui de données, n'en ont plus qu'un seul à maintenir.

3.2.4.4 Services spéciaux

Certains services sont propres aux réseaux IP. Par exemple, le service de présence, consistant à détecter si un utilisateur est connecté au réseau ou non, ne nécessite aucune réservation de ressources dans un réseau IP, à la différence du réseau RTC.

3.2.5 Problèmes de la ToIP

Parmi les problèmes de la téléphonie Internet on cite **[38]** :

3.2.5.1 Sécurité

Dans les versions classiques de la téléphonie, la sécurité est fortement garantie par un réseau spécifique. Avec le multimédia et l'intégration de la téléphonie dans l'ensemble de données deviennent particulièrement complexe de sécuriser l'application de ToIP (*lauthentification et la confidentialité*).

3.2.5.2 Disponibilité

La disponibilité désigne le temps pendant lequel un système est en état de marche ou non, ce qui revient au même, le temps pendant lequel le système est en panne.

3.2.5.3 Gestion

La gestion de l'environnement téléphonique devient beaucoup plus complexe, il faut inclure les flux de la voix dans le système de gestion du réseau, en particulier pour les pannes et la comptabilité.

3.2.5.4 Contrôle

Le contrôle complet de la gestion du réseau afin d'optimiser sa configuration pour que les contraintes de la téléphonie soient satisfaites.

3.2.5.5 Qualité de service

La téléphonie par paquets est une application complexe. Si le réseau est commuté, les chemins utilisés pour le transport des paquets peuvent être dimensionnés par des techniques d'ingénierie (*génial ou pratique*) de trafic.

3.3 Téléphonie IP basée sur le protocole SIP

La téléphonie IP basée sur le protocole SIP a une architecture client/serveur **[18]**. Lorsqu'un utilisateur (*ex. Razika*) lance une application SIP sur son ordinateur ou sur un autre appareil qui supporte la téléphonie IP, il s'inscrit dans le serveur SIP en indiquant l'adresse IP de son appareil. Le serveur stocke et enregistre toutes ces informations.

Comme il est montré dans la figure 3.5, quand un autre utilisateur (*ex. Nouria*) appelle Razika, il envoie son appel au serveur qui l'achemine vers la destination demandée.

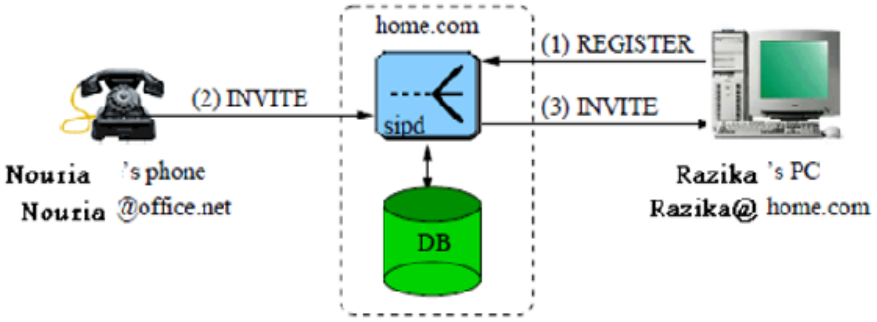


FIGURE 3.5 – Téléphonie IP basée sur le protocole SIP.

3.4 Téléphonie IP basée sur le P2P-SIP

La téléphonie IP par P2P en utilisant le protocole d'initiation de session (*SIP*) a été proposée pour minimiser le coût de maintenance et de configuration des serveurs dans l'architecture classique SIP, ainsi que pour éviter les pannes de ces serveurs **[39]**.

Cependant, l'utilisation d'une architecture P2P améliore la fiabilité et permet au système de s'adapter dynamiquement aux défaillances des noeuds. La figure 3.6 montre l'architecture de P2P-SIP.

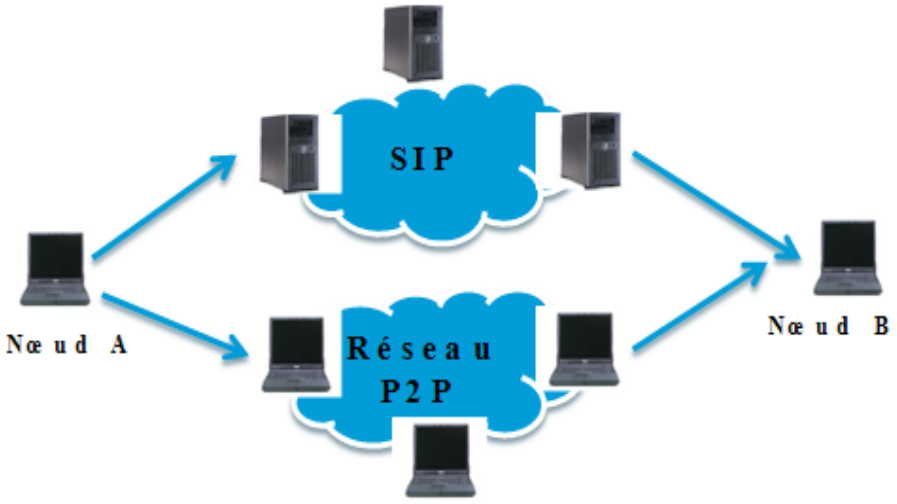


FIGURE 3.6 – Architecture de P2P-SIP.

3.4.1 Fonctions de P2P-SIP

Plusieurs services de P2P-SIP peuvent être présentés pour améliorer la compréhension des fonctions effectuées par chacun **[40]**. La figure 3.7 illustre les principales fonctions de P2P-SIP :

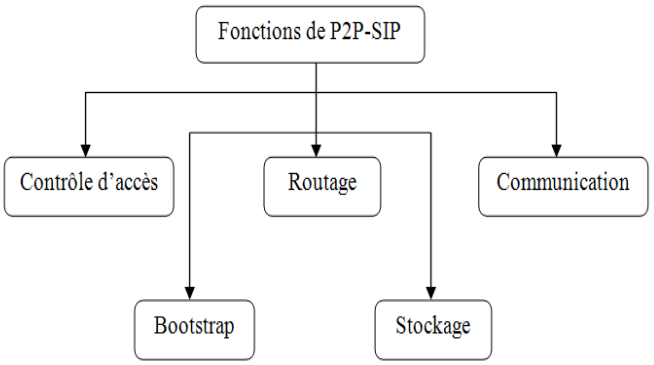


FIGURE 3.7 – Fonctions de P2P-SIP.

3.4.1.1 Contrôle d'accès

Le contrôle d'accès est le service responsable de décider quels utilisateurs sont permis de rejoindre le réseau et utiliser ses ressources. Une fois la décision a été prise, le service doit assigner à chaque utilisateur une identification unique qui l'identifie dans le réseau.

3.4.1.2 Bootstrap

Le Bootstrap est le processus par lequel un noeud communique avec d'autres noeuds déjà reliés au réseau afin de s'identifier et être capable de l'utiliser dans le réseau. Pendant ce processus, le nouveau noeud qui se place dans le réseau, informe ses voisins de son emplacement par la table de routage.

3.4.1.3 Routage

Le routage est responsable de transmettre tous les messages échangés entre les noeuds de réseau P2P-SIP. Ces messages arrangés sous forme d'une requête ou une réponse pour contrôler et maintenir le réseau.

3.4.1.4 Stockage

Le stockage consiste à sauvegarder les informations des utilisateurs du réseau afin de les permettre communiquer avec les autres. À la différence du réseau client/ serveur où le stockage est consacré pour le serveur, dans les réseaux P2P, les ressources distribuées par tous les noeuds.

3.4.1.5 Communication

Le rôle de la communication est d'établir des services de communication telles que la transmission de messages, la téléphonie et la vidéoconférence.

3.4.2 Différentes architectures P2P-SIP

Les architectures de P2P-SIP supportent l'enregistrement et la recherche des utilisateurs, l'établissement des appels, ainsi que le stockage des messages off-line (*lorsque l'utilisateur est absent ou déconnecté*) et la voix/vidéo.

3.4.2.1 Architecture de Henning Schulzrinne

Un noeud ordinaire se connecte à son super noeud d'attache et lui envoie un message de localisation d'un autre noeud ordinaire. Le message est acheminé au super noeud responsable de la clé de destination. Ce dernier l'envoie au super noeud d'attache du noeud ordinaire distant.

Ce pair d'attache traite le message et envoie ainsi la réponse (*contenant l'adresse du noeud distant*) au noeud ordinaire local en passant par son super noeud d'attache. Après la localisation, la communication passera directement entre les deux noeuds ordinaires.

Chaque noeud peut être un super noeud ou un noeud ordinaire suivant ses capacités. Le coût de la recherche dans cette architecture est de O(log n). La solution est coûteuse en maintenance et en stabilité s'il y'a fréquemment des nouveaux super noeuds qui arrivent ou quittent **[19]**.

La figure 3.8 présente l'architecture réseau de Henning Shulzrinne.

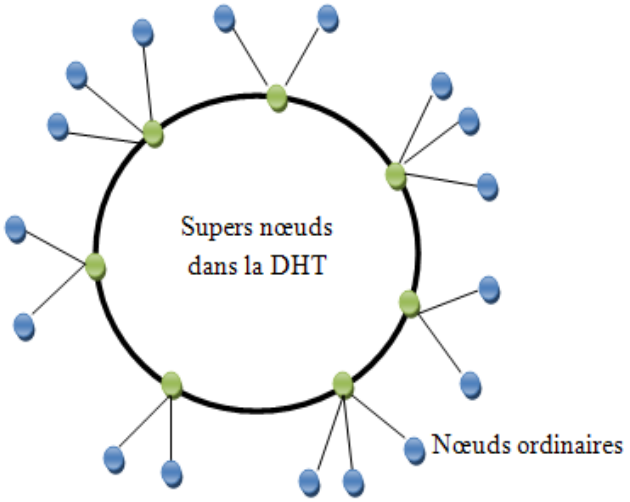


FIGURE 3.8 – Architecture réseau de Henning Schulzrinne.

1. **Enregistrement d'un utilisateur**

Lorsqu'un nouvel utilisateur arrive, il indique son nom (*ex. Nouria@office.com*). En utilisant la table de hachage distribuée (*DHT*), il obtient une clé à partir de son nom et non pas de son adresse IP (*la clé du noeud et celle de l'utilisateur sont calculées séparément*), ensuite il se place dans le réseau P2P (*La clé de Nouria est 42*). Un message SIP REGISTER est utilisé à la fois pour l'insertion et l'enregistrement d'un nouveau noeud dans le réseau.

Quand un autre utilisateur qui possède l'adresse de Nouria veut parler avec lui, il calcule sa clé par la même fonction de hachage en utilisant son nom (*ex. Razika qui a la clé 12*) et il lance la requête recherchée (*clé 42*).

Le noeud utilise son adresse IP pour calculer sa clé (*ex. 14*) et il prend place dans le réseau. Ensuite, il calcule la clé de l'utilisateur en utilisant son nom et publie cette clé par un message REGISTER de clé 42. Le noeud 58 qui est responsable de la clé 42 accepte la registration et maintient dans sa table de routage que la clé 42 se trouve dans le noeud 58.

Si Nouria n'est pas présente alors Razika peut lui envoyer des messages off-line avec la clé 56, qui vont être délivrés à Nouria quand elle devient on-line.

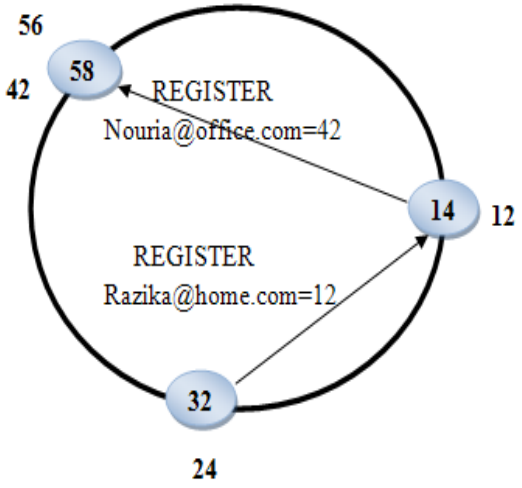


FIGURE 3.9 – Enregistrement d'un utilisateur.

2. **Architecture fonctionnelle**

Un noeud P2P-SIP peut être composé de différents blocs qui lui permettant de s'enregistrer dans le réseau P2P, de localiser d'autres utilisateurs, de transmettre et de recevoir des messages off-line, de détecter les Firewalls et les NATs derrière lesquels il est, comme il est montré dans la figure 3.10.

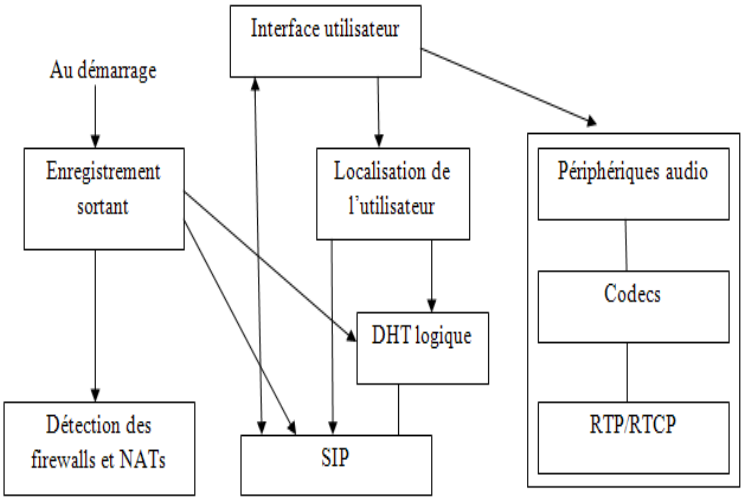


FIGURE 3.10 – Diagramme d'un noeud P2P-SIP.

3. **Procédure d'enregistrement d'un utilisateur**

Lorsqu'un noeud rejoint le réseau, il doit s'enregistrer par son adresse IP pour pouvoir être localisé dans le réseau P2P, et par son nom pour qu'il puisse transmettre et recevoir des messages off-line.

- La procédure d'enregistrement SIP est utilisée par le noeud qui cherche les adresses de SIP en utilisant DNS (*Domain Name Server*) **[31]** et envoie un message SIP REGISTER.

- La procédure de découverte des pairs est utilisée par le noeud qui essaye de découvrir les supers noeuds existants.

La figure 3.11 représente le diagramme d'enregistrement d'un noeud dans un réseau P2P-SIP.

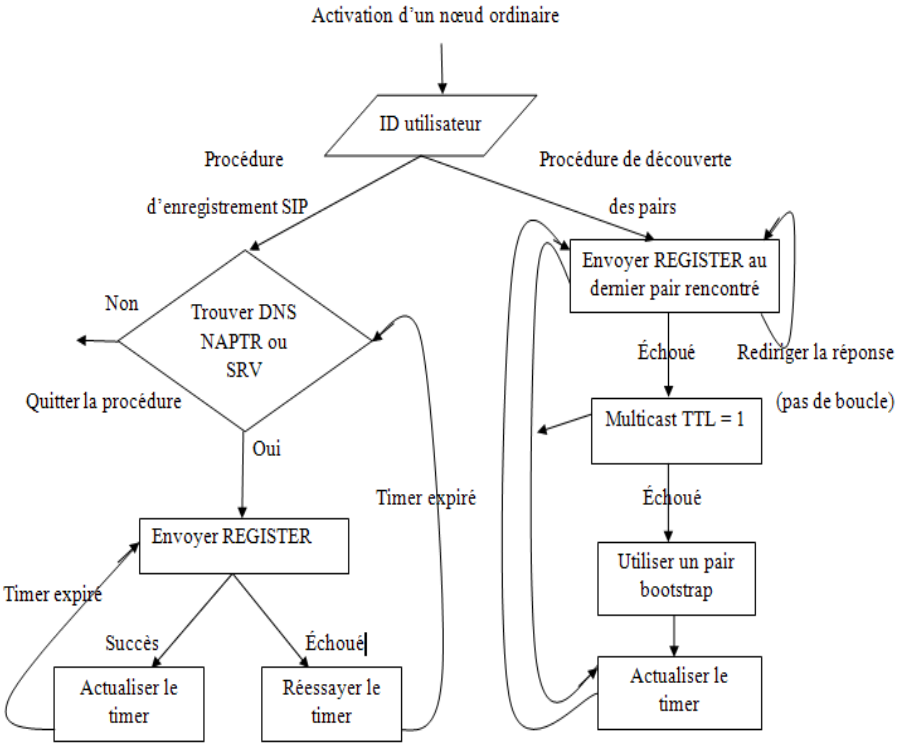


FIGURE 3.11 – Diagramme d'enregistrement d'un noeud.

4. **Recherche d'un utilisateur**

Lorsqu'un utilisateur B qui a la clé 38 veut communiquer avec un autre A qui a la clé 62 comme il est illustré dans la figure 3.12 ci dessous, il calcule la clé de ce dernier, puis il demande au super noeud sur lequel il est attaché de lui chercher le noeud de clé 62.

Le super noeud s'occupe de la recherche dans le réseau P2P en utilisant un algorithme de recherche comme Chord ou CAN **[27]** et fournit le résultat au Peer demandeur. Ainsi, l'user B peut communiquer avec l'user A sans passer par les supers noeuds.

La figure 3.12 illustre la recherche d'un utilisateur dans un réseau P2P-SIP.

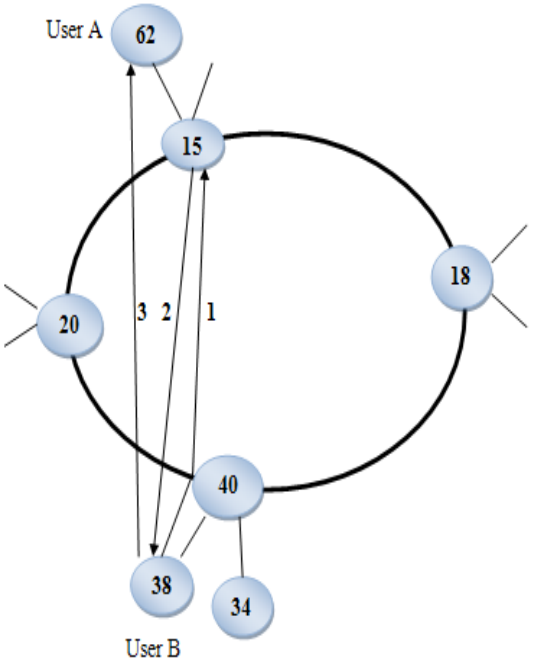


FIGURE 3.12 – Recherche d'un utilisateur.

5. **Départ d'un noeud**

Il existe deux types de défaillance **[33]** comme il est représenté dans la figure 3.13.

- **Départ d'un super noeud**

Lorsqu'un super nœud quitte le système, il doit faire des mises à jour concernant ces noeuds ordinaires ainsi que ses voisins avant qu'il quitte. Les noeuds ordinaires reliés à ce dernier seront ré-attachés à un autre super noeud.

- **Départ d'un noeud ordinaire**

Lorsqu'un nœud ordinaire quitte le système, le super nœud sur lequel il est relié peut détecter sa défaillance par l'absence de message de rafraichissement périodique. Il peut confirmer par l'envoie de message OPTIONS au nœud défaillant, s'il ne reçoit aucune réponse, détecte que le nœud est défaillant.

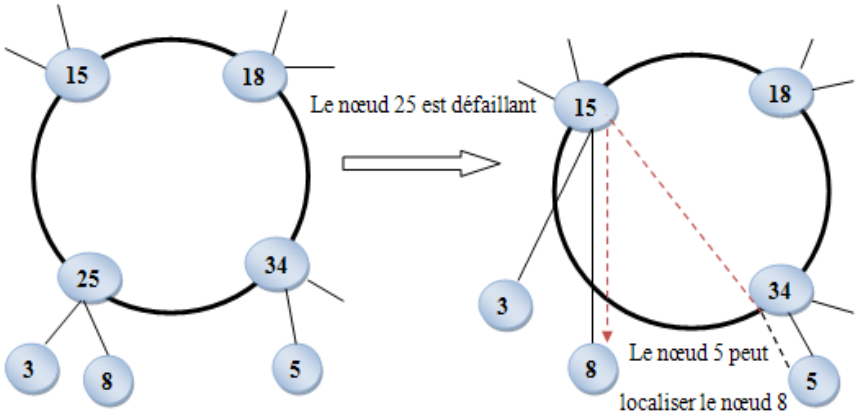


FIGURE 3.13 – Défaillance d'un super noeud dans la DHT.

3.4.2.2 Sosimple

David A. Bryan et Bruce **[37]** proposent une approche entièrement pair à pair de SIP, nommée SOSIMPLE dans le projet P2PSIP de l'IETF **[36]**.

SOSIMPLE est un système de P2P-SIP fournit la VoIP et IM (*Instant Messaging*) avec une DHT **[36]**, il combine la famille de SIP/simple, il est basé sur le protocole Chord pour la transmission fiable des messages de SIP. La DHT est utilisé seulement pour les communications directes entre les clients.

SOSIMPLE utilise un réseau P2P décentralisé structuré lors de son localisation , un noeud A publie ces ressources dans la DHT, contenant son IP réelle. Quand le noeud B veut contacter le nœud A, il effectue une recherche dans la DHT, il obtient l'adresse IP de A et connecter avec lui. La localisation du destinataire est réalisée grâce à la DHT et pas à un serveur **[23]**.

1. **Arrivée d'un nouveau nœud**

Lorsqu'un nouveau nœud rejoint le réseau, il doit d'abord localiser un certain nœud dans le réseau, qui lui sert de bootstrap (*un circuit fermé de noeuds*). Actuellement ce noeud est localisé par des mécanismes hors bande **[37]**. Il doit aussi échanger un certain nombre de messages REGISTER en employant son adresse IP pour calculer l'ID-Noeud.

Le nouveau noeud doit trouver le nœud actuellement responsable pour transformer l'information, il calcule son ID- Nœud, dans notre exemple 503 (*voir la figure 3.14*), et l'envoie dans un message REGISTER au nœuds de bootstrap, qui a ID- nœuds 023 (*1*).

Supposant que le bootstrap n'est pas le noeud actuellement responsable de ce ID-Noeud, il répond avec des informations des noeuds les plus proches (*le nœuds B avec ID- Nœud 445*). Cette information est passée en têtes dans une réponse SIP 302 déplacée Temporairement (*2*).

Le nouveau nœud répète le processus, en utilisant ce nœud plus proche comme nouveau nœud de bootstrap (*3 et 4*).

Enfin, le nouveau nœud atteint le nœud qui est actuellement responsable de son ID dans le réseau (*le nœud C avec ID- Nœud 520*). Le nœud C répond avec une réponse 200 OK comprenant l'information détaillée, dans l'entête du message sur les voisins (*5 et 6*), laissant le nouveau nœud de s'insérer dans le réseau.

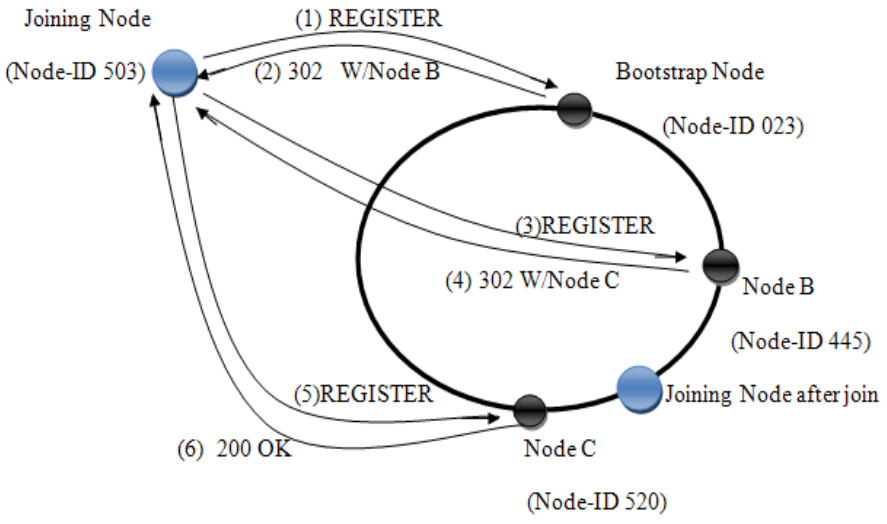


FIGURE 3.14 – Arrivé d'un noeud.

2. **Localisation d'un nouveau nœud**

Quand un noeud souhaite trouver le noeud responsable d'un utilisateur particulier pour le contacter, il commence par le brouillage (*hachage*) du nom d'utilisateur pour produire un ID-Ressource **[37]**. Le noeud de l'utilisateur est déjà placé correctement dans le réseau (*l'enregistrement de noeud est déjà produit ou établi*).

Comme il est illustré dans la figure 3.15, le nœud de Nouria lance la recherche d'une ressource (*ex. Razika*), il cherche dans sa table de raccourcis le noeud qui a un ID-Noeud le plus proche d'ID-Resource à chercher (*ex. noeud A*).

Le noeud Nouria envoie un message au nœud A (*1*). Le nœud A n'est pas responsable de cet ID- Resource, il envoie une réponse temporairement déplacée, y compris le nœud le plus proche (*ex. nœud B*), dans l'entête (*2*).

Le nœud Nouria essaye le nœud B et reçoit encore une réponse avec un autre nœud (*ex. nœud C*) (*3 et 4*). Enfin, le nœud de Nouria essaye le nœud C, ce qui est responsable de cette ressource (*5*).

Si Nouria s'enregistrait en tant qu'utilisateur dans le réseau (*enregistrement d'utilisateur*), il envoie un message de type REGISTER (*1, 3, 5*).

Quand le nœud C reçoit ce message, il se rend compte qu'il est responsable de stocker l'enregistrement de nom de Nouria, son adresse IP et l'envoie des réponses de 200 OK (*6*).

Lorsque Nouria à l'adresse IP de Razika, un appel entre ces deux utilisateurs peut être établi directement (*7*).

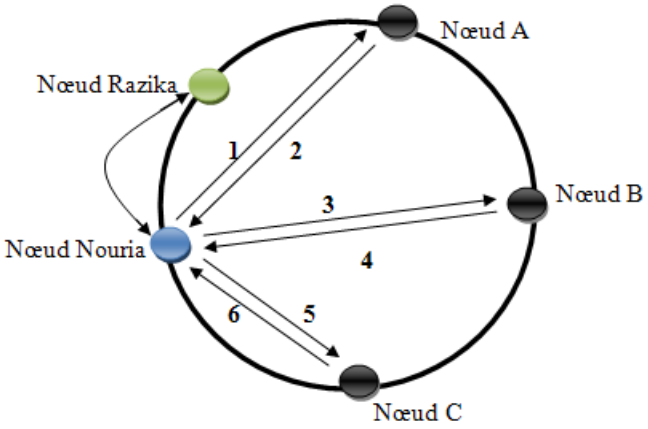


FIGURE 3.15 – Localisation d'un nouveau noeud.

3.4.2.3 Skype

En 2003, les créateurs du logiciel de partage de fichiers P2P KaZaA, publient la première version (*Beta*) de Skype. C'est à la fin de l'été 2004 que la version 1.0 du logiciel est mise gratuitement à la disposition du public **[32]**.

Skype **[35]** est un réseau de la téléphonie Internet (*VoIP*) qui s'appuie sur une architecteur P2P pour transmettre les flux de voix ou de données sur Internet, chaque client skype (*CS*) se comporte à la fois comme un client et un serveur. Au sein du réseau skype, il existe deux types de nœud : Les clients skypes (*nœuds ordinaires*) et les supers nœuds.

Tous les CSs possédant une adresse IP publique peuvent prendre le rôle de super nœud (*SN*) si leurs capacités requises en termes de bande passante et de puissance CPU les permettent. Ceux-ci agissent bien entendu également comme des nœuds ordinaires en permettant de transmettre leurs propres flux.

La figure 3.16 montre l'architecture générale du réseau skype.

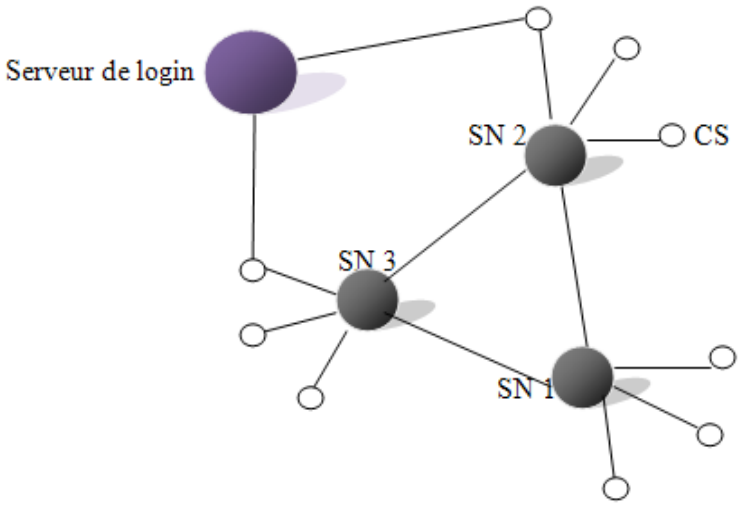


FIGURE 3.16 – Architecture de Skype.

1. **Arrivé d'un utilisateur**

Lorsqu'un nouvel utilisateur veut utiliser le skype, il faut d'abord s'authentifier auprès d'un serveur de login. Le nom d'utilisateur et le mot de passe sont comparés aux informations stockées sur le serveur.

Si l'authentification se passe avec succès, le serveur de login publie que l'utilisateur authentifié est en ligne et identifie les SNs que le client skype maintient la connexion au réseau **[34]**.

Le super nœud a le rôle de concentrateur manipulant la liste des contacts et route les appels à leur destination. Quand une recherche d'utilisateur est effectuée, le SN consulte son index, afin de trouver tout utilisateur qui s'est connecté durant les dernières 72 heures **[34]**.

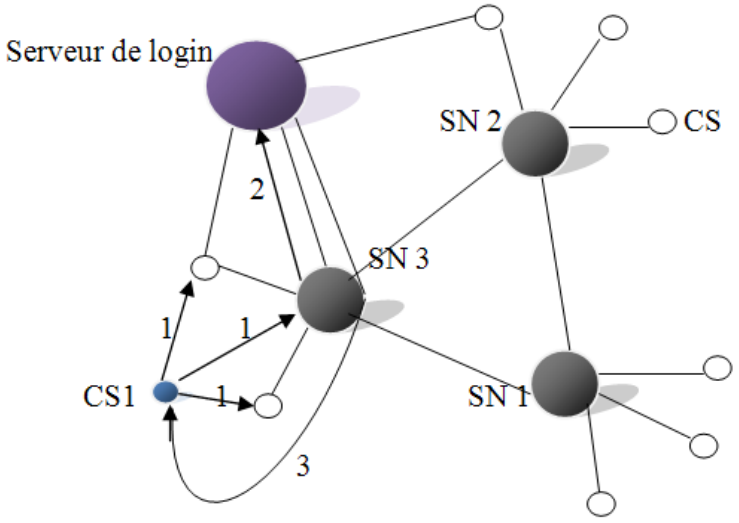


FIGURE 3.17 – Arrivé d'un nouvel utilisateur.

- CS1 envoie une demande d'authentification à tous les nœuds les plus proches.

- SN3 fait suivre cette demande jusqu'à le serveur de login.

- le serveur de login authentifie SC1 et envoie la réponse à SN3 qui l'achemine jusqu'à SC1.

1. **Recherche d'un utilisateur**

Quand le client Skype se connecte au réseau, il envoie des informations concernant sa présence à plusieurs supers nœuds qui les partagent avec d'autres. Le SN doit avoir un index global sur les CSs qui sont connectés au réseau.

Lorsqu'un CS désire rechercher un autre utilisateur, il envoie une requête au SN se trouvant dans son index. Si l'utilisateur recherché n'est pas connu par le SN, ce dernier lui renvoie une liste de SNs où il peut probablement trouver cet utilisateur. Le CS renvoie la même requête à tous les SNs de la liste.

Cette fonction de recherche continue jusqu'à trouver l'utilisateur recherché, sinon le CS affichera que la recherche est échouée. Dans le cas contraire, l'utilisateur sera enregistré dans la liste des amis pour éviter de refaire la recherche à chaque fois.

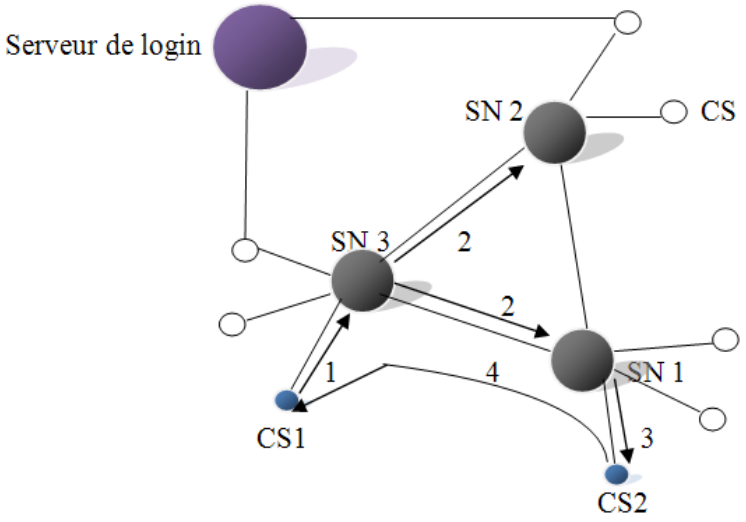


FIGURE 3.18 – Localisation d'un utilisateur.

- CS1 envoie une demande de recherche CS2 à super nœud le plus proche (*SN3*).

- SN3 vérifie son index. s'il ne contient pas le client skype recherché, il achemine la demande à tous les autres SNs.

- SN1 vérifie son index et il trouve le client recherché (*CS2*).

- SN1 envoie une réponse de succès à SN3 (*le client a été trouvé*) et demande d'enregistrer CS2 dans l'index de CS1.

3.4.2.4 Comparaison entre les architectures de P2P-SIP

A l'issue de la représentation des principales architectures P2P-SIP, la conclusion porte sur la manière de rechercher ou de contrôler les différents nœuds dans un réseau P2P. Ceux-ci posent des problèmes de coût et de défaillance des noeuds.

Le tableau 3.1 résume les principales différences entre ces architectures.

Architecture	**Type**	**Avantage**	**Inconvénient**
Henning Shulzrinne	Hybride et non structurée	Utilisation des supers noeuds qui s'occupent de la recherche comme un intermédiaire	Arrivé ou départ fréquent des supers noeuds
Sosimple	Décentralisée et structurée	La localisation se fait entre les noeuds sans passer par un serveur central	Le coût en termes de nombre de messages à échanger entre le noeud initiateur et les autres noeuds
Skype	Hybride et non structurée	Authentification auprés d'un serveur de login	La panne du serveur de login perturbe le fonctionnement de Skype

TABLE 3.1 – Comparaison entre architectures de P2P-SIP.

3.4.3 Problèmes de P2P-SIP

Parmi les principaux problèmes des réseaux P2P-SIP on cite :

3.4.3.1 Tolérance aux pannes

Dans les réseaux P2P-SIP, les mécanismes de détection des pannes ne font pas la différence entre les défaillances physiques et temporelles. Si un super nœud n'envoie pas le message d'acquittement à ses noeuds ordinaires, il sera considéré comme étant en panne. En cas de panne de son super nœud d'attache, un nœud ordinaire ne peut ni passer ni recevoir un appel **[39]**.

3.4.3.2 Surcharge des supers noeuds

La surcharge des supers pairs est plus petite que dans d'autres pairs où les supers pairs doivent stocker tous les pairs qui dépendent d'eux. Ce qui implique l'entretien d'une plus grande quantité de l'information **[41]**.

3.4.3.3 Sécurité

Dans une architecture P2P-SIP, les messages peuvent être transmis par un ou plusieurs noeuds avant d'atteindre sa destination, tout noeud malicieux au sein du réseau peut déposer, modifier ou transférer un mauvais message **[43]**.

Les noeuds utilisant le service d'enregistrement pour la localisation, peuvent ne pas laisser passer des messages. Une grande quantité de tels noeuds, finis par aboutir à une indisponibilité de la fonction de l'emplacement **[42]**.

La sensibilité d'un système de réputation aux attaques dépend du coût auquel les noeuds peuvent être générés, du degré auquel le système de réputation accepte les messages provenant de noeuds qui n'ont pas un chemin de confiance lié à un autre noeud de confiance **[42]**.

3.5 Conclusion

Nous avons présenté dans ce chapitre la manière d'utilisation de la téléphonie Internet en mode P2P en utilisant le protocole de signalisation SIP, qui rassemble les avantages des architectures P2P et ceux du protocole SIP.

Une des principales difficultés soulevés par les applications de la téléphonie Internet est la prise en compte de la priorité de routage dans la recherche et la localisation.

D'après l'étude menée sur les architectures de P2P-SIP, nous n'avons retenue que l'architecture décentralisée, permettant d'optimiser la recherche et la localisation des utilisateurs sans prendre en compte la priorité.

Dans le chapitre suivant, nous allons proposer une solution pour la recherche et la localisation des utilisateurs dans une architecture P2P, basée sur le routage prioritaire.

Chapitre 4

Routage prioritaire en mode P2P-SIP

4.1 Introduction

Les applications de la téléphonie IP en temps réel, ont des besoins en termes de Qualité de Service *(QoS)*. Ces besoins peuvent être exprimés en termes de délai de transmission de paquets, qui représente la mesure de temps où la voix de l'émetteur arrive au destinataire, de gigue, qui représente la variation de délai de transmission, de perte de paquets à cause de la congestion et la bande passante minimale nécessaire pour la téléphonie.

Dans ce chapitre on propose une solution générique concernant la recherche et la localisation des utilisateurs dans n'importe quelle architecture P2P décentralisée non structurée, basée sur la notion de priorité pour les appels téléphoniques d'urgence (*police,pompiers...*).

4.2 Problématique

Une des principales difficultés soulevées pour les applications de la téléphonie Internet, est la prise en compte les paramètres de la qualité de service. Le temps de transmission est une difficulté principale pour la transmission des appels téléphoniques d'urgence sur le réseau.

La diffusion de message à tous les voisins pour la localisation d'un utilisateur dans un réseau P2P, ne minimise pas le temps de transmission de l'appel d'urgence, c'est l'un des problèmes qui attire l'attention de plusieurs chercheurs ces dernières années.

4.3 Proposition d'une solution

Dans l'architecture décentralisée non structurée Gnutella, le processus de la localisation d'un utilisateur est semblable au processus d'effectuer un appel d'urgence dans le réseau P2P-SIP.

Lorsqu'un nouveau nœud se connecte sur le réseau, il commence par rechercher tous les nœud du réseau, il envoie une trame d'identification *(PING)* à tous ces voisins, qui la enchainent vers tous les autres noeuds aux quels ils sont connectés, et ainsi de suite jusqu'à le nœud 7. A chaque noeud du réseau le TTL *(Time To Live)* qui représente le nombre de retransmissions est décrémenté, lorsqu'il s'annule, la retransmission est stoppée.

Pour éviter les boucles dans la transmission, la trame reçue est stockée pendant un court laps de temps. Lorsqu'un pair est identifié, il envoie à l'émetteur une trame de réponse *(PONG)*.

Notre proposition est générique pour n'importe quelle architecture P2P décentralisée non structurée. Consiste à optimiser les appels téléphoniques d'urgence en minimisant un seul paramètre de QoS (*le temps*).

On utilise Gnutella pour modéliser le réseau de files d'attentes *(ensemble de files interconnectées)*.

4.3.1 Principe de fonctionnement

Notre réseau est modélisé par un graphe G défini comme le couple d'ensembles *(N, F)*, où N : ensembles des nœuds et F : ensemble des files.

Chaque nœud contient une file d'attente F contenant des requêtes triées par priorité pour déterminer la requête la plus prioritaire à acheminer.

La figure 4.1 représente la modélisation de réseau de files d'attentes en utilisant Gnutella. Le problème consiste à effectuer un appel d'urgence dans un plus court temps entre un nœud source *(S)* et un nœud destinataire *(D)*.

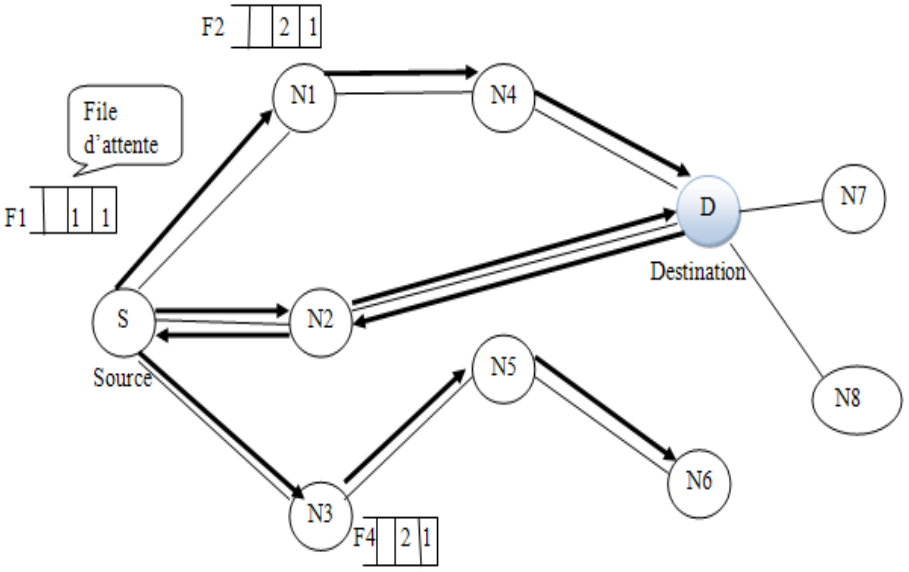


FIGURE 4.1 – Modélisation de réseau.

Nous nous intéressons aux :

- Caractéristiques de la file d'attente de chaque nœud.

- Processus d'arrivé des requêtes.

- Routage prioritaire des requêtes entre les nœuds.

4.3.2 Caractéristique de la file d'attente

Une file d'attente est constituée d'un ou plusieurs serveurs et d'un tampon *(buffer)* , où les requêtes attendent d'être servies. Elle peut être caractérisée par six paramètres, ce qui donne la notation de Kendall :

A/B/S/K/C/Z

- **A** : désigne la distribution des temps entres deux arrivées. Dans Gnutella, ces arrivées sont des requêtes arrivant à un noeud spécifié.

- **B** : désigne la distribution des durées de service. Dans Gnutella, le service est le canal de sorti.

- **S** : c'est le nombre de serveurs.

- **K** : spécifie la capacité de la file, c'est à dire le nombre maximum de requêtes susceptibles d'être stockées dans la file.

- **C** : c'est la taille de la population des requêtes.

- **Z** : désigne la discipline de service, c'est-à-dire l'ordre dans lequel les requêtes arrivant sont rangées, puis sortis de la file.

Dans la modélisation précédente du réseau, on a utilisé M/M/1 qui représente une file à un serveur avec une distribution exponentielle entre les temps des arrivées et les durées de service, dans laquelle la distribution de service est FIFO, avec une capacité et une population infinie. Une telle file peut être schématisée par la figure 4.2.

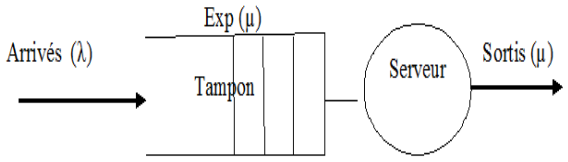


FIGURE 4.2 – File d'attente M/M/1.

x : la variable aléatoire entre deux requêtes successives.

$$\mathbf{F\ (x)\ =\ 1\text{-}\ exp\ (\text{-}}\lambda x), x \geq 0.$$

4.3.3 Processus d'arrivé et servir des requêtes

Les requêtes arrivant à la file avec un taux moyen d'arrivé (λ) se classent selon leur priorité *(la requête plus prioritaire se place dans le sommet de la file)*. Elles se servent selon un taux moyen de service (μ).

Si deux requêtes ont la même priorité, elles sont servi dans l'ordre FIFO (*le premier arrivé est le premier sorti*).

Si deux requêtes sont différentes, elles sont servi selon un service de priorité PR (*service according to PRiority*).

- **Ta** : le temps moyen entre deux arrivées consécutives.

- **Ts** : le temps moyen de service.

- $\lambda = \mathbf{1}/\ \mathbf{Ta}$: le taux moyen d'arrivé des requêtes (*req/sec*).

- $\mu = \mathbf{1}/\ \mathbf{Ts}$: le taux moyen de service (*req/sec*).

- $\rho = \lambda/\ \mu$: la charge du réseau. Pour que le réseau soit stable, il faut que ,ρ<1 ($\lambda < \mu$).

- **P(R)** : est la probabilité qu'il y'ait R requêtes arrivant dans le réseau.

$$\mathbf{P\ (R)} = \rho^R * (1-\rho).$$

- **R** : le nombre moyen de requêtes dans la file :

$$\mathbf{E\ (R)} = \rho/\ \mathbf{(1-}\ \rho).$$

- **L** : la longueur de la file :

$$\mathbf{E\ (L)} = \rho^2/(1-\rho).$$

- **W** : le temps d'attente d'une requête d'être transmise dans la file :

$$\mathbf{E\ (W)} = \rho/\ (\mu * (1-\rho)).$$

- **T** : le temps de réponse d'une requête .

$$\mathbf{E\ (T)} = \mathbf{1}/\ (\mu * (1-\rho)).$$

- **Simulation M/M/1**

Le principe de l'algorithme de simulation suivant est de déterminer la durée de service d'un nombre de requêtes qui arrivent au moment que les autres servent.

H : horloge initiale.

Hp : horloge précédente.

D : nombre de requêtes sortantes

xi : est la durée qui sépare entre deux arrivés.

X : est la fonction de cumulation = $\sum xi$.

R : nombre de requêtes.

B : temps de blocage.

t1 : temps d'arrivé.

t2 : temps de service.

ts : temps de simulation.

tb : est l'instant d'activer le système.

Algorithme de simulation de M/M/1

```
H=0;  Hp=0; D=0; X=0; R=0; B=0; t1=0; t2=ts;
Début
    Tanque (H<ts) faire
        Si (t1<=t2) alors
            H=t1;
            X=X + ((H- Hp)*R);
            R=R+1;
            Hp = H;
            t1=H-(log (rand)/lambda);    // génération de temps d'arrivé.
                Si (R==1) alors
                    t2=H-(log (rand)/mu);    // génération de temps de service.

                    tb =H;
                Finsi ;

        Sinon
            H=t2;
            X=X + ((H- Hp)*R);
            R=R-1; D=D+1;
            Hp=H;
                Si  (R==0) alors
                    t2=ts;
                    B=B+ (H- tb);
                Sinon
                    t2=H-(log (rand)/mu);
                Finsi;
        Finsi ;
    Fintanque ;
Fin
```

4.3.4 Routage prioritaire de requêtes entre les noeuds

A. Algorithme de routage dans Gnutella

L'algorithme suivant nous donne l'acheminement des requêtes de la source à la destination dans le réseau Gnutella sans prendre en compte leur priorité.

Algorithme de routage :

```
Début
    Si (nœud courant = source) et {voisins} ≠ Φ alors
        Lancer la requête à tous les voisins ;
    Sinon
            A la réception de la requête
        Si (TTL ≠ 0) alors
            Si (nœud courant = Destination) alors
                Envoyer un acquittement a la source
            Sinon
                Lancer a requête à tous les voisins.
            Finsi ;
        Sinon
            Stopper la recherche ;
        Finsi ;
    Finsi ;
Fin ;
```

B. **Algorithme de routage avec priorité dans Gnutella**

L'algorithme suivant nous donne l'acheminement des requêtes de la source à la destination dans le réseau Gnutella en prennant en compte leur priorité.

Algorithme de routage avec priorité

```
Début
  Si (nœud courant = source) et {voisins} ≠ ϕ alors
     Lancer la requête prioritaire à tous les voisins ;
     Tanque (reçu =vrai) faire
        Enfiler la requête prioritaire dans la file
     Fintanque
  Sinon
     Si (nœud courant ≠ destinataire) alors
        Si (file ≠ ϕ) et (TTL <> 0) alors
           Défiler la requête prioritaire;
           Lancer la requête prioritaire à tous les voisins ;
        Finsi
     Sinon
        Si (reçu = vrai) alors
        Enfiler la requête prioritaire dans le file de destinataire ;
        Envoyer ACK à la source ;
     Finsi ;
  Finsi ;
Fin ;
```

C. **Exemple d'implémentation**

Soit G (7, 7) un graphe de 7 nœuds numérotés (S, 1, 2, 3, 4, 5, D) et 7 files numérotées de F1 à F7.

Soit S le noeud source cherchant le noeud destinataire D. Le déroulement de l'algorithme précédent sur cet exemple est donné comme suit :

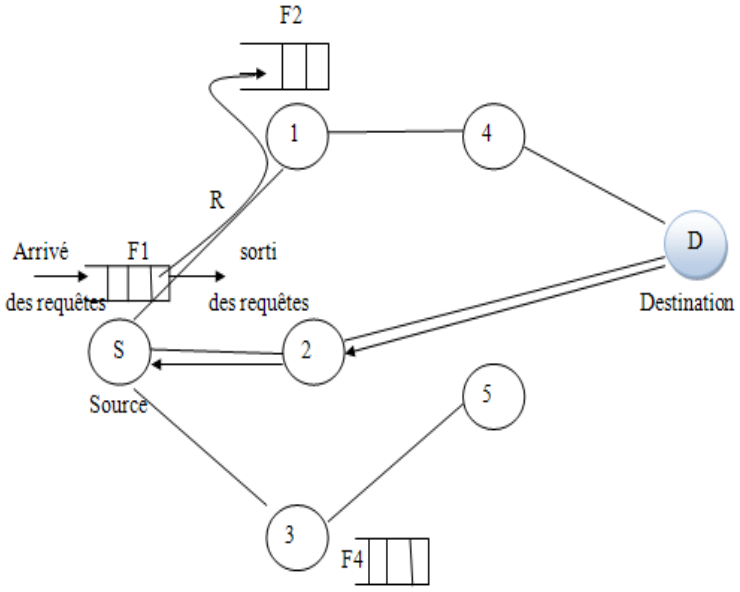


FIGURE 4.3 – Exemple d'implémentation.

- Nœud courant = Source.

TTL = 3.
Voisins = 1, 2, 3.
Le noeud S lance la requête à 1, 2, 3.

- Nœud courant =1.

Le noeud 1 reçoit la requête de S.
TTL = 2 $\neq$0.
Le noeud 1 $\neq D$.
$Voisins = 4$.
Le noeud 1 lance la requête à 4.

- Nœud courant = 2

Le noeud 2 reçoit la requête de S.
TTL = 2$\neq$ 0.
Le noeud 2 $\neq$ D.
Voisins = D.
Le noeud 2 lance la requête à D.

- Nœud courant = 3.

Le noeud 3 reçoit la requête de S.
TTL = 2 ≠ 0.
Le noeud 3 ≠ D.
Voisins = 5.
Le noeud 3 lance la requête à 5.

- Nœud courant = 4

Le noeud 4 reçoit la requête de 1.
TTL = 1 ≠ 0.
Le noeud 4 ≠ D.
Voisins = D.
Le noeud 4 lance la requête à D.

- Nœud courant = D.

Le noeud D reçoit la requête de 2.
TTL = 1≠0.
Le noeud D = D.
Le noeud D envoie un acquittement à la source.

- Nœud courant = 5.

Le noeud 5 reçoit la requête de 3.
TTL = 1≠ 0.
Le noeud 5 ≠ D.
$Voisins = \phi$.

- Nœud courant = D.

Le noeud D reçoit la requête de 4.
TTL = 0.
Le noeud D est un noeud identifié.
Stopper la recherche.

4.4 Conclusion

Dans ce chapitre, nous avons proposé une solution basée sur le routage prioritaire en mode P2P-SIP. Nous avons utilisé l'architecture P2P décentralisé et non structuré Gnutella avec des files d'attentes.

On a utilisé un algorithme de simulation de la file d'attente M/M/1 pour optimiser le temps total de service des appels d'urgences, et un algorithme de routage prioritaire pour optimiser la recherche et la localisation des utilisateurs en minimiant le temps. Avec cet algorithme, nous pouvons nous arrêter dès qu'on trouve le noeud recherché (*le destinataire*).

Dans le chapitre suivant, nous allons présenter l'évaluation de performances de la solusion proposée.

Chapitre 5

Évaluation de performances

5.1 Introduction

Aprés avoir détaillé le fonctionnement de notre solution, nous présentons dans ce chapitre les principales étapes de réalisation de notre simulateur, dont l'objectif est de permettre d'évaluer les performances et de montrer l'efficacité de la solution proposée.

Nous commençons par la présentation des techniques d'évaluation, l'environnement de simulation et les paramèttres que nous allons prendre en considération lors de l'évaluation des performances de routage concernant les appels téléphoniques d'urgence.

5.2 Techniques d'évaluation

Les techniques d'évaluation de performances se résument en trois axes :

5.2.1 Analytique

Il s'agit de réduire le système en un modèle mathématique et l'analyser numériquement. L'approche analytique est parfois rapide à réaliser, mais présente le souci de la représentation fidèle du système.

Il est parfois très complexe de modéliser le comportement réel du système mathématiquement. Généralement, on pose des hypothèses qui simplifient l'étape de modélisation du système et rendent l'évaluation numérique faisable. Ces hypothèses simplificatrices peuvent toucher la fidélité de représentation du système, mais permettent toutefois de traduire son

comportement approché.

Il existe de nombreux outils mathématiques permettant de telle évaluation, les automates, les réseaux de pétri,les réseaux de files d'attentes, les approches probabilistes, les approches déterministes, etc.

5.2.2 Mesure

Il s'agit de faire des mesures et les analyser directement sur un système réel. Cette technique permet de comprendre le vrai comportement du système, mais faire des mesures sur des systèmes réels n'est pas toujours possible, car ça pourrait gêner le fonctionnement du système ou aussi pour des problèmes de coûts (*système non encore existant, instrument de mesure complexe, etc*).

Les résultats de la mesure ne sont pas génériques et ne reflètent qu'une trajectoire du système.

5.2.3 Simulation

La simulation est la méthode d'évaluation de performances la plus prédominante dans le domaine des réseaux . Elle est largement utilisée pour évaluer les nouvelles architectures et les protocoles de communication, car elle permet de tester à moindre coût ces nouveaux protocoles et d'anticiper les problèmes qui pourront surgir durant leur implémentation réelle.

Pour le faire, elle construit un modèle du système réel en représentant toutes ses entités, leur comportement et leur interaction pour mener en suite, des expériences sur ce modèle avec une simple modification des paramètres de simulation, dont les résultats seront facilement analysables et interprétables.

5.3 Choix du MATLAB

MATLAB (*abréviation de "Matrix Laboratory"*) est un langage simple et très efficace, optimisé pour le traitement des matrices et le calcul numérique. Il est beaucoup plus concis que les "vieux" langages (*C,Pascal, Fortran, Basic*). On peut traiter la matrice comme une simple variable.

MATLAB contient une interface graphique puissante, ainsi qu'une grande variété d'algorithmes scientifiques.

MATLAB contient également un langage de programmation de haut niveau dans lequel on retrouve la majorité des concepts des langages de programmation modernes. L'ordre d'exécution des instructions est déterminé par des structures de contrôle. Il permet aussi la création de fonctions et distingue les données locales et globales.

Ces avantages ont rendus de MATLAB, un langage de programmation et de simulation très utilisé.

5.4 Paramèttres de simulation

Le tableau 5.1 contient les paramètres du réseau sur lequel les simulations ont été effectuées.

Définition de la constante	**Valeur Initiale**	**Type**	**Unité de mesure**
Nombre de noeuds	7	Nombre	Nombre
Position de la source(x, y)	(0,0)	Entier	
Taille de tableau de requêtes	(2,2)	Entier	
Temps de simulation	6000	Entier	
Taille du réseau	100*100	Surface	M*M
Taux d'arrivé λ	0.5	Réel	Arrivé /seconde
Taux de service μ	1	Réel	Arrivé /seconde

TABLE 5.1 – Paramèttres de simulation.

5.5 Étapes de réalisation du simulateur

Les étapes de réalisation de notre simulateur sont les suivantes :

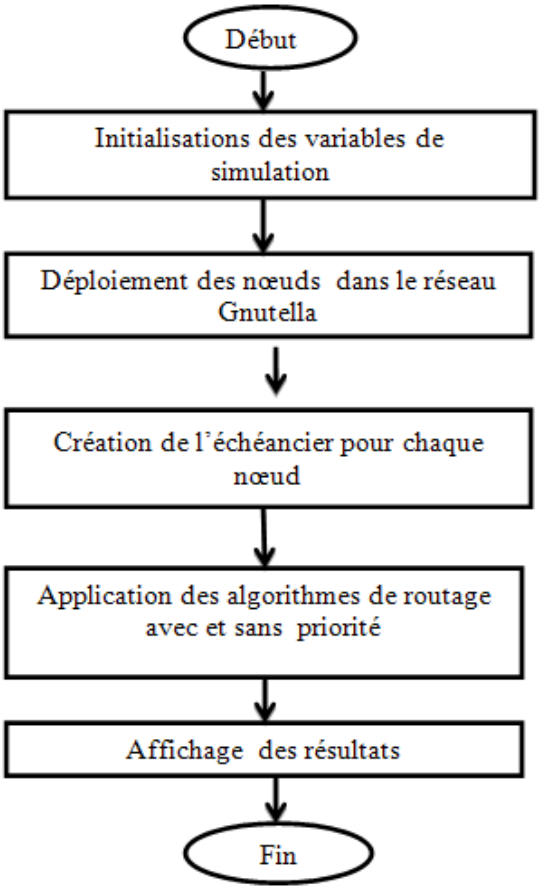


FIGURE 5.1 – Principales fonctions du simulateur.

5.5.1 Initialisation des variables du simulation

Cette étape est exécutée automatiquement au début de programme de simulation, elle inclut la déclaration des variables globales (*nombre de noeuds, zone de déploiement simulée, temps de simulation,...*) et leur initialisation, ainsi que la création des noeuds sous forme d'une structure qui comporte (*l'identité des noeuds, leurs coordonnés, leur caractéristiques*) et leur dispersion aléatoire sur la zone de déploiement.

5.5.2 Déploiement des noeuds du réseau

Les noeuds constituant notre réseau, sont déployés d'une façon aléatoire sur une surface de (100*100)m2 avec des liens de communication (voir la figure 5.2) . Chaque noeud est représenté par ses coordonnées (x,y).

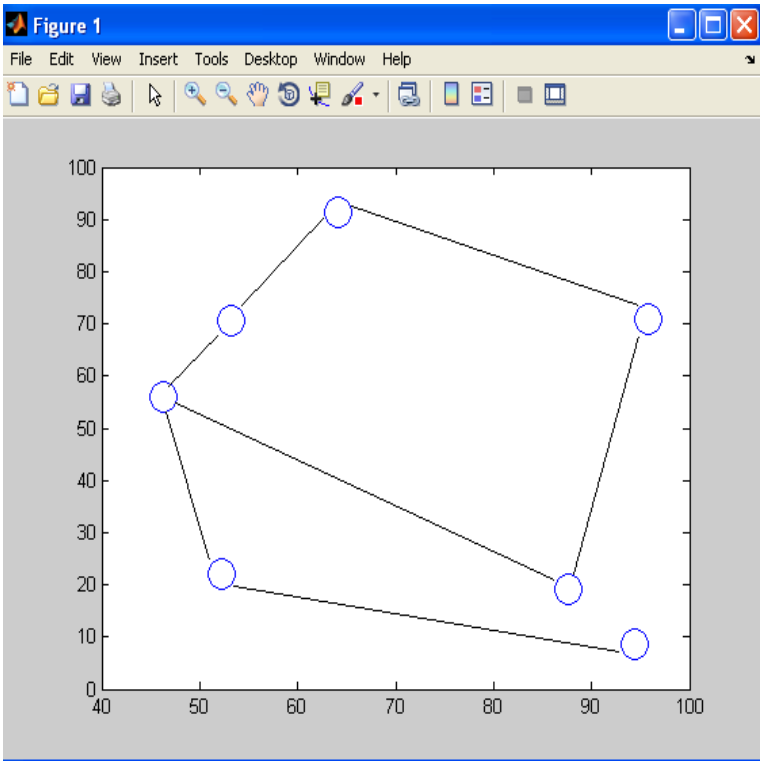


FIGURE 5.2 – Déploiement du réseau.

5.5.3 Création de l'échéancier

Les requêtes acheminées sur le réseau sont rangées dans un échéancier de requêtes. Ce dernier est représenté par (*l'identité du noeud, identité de la requête prioritaire, identité de la requête non prioritaire, le temps d'arrivé de la requête prioritaire et le temps d'arrivé de la requête non prioritaire*).

L'arrivée des requêtes dans la file d'attente suivent une loi de poisson de paramètre λ, ils sont servis suivant une loi exponentielle de paramètre μ (*ta = ta -log(uniforme())=lambda)*).

Le tableau 5.2 représente un exemple d'échéancier d'un noeud.

Noeud 1			
Identité de la requête prioritaire	1	1	0
Identité de la requête non prioritaire	2	0	2
Temps d'arrivé de la requête prioritaire	0.03	0.4	0
Temps d'arrivé de la requête non prioritaire	0.067	0	0.01

TABLE 5.2 – Exemple d'échéancier d'un noeud.

5.5.4 Application de l'algorithme de routage

• Avec priorité : dans cet algorithme, nous avons ordonné les requêtes selon leur priorité (*1 pour les requêtes prioritaire et 2 pour les requêtes non prioritaire*). S'il existe des requêtes prioritaires dans la file d'attente, elles seront servis avant celles qui sont non prioritaire.

• Sans priorité : dans cet algorithme, un noeud envoie les requêtes (*l'appel*) selon l'ordre de leur arrivé, sans prendre en considération la priorité (*le premier arrivé le premier servi*).

5.5.5 Affichage des résultats

Les résultats de la phase application de l'algorithme de routage seront utilisés pour tracer des courbes pour faire une comparaison entre le routage prioritaire et non prioritaire selon des métriques de performances choisisses.

5.6 Métriques de performance

Les deux métriques essentielles lors de l'étude d'un routage dans les réseaux P2P sont :

Le temps de réponse : il est définit à l'instant où les requêtes enfilées dans l'échéancier jusqu' à ce que la réponse soit reçue par le demandeur.

Le nombre de requêtes : est le nombre de requêtes envoyées par chaque noeud.

5.7 Résultats de la simulation

La figure 5.3 présente les résultats de simulation obtenus en appliquant le routage sans priorité et avec priorité suivant les deux métriques indiquées précédement (*le temps de réponse et le nombre de requêtes envoyées par chaque noeud*).

Nous remarquons que le temps de réponse pour acheminer les requêtes de la source au destinataire en utilisant l'algorithme de routage sans priorité est plus élevé par rapport à celui du routage avec priorité qui est très petit.

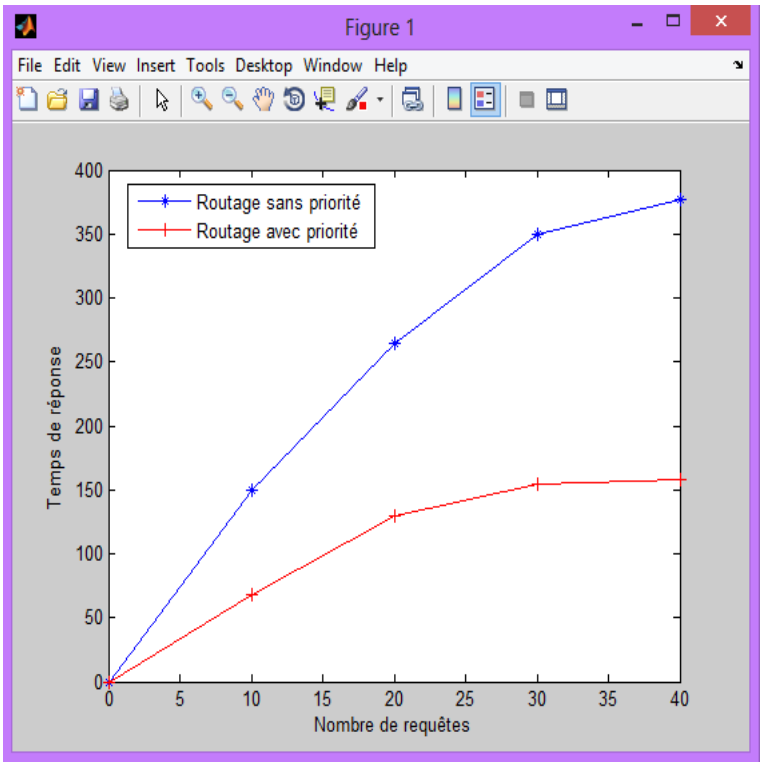


FIGURE 5.3 – Temps de réponse en fonction de nombre de requêtes envoyées.

Dans ce qui suit, nous allons exposer et analyser les résultats obtenus, et de faire une comparaison de performances de routage sans priorité et avec priorité.

Le tableau 5.3 représente la comparaison entre les temps de réponse en appliquant l'algorithme de routage avec priorité et celui sans priorité.

Nombre de requêtes	**Temps de réponse dans le routage avec priorité(ms)**	**Temps de réponse dans le routage sans priorité(ms)**
0	0	0
5	25	75
10	70	150
15	100	210
20	140	260
25	149	310
30	150	350
35	157	355
40	158	380

TABLE 5.3 – Comparaison entre les temps de réponse des deux routages.

Nous terminons par l'analyse des résultats obtenus, ces derniers montrerent l'importance de routage avec priorité par rapport au routage sans piorité en termes de temps de réponse et du nombre de requêes envoyées.

5.8 Conclusion

Dans ce chapitre, nous avons présenté une description générale de l'évaluation de performances de notre solution et les résultats de la simulation obtenus par l'application du routage avec et sans priorité.

D'aprés les résultats de la simulation, nous avons prouvé que le routage avec priorité est plus efficace par rapport à celui sans priorité.

Conclusion générale et Perspectives

LA téléphonie IP prennent des dimensions de plus en plus importantes depuis quelques années. La téléphonie entre pairs via l'Internet commence à prendre une part importante dans le mode des télécommunications.

Nous avons présenté dans ce mémoire plusieurs notions sur les réseaux P2P, qui sont une solution avantageuse pour le partage et le transfert de contenu. Avec leur architecture déccentralisée, le routage P2P est au niveau applicatif dont l'objectif est de diminuer au maximum possible le temps des appels téléphoniques d'urgence.

À l'heure actuel, SIP se présente comme le protocole de signalisation le plus adéquat aux applications de la téléphonie IP à cause de sa simplicité.

Dans le contexte de notre travail, nous avons introduit les concepts de la téléphonie IP basée sur le P2P-SIP.

Notre solution est basée sur une architecture décentralisée non structurée, ainsi sur le protocole de signalisation SIP. Nous avons utilisé Gnutella pour optimiser la recherche et la localisation des utilisateurs , qui prend en considération les paramètres de la QoS en terme de temps pour les appels téléphoniques d'urgence.

En perspectives, nous allons amélioré notre solution pour minimiser le nombre de sauts d'une requete afin d'arriver au destinataire.

Une autre perspective, est de pouvoir sécuriser les appels téléphoniques d'urgence dans les réseaux P2P.

Bibliographie

[1] *Peer to peer : partager et repartir traitements et données, extrait du magazine hebdomadaire : Décision informatique* http ://www.01netcom/article/189227.html.

[2] Clarke Siobha, *Dependability in Peer to peer system,* IEEE internet Computing, 2004.

[3] Nathalie BUDAN, Benoit TEDESCHI, Stéphane VAUBOURG, *nouvelles technologies réseau, les réseaux peer to peer Fonctionnement, exemples, limites,* Mémoire d'Ingénieur en Informatique, 2000.

[4] Houda NAFI, *protocole pour la sécurité des réseaux sans fil Peer to peer, technologie de l'information et de communication,* Mémoire de Magistère en Informatique, Université de Ouargla.

[5] Patrick MARLIER, *Sécurité du peer to peer ,* www.labo-asso.com.

[6] Fabrice Schuler, *Etude et utilisation des technologies des P2P,* Avril 2005.

[7] Fouzia BOUDRIES, Samia EDJDOUB, *Conception et réalisation d'une application peer to peer de Messagerie Instantanée,* Mémoire d'Ingénieur en Informatique, Université de Bejaia, 2008.

[8] Marc Bourreau, *Le peer to peer et la crise de l'industrie du disque : une perspective historique,* ENST, Departement EGSH, et CREST-LEI,Université Paris Dauphine http ://www.freescape.eu.org/biblio/IMG/pdf/music1.pdf.

[9] Gabrielle Feltin, Guillaume Doyen, Olivier Festor, *les protocoles peer to peer, leur utilisation et leur détection,* Octobre 2003 http ://2003.jres.org/actes/paper.70.pdf.

[10] Mourad AMAD, *Découverte et localisation de service en mode P2P,* Mémoire de Magistère en Informatique, Université de Bejaia, 2005.

[11] David Clark, *Face-to-Face with Peer-to-Peer Networking,* IEEE internet computing 2001.

[12] *Le protocole SIP,* RFC 2543, http ://www-igm.univ-mlv.fr/ dr/XPOSE2002/DEBOURDEAU.

[13] Patrice Kadionik, *Le projet HomeSIP : la domotique avec le protocole SIP,* http ://www.unixgarden.com/index.php/gnu-linux-magazine-hs/le-projet-homesip-la-domotique-avec-le-protocole-sip, 2006.

[14] *Le Protocole SIP Avancé et ses Extensions,* http ://www.efort.com.

[15] A .Aoun, *Le protocole SIP,* La visioconférence SIP, 2007.

[16] Henrik Ingo, *Session Initiation Protocol (SIP) and other Voice over IP (VoIP) protocols and applications,* http ://open-life.cc/system/files/FSWC+Henrik+Ingo+Article+SIP,+VoIP+and+FLOSS.pdf.

[17] http ://www.numidit.dz/index.php option=com contentview=articleid=843Avoix-sur-ip-voipcatid=423Avoiplimitstart=3.

[18] Kundan Singh, Henning Schulzrinne, *Peer-to-Peer Internet Telephony using SIP,* , IEEE, Columbia University, 2004.

[19] Kundan Narendra Singh, *Reliable, Scalable and interoperable Internet Telephony,* Thèse de doctorat, Université de Colombia, 2006.

[20] Thanh Son, D.M.A Walsh, J.A. Cooper, G. Brodin, *A SIP based approach for inter-network operations of wireless positioning systems,* http ://ieeex-plore.ieee.org.www.sndl1.arn.dz/ielx5/10989/34628/01651707.pdf?tp=arnumber=1651707isnumb

[21] Henning Schulzrinne, Jonathan Rosenberg, *Signaling for internet telephony,* IEEE internet computing, 2000.

[22] Sébastien FONTAINE, *Entête SIP,* http ://www.architoip.com/entete-sip.

[23] Zoubida BOUKHERROU, *Optimisation du routage pour la VoIP : Application à signalisation SIP,* Mémoire de Magistere en Informatique, Université de Bejaia, 2006.

[24] Mohamed El Mahdi BOUMEZZOUGH, *Etude et mise en oeuvre du service pilote ToIP de RENATER,* Mémoire d'Ingénieur en Informatique, 2009.

[25] Wenyu Giang, Jonathan lennox, Sankaran Narayanan, and Henning Schulzrinne, *Integrating Internet telephony services,* IEEE Internet computing, 2002.

[26] Franck Salque, Xavier Bruns, *la téléphonie sur IP : qui fait quoi, pour qui et comment ?,* 2004.

[27] Mark Handley, Richard Karp, Sylvia Ratnasamy, Paul Francis, *A Scalable Content-Adressable Network,* ACM SIGCOMM'01, San Diego, 2001.

[28] Nilanjan Banerjee, Arup Acharya, Sajal, Das, *Enabling SIP-based sessions in adhoc networks,* wireless networks, 2007.

[29] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, *SIP : Protocole d'initialisation de session,* Copyright (C) The Internet Society, 2002.

[30] Alan B. Johnston, *SIP : Understanding the Session Initiation Protocol,* Second edition, Artech House, Boston London. ISBN 1-58053-655-7,2004.

[31] J. Rosenberg, H. Schulzrinne, *Session initiation protocol (SIP) : locating SIP servers,* RFC3263, Internet Engineering Task Force, 2002.

[32] Pierre Barraquet, Erwann Ben Souiden, Geoffroy Colin, Mathieu Guillon, Mathieu Osty, Chloé Rolland, *Skype,* , IEEE, 2004.

[33] ABEDLLI Nabila, AIT OUALI Kayssa, *Découverte et Localisation des Usagers dans un réseau Peer to Peer,* Mémoire d'Ingénieur en Informatique, Université de Bejaia, 2009.

[34] Michael Gough, *Skype Me! From Single User to Small Entreprise and Beyond, édition Syngress Publishing,* 2006.

[35] Salman A. Baset, Henning G. Schulzrinne, *An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol,* Department of Computer Science, Columbia University, 2004.

[36] David A. Bryan, Bruce B Lowekamp, *SOSIMPLE : A SIP/SIMPLE Based P2PVoIP and IM System,* Computer Science Department, Williamsburg VA 23185.

[37] David A. Bryan, Bruce B. Lowekamp, Cullen Jennings, *SOSIMPLE : A Serverless, Standards-based, P2P SIP Communication System,* IEEE, AAA-IDEA, 2005.

[38] Laurent Ouakil, Guy Pujolle, *Téléphonie sur IP : SIP, H.323, MGCP, Qos et sécurité, Asterisk, VoWIFI, offre multiplay des FAI, Skype et autres softphones, architecture de IMS,* livre, 2eme édition.

[39] Ibrahima DIANE, Ibrahima NIANG, *Schéma DHT hiérarchique pour la tolérance aux pannes dans les réseaux P2P-SIP,*,IEEE, Université Cheikh AntaDiop de Dakar,BP 5005 - Dakar-Fann.

[40] Diego Suarez, José M.Sierra, Antonio Izquierdo, Henning Schulzrinne, *Survey of Attacks and Defenses on P2PSIP Communications,* IEEE COMMUNICATIONS SURVEYS TUTORIALS, VOL. 14, NO. 3, THIRD QUARTER, 2012.

[41] Isaias Martinez-Yelmoa, Alex Bikfalvib , Ruben Cuevasa ,Carmen Guerreroa , Jaime Garciaa, *H-P2PSIP : Interconnection of P2PSIP domains for Global Multimedia Services based on a Hierarchical DHT Overlay Network, IEEE.*

[42] Warodom WERAPUN, *Architectures de réseaux pour la délivrance de services à domicile,* Thèse de Doctorat en Informatique, Université de Toulouse, Sep 2012.

[43] J. Seedorf, *Security Challenges for P2P-SIP,* IEEE Network, Special Issue on Securing Voice over IP, Sep 2006.