République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE d'ORAN Mohamed Boudiaf

Faculté des Sciences

Département d'informatique

Spécialité : Informatique ***Option : Modélisation Optimisation et Evaluation des performances des Systèmes (MOEPS)***

MEMOIRE

Présenté par

Mr. DOUARA Bachir Naceureddine

Pour l'obtention du diplôme de Magister en informatique

Thème

Sécurité du protocole SIP dans les réseaux Ad-hoc

Soutenu le .....devant la commission d'examen composée de :

Qualité	**Nom et Prénoms**	**Grade**	**Etb d'origine**
Président	Mr. Mohamed BENYETTOU	Professeur	USTO-MB
Rapporteur	Mr. Khaled BELKADI	Maître de conférences-A	USTO-MB
Examinateur	Mr. Kaddour SADOUNI	Maître de conférences-A	USTO-MB
Examinatrice	Mme. Samira CHOURAQUI	Maître de conférences-A	USTO-MB
Invité	Mr. MAMMAR Soulimane	Maître Assistant-A	ENSET-Oran

Année universitaire : 2011/2012

Remerciements

Je remercie ‘’DIEU’’ tout puissant, de m’avoir donné la force, le courage et la patience de mener à terme ce travail.

Mes remerciements les plus chaleureux vont principalement à mon encadreur Monsieur **BELKADI Khaled** maître de conférences-A à l'USTOMB qui m'a orienté, guidé et soutenu sans hésitation ni relâche dans le choix et l'accomplissement de ce travail.

Je tiens également à remercier Monsieur **MAMMAR Soulimane** pour l’aide, pour l’intéressantes documentations qu’il a mis à ma disposition, son suivi, et ces discussions qui m’ont aidé à résoudre les problèmes rencontrés pendant la réalisation de ce mémoire.

J’adresse mes remerciements les plus sincères à monsieur **BENYETTOU Mohamed**, professeur à USTO-MB et directeur du laboratoire de recherche «**LAMOSI** », qui me fait l’honneur d’accepter de présider, d’évaluer et de juger ce modeste travail.

J’exprime ma gratitude envers monsieur **Kaddour SADOUNI** maître de conférences-A, d’avoir accepté d’examiner mon travail. Mes remerciements s’adressent également à Madame **CHOURAQUI Samira** maître de conférences-A, pour avoir accepté d’être membre de mon jury, et de juger ce travail.

Je remercie tous mes professeurs chacun par son nom pour leur patience et leur abnégation tout au long de mes études.

Enfin je remercie tout le cadre pédagogique et administratif qui sans eux ma mission ne se serait pas accomplie.

Dédicaces

Si je suis arrivé là c'est grâce à Dieu.

J'adresse un grand salut à mes parents, à ma famille et à toutes les personnes qui m'ont soutenu.

Résumé

Les réseaux ad-hoc sont des réseaux dynamiques, caractérisés par l'absence d'infrastructures, de ressources limitées, et moins de sécurité. L'adaptation des protocoles Internet avec ces réseaux est un défi. Le protocole SIP (Session Initiation Protocol) utilise un ensemble de serveurs pour créer, gérer, et terminer les sessions multimédia. Récemment de nombreuses solutions ont été proposées pour adapter SIP dans les environnements ad-hoc. LCA (Loosely Coupled approach) est l'une de ces solutions qui utilise la même technique que le protocole AODV (Ad-Hoc On-Demand Distance Vector). L'aspect sécuritaire dans LCA a été ignoré. En effet, il est vulnérable aux attaques affectant le processus d'établissement de session. Le but de ce mémoire est de sécuriser LCA, en utilisant le schéma de signature basée sur l'identité qui est sans certificat. Dans le schéma proposé (Secure_LCA), nous utilisons l'adresse URI SIP comme une clé publique qui peut aboutir à des clés de petites tailles par rapport à d'autres techniques cryptographiques ; également il réduit l'overhead et le coût de communication et de stockage.

Mots- clés : Ad hoc, signature basée sur l'identité, LCA, SIP, Sécurité.

Abstract

Ad-hoc networks are dynamic networks, characterized by the absence of infrastructure, limited resources and less of security, the adapting of internet protocols with these networks is a challenge, the session initiation protocol (SIP) use a set of servers to create, manage, and tear down multimedia session. Recently many solutions have been proposed to adapt SIP in ad hoc environments. LCA (Loosely Coupled approach) is one of these solutions, which use the same technique as AODV (Ad-Hoc On-Demand Distance Vector). The security aspect in LCA was ignored. In fact, it is vulnerable to attacks affecting the session establishment process; the aim of this thesis is to secure LCA, by using the identity-based signature scheme, which is certificate-less. In the proposed schemes (Secur_LCA) we use SIP URI address as a public key that can leads to smaller key size as compared to other cryptographic techniques, also to save overhead costs of communication and storage.

Keywords—Ad hoc, Identity-based signature, LCA, SIP, Security.

Table des matières

Liste des figures

Chapitre 1 : Le protocole de signalisation SIP

Chapitre 2 : Les réseaux Ad-hoc

Chapitre 3 : Etat de l'art « Décentralisation du protocole SIP dans les réseaux Ad-hoc »

Chapitre 4 : Notre Contribution Secure_LCA

Liste des tableaux

Chapitre 1 : Le protocole de signalisation SIP

Chapitre 3 : Etat de l'art « Décentralisation du protocole SIP dans les réseaux Ad-hoc »

Chapitre 4 : Notre Contribution Secure_LCA

Sigles et abréviations

Introduction générale

Les applications multimédia utilisent essentiellement deux protocoles, l'un pour la signalisation et l'autre pour le transfert de données. Dans ce mémoire, nous nous sommes intéressés aux protocoles de signalisation. Les protocoles de signalisation sont utilisés pour l'établissement des sessions de communication multimédia entre plusieurs participants.

Le protocole SIP (Session Initiation Protocole) est l'un des protocoles piliers de la signalisation qui a montré son efficacité et ses performances, grâce à sa simplicité par l'utilisation d'une architecture répartie, basée sur plusieurs entités centralisées. C'est pourquoi, les chercheurs voulaient étendre son fonctionnement sur de nouveaux types de réseaux, tel que les réseaux ad-hoc.

Les réseaux ad-hoc, sont des réseaux ne disposant d'aucune infrastructure préexistante et formés de noeuds mobiles interconnectés par des liaisons sans fil. Leurs architectures évoluent au gré de l'apparition et du mouvement des noeuds. L'absence d'infrastructure se traduit par la nécessité de mettre en place des solutions adaptées reposant sur la participation de l'ensemble des noeuds formant le réseau ad-hoc. Avec leurs caractéristiques particulières telles que la mobilité et la facilité de mise en place, les réseaux ad-hoc sont déployés dans diverses applications comme les PANs (Personal Area Network), les opérations de secours et de sauvetage ainsi que les applications militaires et tactiques.

Pour exploiter le protocole SIP dans les réseaux ad hoc, il est indispensable de résoudre le problème de centralisation sur lequel se base SIP et le rendre capable de fonctionner de manière complètement distribuée. Plusieurs solutions ont été proposées dans ce cadre. Les chercheurs ont tenté de décentraliser le SIP dans les réseaux ad-hoc mais ils n'ont pas pensé à la sécurité de leurs solutions. Du point de vue de la sécurité, les réseaux ad hoc soulèvent de nombreux problèmes en comparaison aux réseaux avec infrastructure. En effet les nœuds sont connectés via des liens sans fil qui sont particulièrement vulnérables aux différentes attaques possibles. Cela se justifie par les contraintes et les limitations physiques (puissance de calcul et capacité de stockage limitées en plus des restrictions de la bande passante), qui font que le contrôle des données transférées doit être minimisé.

Parmi les solutions qui ont été proposées nous avons la solution LCA (Loosely Coupled Approche) [Banerjee et al, 2006] qui utilise une technique similaire au protocole AODV (Ad-hoc On-demande Distance Victor) pour établir des sessions multimédia. Chaque entité dans LCA joue le rôle d'un routeur et prend part de la responsabilité d'acheminement des messages de signalisation. Cette manipulation rend le LCA très vulnérable aux attaques. Le but de ce mémoire est de sécuriser le schéma de LCA en utilisant des fonctions cryptographiques robustes tout en minimisant la charge du

calcul complexe. Nous proposons un nouveau schéma nommé SECURE_LCA [Douara, 2012a] en utilisant le concept de la signature basé sur l'identité afin de garantir l'authentification et l'intégrité.

Ce mémoire est composé de quatre chapitres. Dans le premier chapitre, nous présentons le protocole de signalisation SIP, son fonctionnement ainsi que les mécanismes de sécurité qu'il utilise. Dans le deuxième chapitre, nous définissons précisément ce que c'est qu'un réseau sans fil Ad-hoc, nous en donnerons les principes fondamentaux, les propriétés et les protocoles que doivent suivre de telles structures, et aussi les challenges auxquels est confrontée la sécurité de ces réseaux. Dans le troisième chapitre, nous discutons la décentralisation de SIP, tout en expliquant les problèmes de son déploiement dans les réseaux ad-hoc et enfin nous présentons quelques solutions déjà proposées pour les résoudre. Dans le quatrième chapitre nous discutons la solution LCA et nous présentons le principe de la cryptographie basée sur l'identité afin de l'utiliser dans le schéma que nous proposons et qu'on a nommé Secure_LCA [Douara, 2012a]. Enfin nous clôturons ce mémoire par une conclusion qui synthétise le travail réalisé

Chapitre 1

Le protocole de signalisation SIP

1.1 Introduction

Durant plusieurs années, l'existence de deux réseaux parallèles dans le domaine de la communication ; l'un pour la transmission de la voix et l'autre pour le transfert des données a contraint les entreprises à s'équiper des deux réseaux pour pouvoir transférer la voix et les données ce qui avait généré des surcouts, tant en immobilisation de matériels qu'en ressources humaines.

Cette situation a évolué et s'est rapidement modifiée, avec l'introduction des mécanismes de numérisations de la voix qui est découpée en paquets transférés par un réseau IP (Internet Protocol) vers une autre application qui se charge de la transformation inverse (dé numérisation).

Grace à ces techniques, la voix est vue comme une simple donnée. Les nouvelles capacités des réseaux à haut débit permettent d'offrir le transfert des données de manière fiable, en temps réel. C'est ce qui a incité plusieurs organismes, entreprises et opérateurs téléphoniques à investir dans ce domaine, et ce la on vue de mettre en place des routines et des protocoles pour le transfert de la voix via les réseaux IP, tout en pour fixant les normes de communication afin d'assurer une complète interopérabilité.

Un protocole de signalisation est utilisé pour ouvrir des sessions multimédias, entre plusieurs participants. Il permet de négocier la façon de coder les informations selon le type de media utilisé et la bande passante en usage pour assurer la compatibilité de l'information échangée. Il est utilisé aussi pour libérer les sessions et, en cas de besoin, les modifier tout en authentifiant l'identité des participants.

Le protocole SIP [Rosenberg, et al, 2002] est le plus important et le plus populaire des protocoles existants, il fait l'objet de ce chapitre. Le but est de décrire les grands principes de SIP et de présenter les solutions de sécurité envisagées dans le RFC3261 [Rosenberg, et al, 2002]. L'idée de ce chapitre n'est pas de décrire la totalité du protocole SIP mais de mettre en avant les principes qui seront nécessaires pour les contributions de ce travail.

1.2 SIP (Session Initiation Protocol)

Le protocole SIP est originalement développé par L'IETF Multi-Party Multimedia Session Control Working Group, connue sous le nom MMUSIC. La première version du protocole a été établie en 1997, des améliorations significatives ont été retenues dans la version 2.0 en 1998. Le protocole SIP a été standardisé en mars 1999 et publié dans le RFC 2543 [Handley et al, 1999] en avril 1999. Des clarifications et des rectifications de quelques

bugs ont été soumises au début juillet 2000 et publié dans RFC 2543 " Bis ", le RFC 3261 a remplacé le RFC 2543 original du protocole SIP.

1.2.1 L'architecture globale

Le protocole SIP permet comme son nom l'indique d'initier, mais également de modifier et de terminer des sessions voix mais aussi multimédias. La session voix est l'équivalent de notre « appel téléphonique ». SIP se situe au niveau applicatif. Pour fonctionner, SIP a donc besoin d'autres standards ou protocoles. A ce titre, SIP est souvent décrit comme un protocole « chapeau » puisqu'il s'appuie sur d'autres briques protocolaires comme UDP [Postel, 1980] ou TCP [Postel, 1981] pour la couche Transport. La figure 1.1 présente la pile protocolaire SIP pour la signalisation et le média.

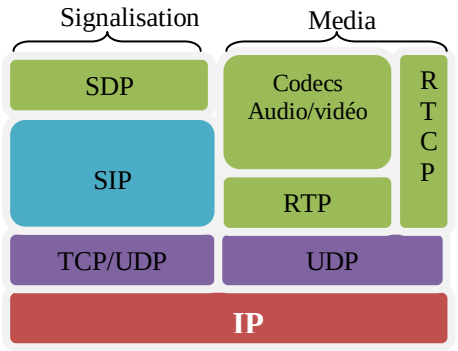


Figure 1-1 La pile protocolaire de SIP

Le fonctionnement de SIP s'appuie sur une architecture générique appelée « trapézoïde SIP » comme l'illustre la figure 1.2. Il existe deux grandes catégories d'acteurs dans cet environnement :

- les clients appelés « User Agent » (UA) qui initient et reçoivent les appels ;
- les serveurs qui relaient ou traitent les messages SIP émis par les UA ou les autres serveurs.

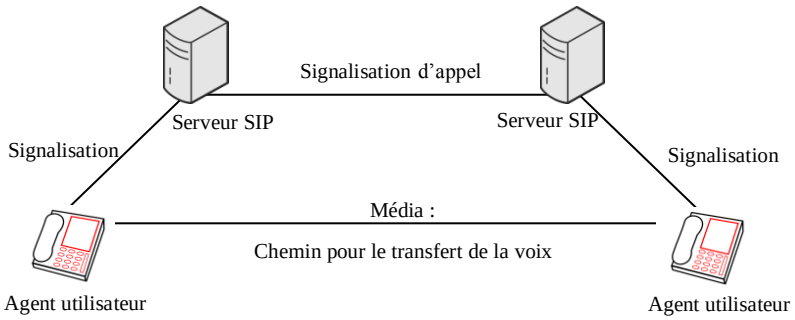


Figure 1-2 Architecture élémentaire SIP

L'établissement d'une communication se fait au travers d'échanges de messages entre les différents éléments du réseau. Ces échanges font partie de la signalisation. Une fois la session établie, les échanges de données (voix, images, vidéo) se font directement entre les deux extrémités. La voix est quant à elle transportée par le protocole RTP (Real-time Transport Protocol) [Schulzrinne, et al].

1.2.2 Adressage et nommage

SIP choisit l'Email comme adresse de la forme : "user@domain", " user@IP adresse", "numéro telephone@gateway". Cette adresse est appelée URI (Uniform Resource Identifier), Le nom de domaine peut être le nom de l'hôte sur lequel l'utilisateur est logé. L'adresse de la forme "numéro telephone@gateway" désigne le numéro du téléphone PSTN [Schulzrinne and Rosenberg, 2000]. A l'inverse des numéros de téléphone, les adresses SIP désignent une entité plutôt qu'un terminal. Cette différence fondamentale permet à celles-ci de représenter un individu, quelque soit sa localisation physique et également quelque soit le terminal à sa disposition.

1.2.3 Les entités SIP

Dans une architecture SIP on trouve deux familles d'entités SIP, les agents utilisateurs (UAC, UAS) et les serveurs.

1.2.3.1 Les agents utilisateurs

L'agent utilisateur – ou le terminal SIP – est le User Agent (UA). Il émet et reçoit les appels. Chaque UA est à associer à un identifiant URI SIP. On distingue :

1.2.3.1.1 Agent utilisateur client U.A.C (User Agent Client)

C'est un processus de type Client qui représente la machine de l'appelant. Son rôle est d'initier des requêtes.

1.2.3.1.2 Agent utilisateur serveur U.A.S (User Agent Server)

C'est un processus de type Serveur qui représente la machine appelée. Son rôle est de contacter l'utilisateur lorsqu'une requête SIP est reçue et de renvoyer une réponse au nom de l'utilisateur.

1.2.3.2 Les serveurs

Concernant les serveurs, il en existe 4 types, qui sont le serveur proxy (*Proxy Server PS*), le serveur de redirection (*Redirect Server RS*), le serveur de localisation (*Location Server LS*) et le serveur d'enregistrement (*Registrar Server RG*).

1.2.3.2.1 Serveur proxy - PS

Les proxys SIP sont des éléments qui routent les requêtes SIP vers des UAS et les réponses SIP vers les UAC. Une requête peut traverser de nombreux proxys avant d'arriver au UA. Chaque Proxy SIP fait la décision de routage, la modification des requêtes avant de la transférer vers un autre élément. Au retour, les réponses vont être routées à travers le même ensemble de proxy que la requête. Il existe deux types de serveur proxy :

- ***Proxy sans état***: Il se contente de renvoyer les messages sans garder de trace de la session.
- ***Proxy à état plein***: Il garde en mémoire l'état de l'appel et notamment l'état des transactions. Ainsi ces serveurs peuvent gérer une bonne partie des transactions de SIP et peuvent faire le choix de router un appel sortant.

1.2.3.2.2 Serveur de redirection - RS

Il réalise simplement une association d'adresses (mapping), chaque adresse d'UA est associée vers une ou plusieurs nouvelles adresses de proxy. Lorsqu'un client appelle un terminal mobile, le RS redirige l'appel vers le PS le plus proche auquel est relié le destinataire.

Un RS est consulté par l'UAC comme un simple serveur et ne peut émettre ou modifier de requêtes contrairement au PS.

1.2.3.2.3 Serveur de localisation - LS

Il est utilisé pour donner la position courante des utilisateurs dont la communication traverse les RS et PS auxquels ils sont rattachés. Cette fonction est assurée par le service de localisation.

1.2.3.2.4 Serveur d'enregistrement

Il est utilisé pour traiter les requêtes d'enregistrement *Register*. Il offre également un service de localisation comme le LS. Chaque PS ou RS est généralement relié à un RG.

1.2.4 Les messages SIP

Le protocole SIP est bâti sur une architecture Client/serveur et utilise des messages textuels semblables au HTTP. Un message SIP peut être soit une demande d'un client à un serveur, soit une réponse d'un serveur à un client. Et dans ces deux types de message (demande et réponse), le message est structuré comme le montre le tableau 1.1.

Tableau 1-1 Structure du message SIP

Ligne de début
En têtes
Ligne vide
CRLF
Corps du message

1.2.4.1 Ligne de Début

La ligne de début définit le type du message SIP. Elle peut être soit une Ligne de requête, dans ce cas le message SIP est appelé méthode SIP, ou ligne d'état, dans ce cas le message SIP est une réponse SIP.

1.2.4.1.1 Méthodes SIP

Le tableau 1.2 montre les différents champs qui constituent une requête SIP.

Tableau 1-2 Ligne de requête

Méthode	**URI**	**Version SIP**	**CRLF**

Il existe six méthodes de base:

1. INVITE : Requête d'établissement d'une session, invitant un usager (humain ou non) à participer à une communication téléphonique ou multimédia ; l'émetteur de cette requête y indique les types de média qu'il souhaite et peut recevoir, en général au travers d'une description de session SDP (Session Description Protocol) [Handley et al, 2006].

2. ACK : Requête d'acquittement, émise pour confirmer que le client émetteur d'un INVITE précédent a reçu une réponse finale ; cette requête peut véhiculer une description de session qui clôt la négociation.

3. OPTIONS : Un proxy server en mesure de contacter un terminal appelé, doit répondre à une requête OPTIONS, en précisant ses capacités à contacter le même terminal.

4. BYE : Cette requête est utilisée par le terminal de l'appelé afin de signaler qu'il souhaite mettre fin à la session.

5. REGISTER : Requête à destination d'un serveur SIP et permettant de lui faire parvenir de l'information de localisation (machine sur laquelle se trouve l'utilisateur).

6. CANCEL: Requête d'annulation, signifiant au serveur de détruire le contexte d'un appel en cours d'établissement (cette requête n'a pas d'effet sur un appel en cours).

D'autres requêtes existent mais sont issues d'autres RFC comme : MESSAGE pour l'envoi de message instantané, PRACK pour la sécurisation des réponses provisoires, PUBLISH pour l'envoi d'une information relative à un état vers un serveur, INFO pour l'envoi d'information ne modifiant pas la session et UPDATE pour la mise à jour des paramètres média avant la réponse finale au premier INVITE.

1.2.4.1.2 Les réponses SIP

Une réponse à une requête SIP est caractérisée, par un code et un motif, appelés code d'état et raison phrase respectivement comme le montre le tableau 1.3.

Tableau 1-3 Ligne d'état

Version SIP	Code d'état	Phrase raison	CRLF

Le code d'état est un entier codé sur trois chiffres indiquant une réponse à l'issue de la réception de la requête. Ce résultat est précisé par une phrase, textbased, expliquant la cause du refus ou le motif de l'acceptation de la requête. Le code d'état est donc destiné à l'automate gérant l'établissement des sessions SIP et les motifs aux programmeurs. Il existe six classes de réponses et donc de codes d'état, représentées par le premier chiffre, Le tableau 1.4 donne quelques réponses possibles :

Tableau 1-4 Les principales familles des réponses

Code	Définition de la famille de réponse	Principales réponses
1XX	Réponse intermédiaire d'information (traitement en cours)	100 Trying 180 Ringing
2XX	Succès	200 OK
3XX	Redirection	301 Moved permanently 302 Moved temporarily
4XX	Erreur client	400 Bad Request 401 Unauthorized
5XX	Erreur serveur	500 Server Internal Error 501 Not Implemented
6XX	Echec global du traitement	600 Buzy Everywhere

1.2.4.2 En-têtes SIP

L'en-tête contient les informations permettant l'acheminement du message comme : la référence de l'émetteur, le destinataire, référence de la transaction et de la session, les éléments de sécurité. Ainsi l'en-tête permet l'établissement d'une session en termes de localisation, de nommage et d'adressage**,** Les champs des entêtes les plus usuels sont le From, le To, le Call-ID, le Cseq, le Contact. Les principaux champs avec leur signification sont donnés dans le tableau 1.5.

Tableau 1-5 Les principaux champs d'en-tête des messages SIP

Champ d'en-tête	Description
Authorization :	Information d'authentification pour l'usage d'une ressource par un UA
Call-ID :	Identifiant unique pour un échange d'établissement particulier
Contact :	Généralement, URL de l'usager
Content-Length :	Longueur du message en octets
Content-Type :	Type du corps du message (par exemple une description SDP)
CSeq :	Identifie une requête à l'intérieur d'une session
Encryption :	Précise que le contenu est chiffré
From :	Initiateur de la requête
Max-Forwards :	Limite au nombre de serveurs et de proxies qui peuvent router le message
Proxy-Authenticate :	Information pour l'authentification d'un usager auprès d'un proxy
Proxy-Authorization :	Information pour l'authentification d'un usager auprès d'un proxy
Proxy-Require :	Précise un mécanisme qui doit être fourni par le proxy
Timestamp :	Date d'émission du message
To :	Précise le destinataire de la requête
Unsupported :	Liste les mécanismes non supportés par le serveur
User-Agent :	Information sur l'UA qui a généré le message
Via :	Dénote le chemin emprunté par la requête jusqu'à l'instant présent
WWW-Authenticate :	Inclus dans les réponses 401, dans le but d'authentifier l'émetteur de la requête

1.2.4.2.1 En-tête générale

Le champ d'en-tête général s'applique à la fois aux messages de requête et de réponse et contient les informations indispensables pour le déroulement de la session comme le chemin que doit ou qu'a traversé le message, le temps d'expiration du message, la destination du message etc.

1.2.4.2.2 En-tête d'entité

Le champ d'en-tête d'entité définit le type d'informations contenues dans le corps du message ou la ressource identifiée par la requête en l'absence du corps du message.

1.2.4.2.3 En-tête de requête

Le champ d'en-tête de requête autorise le client à ajouter des informations concernant sa requête lui même à destination du serveur.

1.2.4.2.4 En-tête de réponse

Le champ d'en-tête de réponse autorise le serveur à ajouter des informations concernant sa réponse qui ne peuvent pas être placées dans la ligne d'état. Ces informations sont sur lui même et sur l'accès à la ressource identifiée par la requête URI.

1.2.4.2.5 CRLF

Balise pour indiquer la fin du champ d'en-têtes et le début du Corps du message.

1.2.4.2.6 Corps du message

Ce champ est facultatif. Il est utilisé pour contenir, entre autre, les descriptions de la session faite par Session Description Protocol (SDP) [Handley et al, 2006]. Les participants dans une session SIP utilisent le protocole SDP pour arriver à une vue commune d'une session multimédia entre eux. Un participant offre à l'autre une description de la session désirée dans sa perspective, et l'autre participant répond par la session désirée de son point de vue. Ceci permet de donner une vision complète de la session.

1.2.5 Fonctionnement

Nous allons présenter dans la suite de cette section le fonctionnement de SIP en décrivant ses deux fonctions de base qui sont l'enregistrement et le déroulement d'une session.

1.2.5.1 Enregistrement au réseau SIP

L'enregistrement est la première étape qu'un terminal SIP doit faire avant de commencer une session. Il envoie entre autre son adresse IP et son adresse SIP au RG afin de l'enregistrer et de le localiser.

La Figure 1.3 décrit un exemple d'enregistrement d'un terminal SIP *A* auprès d'un *RG*.

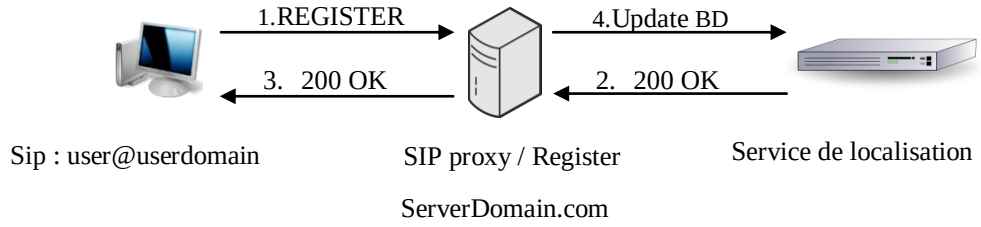


Figure 1-3 Exemple d'enregistrement au réseau SIP

(1) le terminal *A* envoie une demande d'enregistrement *Register* au serveur *RG* qui contient son adresse IP et son URI SIP.

(2) Le *RG* envoie une demande de mise à jour de la base de données du *LS* contenant les informations de *A*.

(3) Le *LS* confirme la mise à jour en envoyant un message d'acquittement *200 OK* au *RG*.

(4) Le *RG* confirme l'enregistrement du terminal *A* en lui envoyant un *200 OK*.

En dehors des mécanismes de sécurité pouvant être implémentés, la procédure d'enregistrement se résume par l'envoi d'un message REGISTER par l'UA qui reçoit une réponse 200 OK du REGISTRAR. Cette description, ne prend pas en compte le mécanisme d'authentification HTTP Digest relativement usuel dans l'implémentation de SIP. Ce mécanisme modifie la nature des transactions. Ce point sera traité ultérieurement.

1.2.5.2 Déroulement d'une session

La figure 1.4 illustre le déroulement d'une session SIP, entre deux interlocuteurs SIP A : Ahmed@usto.dz et B : Ali@enset.dz, en trois étapes. Les deux terminaux sont supposés déjà enregistrés auprès de leurs RGs respectifs.

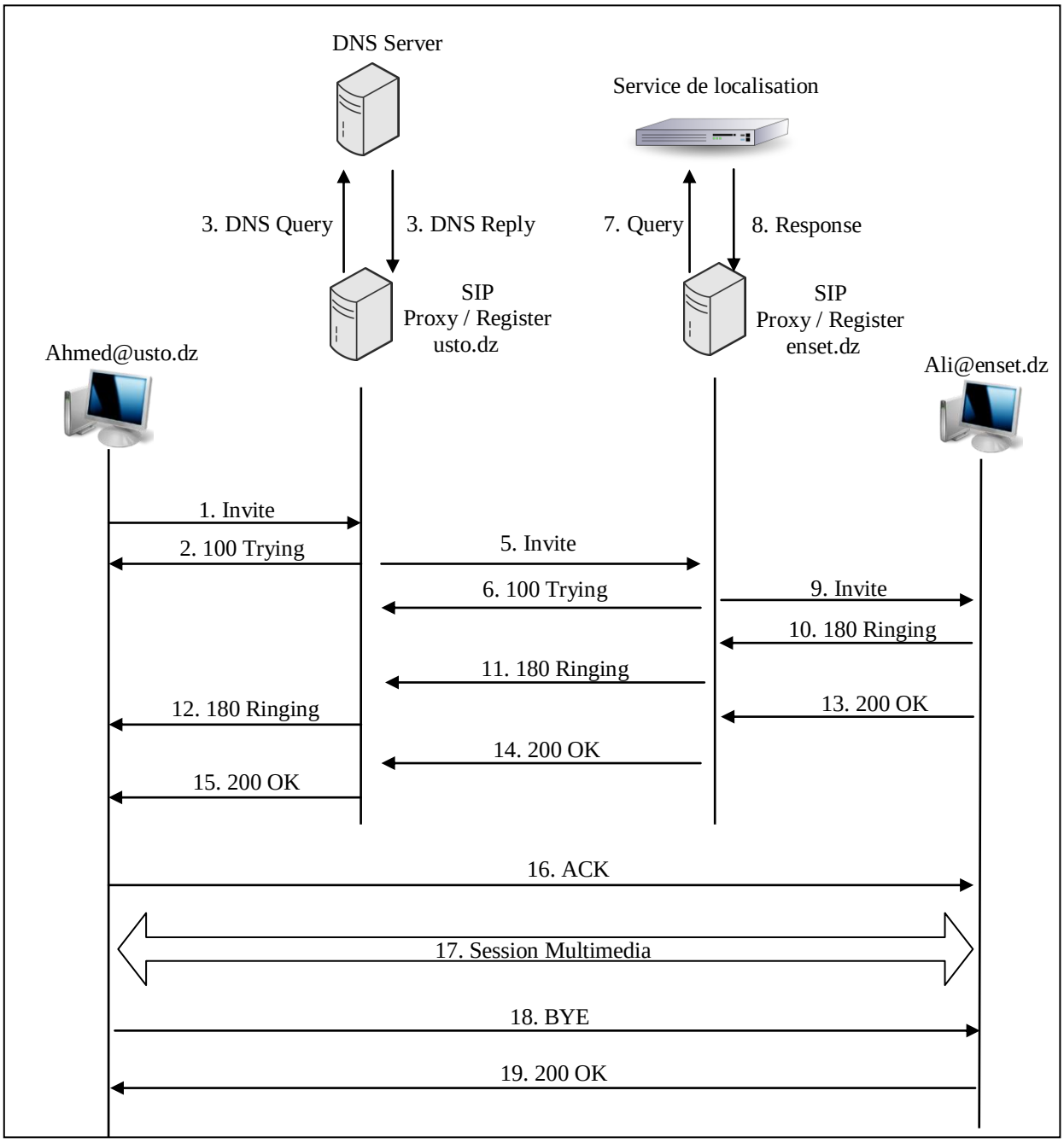


Figure 1-4 Etablissement de session

A. Demande d'établissement de session

Les étapes 1 à 16 (figure 1.4) décrivent l'établissement de session entre les terminaux *A* et *B*.

(1) L'UA *A* envoie une requête *INVITE* au PS usto.dz, cette requête contient l'URI SIP de *B* et une syntaxe SDP décrivant les caractéristiques du média qui sera échangé durant la session.

(2) Le PS usto.dz renvoie un message *100 TRYING* comme première réponse de l'*INVITE*.

(3) Le PS usto.dz interroge la base de données du serveur DNS.

(4) Le serveur DNS localise le PS de l'enset.dz et envoie son adresse au PS usto.dz.

(5) Le PS usto.dz ajoute un champ VIA additionnel, contenant sa propre adresse, à la requête *INVITE* initiale pour que les réponses traversent le même ensemble de proxys. Ensuite la requête INVITE est envoyée au PS enset.dz.

(6) Le PS enset.dz essaye de localiser le terminal *B* et envoie un message *100TRYING* au PS usto.dz comme première réponse.

(7) Le PS enset.dz interroge la base de données du LS.

(8) Le LS envoie l'adresse IP et le numéro du port de *B* au PS enset.dz.

(9) Après la localisation de *B*, le PS enset.dz ajoute un champ VIA additionnel contenant son adresse, à la requête *INVITE* avant de l'acheminer vers *B*.

(10) Dès l'arrivée de la requête *INVITE* à *B*, ce dernier génère une réponse *180 RINGING* qui suit le chemin inverse de la requête *INVITE* à destination de *A* (étapes 11,12).

(13) Quand *B* décroche une réponse *200 OK* est envoyée à *A* suivant le même chemin du *180 RINGING* (étapes 14, 15).

(16) *A* acquitte la réception du *200 OK* en envoyant un *ACK* à *B*. ce dernier message est envoyé directement de *A* à *B* sans passer par aucun PS.

B. Etablissement de la session multimédia

Dans cette phase les deux interlocuteurs échangent leurs données. Les protocoles utilisés pour assurer ces transferts sont le RTP et le RTCP, comme illustré dans l'étape 17 de la figure 1.4.

C. Libération de la session

La figure 1.4 montre un exemple de libération de session initié par *B*.

(1) Une requête *BYE* est générée par le terminal de *B*. Cette requête est envoyée directement à *A* sans passer par aucun proxy.

(2) *A* confirme la fin de session avec un *200 OK*.

1.2.6 SIP et la sécurité

Le RFC de SIP prévoit un certain nombre de mécanismes de sécurité pour assurer la confidentialité, l'intégrité, l'anonymat et l'authentification au travers de la signalisation. Les mécanismes qui fournissent la sécurité dans SIP peuvent être classés en deux sortes de protection : de **terminal-au-terminal** ou de **proxy-par-proxy** [Salsano et al, 2002]. Les mécanismes terminal-au-terminal impliquent les agents utilisateurs SIP. Les mécanismes proxy-par-proxy assurent la communication entre deux entités SIP successives dans le parcourt de message de signalisation. SIP ne fournit pas de caractéristiques spécifiques pour la protection proxy-par-proxy et compte sur la sécurité dans la couche réseau ou la couche du transport (Figure 1.5). Les mécanismes proxy-par-proxy sont requis parce que des éléments intermédiaires peuvent jouer un rôle actif dans le traitement SIP en lisant et/ou en écrivant certaines parties des messages SIP.

Deux principales techniques de sécurité sont employées par SIP: l'authentification et le cryptage des données :

L'authentification est employée pour authentifier l'envoyeur du message et pour assurer qu'une certaine information critique de message soit non modifiée dans le transit. Elle doit empêcher un agresseur de modifier et/ou d'écouter des requêtes et des réponses SIP.

Le cryptage de données permet seulement au donataire destiné de décrypter et de lire les données. Le cryptage terminal-au-terminal fournit une confidentialité pour toute les informations (certaines entête et le corps de message SIP) qui n'on pas besoin d'être lues par des serveurs intermédiaires de proxy.

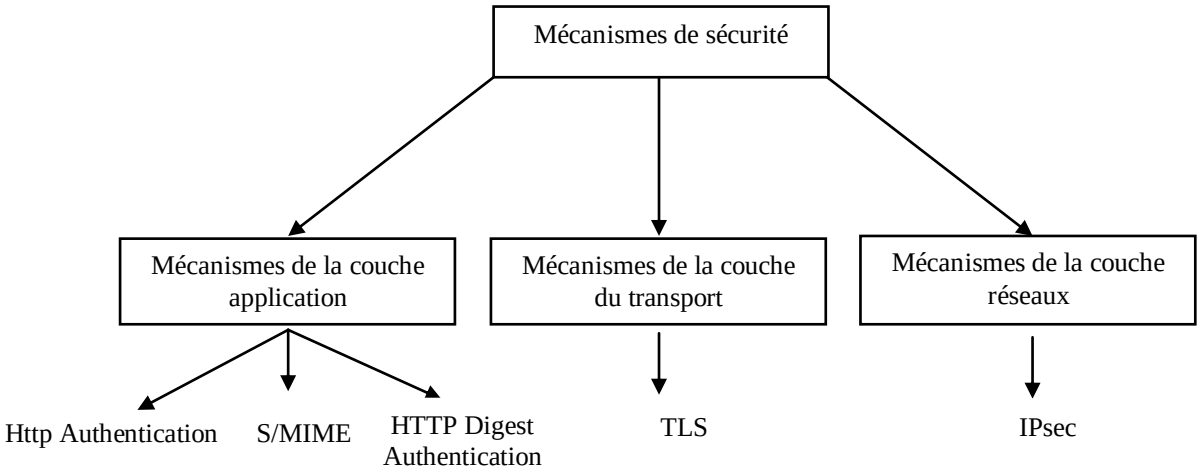


Figure 1-5 Mécanismes de sécurité SIP

1.2.6.1 La sécurisation de la signalisation

1.2.6.1.1 L'authentification HTTP

L'authentification HTTP (méthode « Digest » et méthode « Basic ») [Franks et al, 1999] est un mécanisme basé sur un challenge/réponse. Il permet tout d'abord au client SIP de s'enregistrer auprès du REGISTRAR et ensuite d'avoir accès aux différentes ressources quand le serveur lui demande : une authentification est généralement demandée pour une requête INVITE. La figure 1.6 illustre l'authentification dans le cas d'un enregistrement.

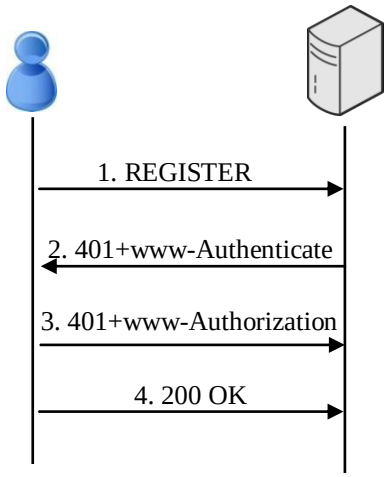


Figure 1-6 Enregistrement et authentification HTTP Digest dans un contexte SIP

Le principe de l'authentification HTTP Digest est classique. Le serveur envoie un challenge au client « nonce ». Ce dernier répond par une valeur « response » dérivée de ce challenge et d'un secret qu'il partage avec le serveur, généralement fourni avec le login par l'opérateur. Le serveur s'assure alors que le client possède effectivement le secret en calculant à son tour la réponse et en vérifiant la cohérence des deux. La version 2 de SIP déconseille la version « Basic » HTTP qui nécessite l'envoi du mot de passe en clair.

Les échanges de messages pour un enregistrement et le principe de l'authentification sont illustrés dans la figure 1.7. Le premier message informe le serveur du souhait du client de s'enregistrer par l'envoi d'une requête REGISTER. La réponse « 401 Unauthorized » permet au serveur d'envoyer son challenge sous la forme du champ « nonce » inclus dans le message SIP. Le client calcule la réponse « response » avec le secret pré-partagé qui renvoie dans une nouvelle requête REGISTER. Si la valeur « response » est conforme à l'attente du serveur, ce dernier envoie donc une réponse « 200 Ok ». Le client est enregistré, il peut donc téléphoner.

SIP fournit donc un mécanisme d'authentification simple basé sur HTTP Digest directement intégré dans l'en-tête des messages SIP. A chaque requête d'un usager, le serveur

SIP peut demander une authentification (sauf pour ACK). Enfin, il faut noter que HTTP Digest est le seul mécanisme de sécurité entièrement situé dans l'en-tête SIP.

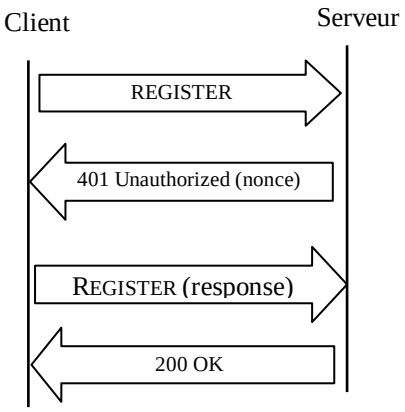


Figure 1-7 Enregistrement et authentification HTTP Digest dans un contexte SIP

1.2.6.1.2 S/MIME

Les messages SIP portent des corps de type MIME21 et peut donc utiliser sa version sécurisée S/MIME [Ramsdell, 2004]. S/MIME permet de sécuriser une partie des messages SIP en utilisant le principe de chiffrement clé publique. Il permet d'assurer la confidentialité, l'authentification et l'intégrité. Les certificats permettent soit de chiffrer, soit de signer les messages SIP. La confidentialité et l'intégrité sont assurées par l'utilisation de la clé publique du destinataire. L'authentification et l'intégrité sont quant à eux assurés en utilisant la clé privée de l'émetteur. S/MIME dans un contexte SIP permet trois utilisations, la transmission d'un certificat, la signature et le chiffrement.

Le RFC3261 [Rosenberg, et al, 2002] décrit l'application de S/MIME au contexte SIP. Il définit en particulier un mode tunnel qui assure la confidentialité, l'intégrité et l'authentification. Cette propriété est fournie à deux niveaux dans la signature et le chiffrement. En effet S/MIME utilise la clé privée de l'usager pour signer l'en-tête du message SIP, l'appelant s'authentifie ainsi. L'authentification du destinataire est quant à elle garantie en appliquant sa clé publique au corps du message. Le choix des champs à chiffrer a principalement un impact sur la confidentialité et l'intégrité mais pas sur l'authentification, qui repose sur l'utilisation des certificats.

S/MIME permet une authentification mutuelle entre l'appelant et l'appelé. Cette solution bout-en-bout nécessite l'échange des certificats auparavant. Le dispositif pourrait alors être porté par le fournisseur qui déploierait les softphones avec les associations adresse/certificat. Cette configuration est envisageable dans un domaine de confiance. SIP

prévoit d'autres mécanismes d'authentification pour la signalisation mais dans les couches sous-jacentes.

1.2.6.1.3 TLS et SIPS

SIP prévoit la sécurisation des échanges au niveau de la couche Transport avec Transport Layer Security (TLS) [Dierks, 1999]. TLS, anciennement nommé Secure Sockets Layer (SSL), est un protocole de sécurisation des échanges sur Internet. TLS est un protocole modulaire dont le but est de sécuriser les échanges Internet entre le client et le serveur indépendamment de tout type d'application. TLS agit comme une couche supplémentaire au-dessus de TCP. Ainsi TLS ne s'occupe pas de fiabilité de couche Transport ni du maintien de la connexion. Les services offerts sont : l'authentification, l'intégrité et la confidentialité. Son implémentation native dans de nombreux navigateurs a fait de TLS le standard de sécurisation des applications Web : HTTPS correspondant à l'association d'HTTP avec TLS. Son utilisation est principalement associée à l'utilisation des certificats X.50922 pour l'authentification des serveurs et le chiffrement des échanges (i.e. la signalisation). Le RFC initial de SIP ne décrivant que très sommairement l'association SIP/TLS, [Peterson and Jennings, 2006] a été édité pour préciser le fonctionnement des deux protocoles.

TLS fournit la sécurité de la couche Transport en mode connecté. L'utilisation de TLS est spécifiée dans le champ via de l'en-tête SIP ou dans l'URI SIP. Ce choix entraine l'ouverture d'une connexion TLS classique permettant par la suite des échanges de messages SIP chiffrés. Par ailleurs, TLS est bien adapté aux architectures dans laquelle la sécurité de proche-en-proche est demandée alors qu'il n'existe pas de relations de confiance. Les serveurs peuvent s'échanger leur certificat et les faire vérifier auprès d'une autorité de confiance. TLS est spécifique à une application qui est explicitement associée à un port (5061 pour SIP/TLS, 5060 pour SIP/TCP ou UDP).

TLS permet au client d'authentifier le serveur. L'utilisation d'un certificat client autoriserait une authentification mutuelle au niveau Transport, mais obligerait le serveur à posséder le certificat avec la clé publique de tous les usagers. On peut considérer que HTTP Digest + TLS permet une authentification mutuelle. Concernant une connexion TLS entre deux domaines, le RFC de SIP préconise fortement une authentification mutuelle.

1.2.6.1.4 IPSec

Pour protéger les échanges dans les réseaux, une des solutions usuelles consiste à utiliser le protocole IPsec (IP security) [Kent and Atkinson, 1998], la version sécurisée d'IP. De même que SIP prévoit la sécurisation des échanges au niveau de la couche Transport, il

envisage une protection au niveau Réseau avec IPSec. Ce protocole permet en effet d'authentifier l'origine de paquets IP, de garantir l'intégrité voire la confidentialité. IPSec permet donc de protéger des communications et la signalisation entre deux entités. Deux modes sont possibles : le mode transport ou le mode tunnel. Quelque soit le mode, le serveur SIP peut modifier les en-têtes SIP et permettre l'établissement de l'appel. D'une manière générale, les clients SIP n'implémentent pas cette solution. IPSec est donc principalement utilisé pour protéger le trafic entre deux domaines [Sawda and Urien, 2006].

IPSec permet l'encapsulation des datagrammes IP. Toutes les applications dont celles basées sur SIP peuvent donc bénéficier de ses propriétés de sécurité. IPSec est un ensemble de protocoles complètement indépendant de SIP spécifiant essentiellement deux aspects :

- l'encapsulation des datagrammes IP dans d'autres datagrammes IP de manière à fournir des services de sécurité classiques : intégrité, confidentialité, authentification.
- la négociation des clés et des associations de sécurité utilisées lors de l'encapsulation.

1.3 Conclusion

SIP est un protocole pour l'interactivité en temps réel. Il a été développé par l'IETF et s'inspire du protocole HTTP. SIP est plus modulaire et peut fonctionner avec d'autres protocoles. Sa simplicité, sa rapidité ainsi que sa légèreté d'utilisation, qui sont complets, sont autant d'arguments qui peuvent influer favorablement sur le choix des investisseurs. De plus, ses avancées en matière de sécurité des messages sont un atout supplémentaire par rapport à ses concurrents, d'autant qu'il possède l'avantage de ne pas être attaché à un médium particulier et qu'il est indépendant des protocoles de transport de la couche basse. Aussi dans la suite de ce mémoire, nous avons opté pour la sécurisation de SIP dans les réseaux Ad hoc. Mais avant cela, nous allons d'abord présenter les réseaux ad hoc dans le prochain chapitre.

Chapitre 2

Les réseaux Ad-hoc

2.1 Introduction

Les équipements mobiles deviennent de plus en plus petits et puissants en termes de capacité de traitement et de stockage de données. De plus, ils sont dotés d'une multitude de fonctionnalités qui permettent d'assurer différents types d'applications et de services.

Parmi les applications et services offerts via un équipement mobile, figurent les services de connexions et les services de données correspondants. Ces derniers représentent le service le plus demandé par les utilisateurs mobiles. Par exemple, les connexions entre deux téléphones mobiles cellulaires sont assurées par les BSC (Base Station Controller) et les MCS (Mobile services Switching Center), les ordinateurs portables sont connectés à Internet via des points d'accès fixes.

Il y a, en outre, des situations spécifiques où les besoins de connexions des utilisateurs ne sont pas assurés par le réseau dans une zone géographique donnée. Dans cette situation fournir la connectivité est un réel défi. Récemment, de nouvelles alternatives pour fournir les services ont été proposées. Elles sont basées sur le fait d'avoir des stations mobiles interconnectées les unes aux autres grâce à une configuration autonome, créant ainsi un réseau Ad hoc flexible et performant. Parallèlement, le réseau Ad hoc peut être utilisé pour l'extension d'un réseau filaire. Dans ce cas, les nœuds mobiles peuvent avoir accès à l'Internet à travers une passerelle, pour étendre les services de l'Internet au-delà de l'infrastructure filaire.

Historiquement, les réseaux mobiles Ad hoc ont été d'abord introduits pour l'amélioration des communications dans le domaine militaire. Dans ce contexte, il n'existe pas d'infrastructure existante pour relier les communications, vue la nature dynamique des opérations et des champs militaires.

Les premières applications dans les réseaux Ad hoc sont apparues avec le projet PRNet (Packet Radio Network) [Freebersyser et al, 2001] en 1972. Ce projet a été inspiré par l'efficacité de la technologie par commutation de paquet, le partage de la bande passante, le routage 'store-and-forward', et ses applications dans l'environnement mobile sans fil.

SURAN (Survivable Radio Networks) [Westcott et al, 1984] a été développé par la DARPA en 1983 pour adresser les principaux problèmes du projet PRNet dans le domaine de la stabilité, la sécurité, la capacité de traitement et gestion d'énergie. Les objectifs étaient de proposer des algorithmes qui peuvent supporter jusqu'à une dizaine de milliers de noeuds,

tout en utilisant des mécanismes radio simples, avec une faible consommation d'énergie, et un faible coût. Ce travail a amené à la conception de la technologie LPR (Low-cost Packet Radio) [Fifer et al, 1987] en 1987, dotée d'une couche radio DSSS (Direct Sequence Spread-Spectrum) avec un processeur pour la commutation de paquets intégré (Intel 8086). De plus, une famille de protocoles pour la gestion du réseau a été développée, et une topologie hiérarchique du réseau basée sur un clustering dynamique est utilisée pour remédier au problème de la stabilité. Des améliorations pour l'adaptabilité de la couche radio, la sécurité et l'augmentation de la capacité ont été proposées.

L'évolution des infrastructures du réseau Internet et la révolution de la micro informatique ont permis de rendre faisables et applicables les idées initiales des réseaux radio de paquets. Le programme GloMo (Global Mobile) [Leiner et al, 1996] initié par la DARPA en 1994 avait comme objectif de supporter les communications multimédia n'importe quand et n'importe où à travers des équipements sans fil.

Tactical Internet (IT) [Freebersyser et al, 2001] est l'une des implémentations des réseaux sans fil Ad hoc grandeur nature développée par l'armée américaine en 1997, utilisant des débits de plusieurs dizaines de kilobits par seconde.

Un autre déploiement a été réalisé en 1999, avec ELB ACTD (Extending the Littoral Battle-space Advanced Concept Technology Demonstration) [Althouse, 1999] qui permet de démontrer la faisabilité de concepts militaires pour les communications des bateaux en mer aux soldats sur la terre par l'intermédiaire d'un relais aérien. 20 noeuds dans le réseau ont été considérés.

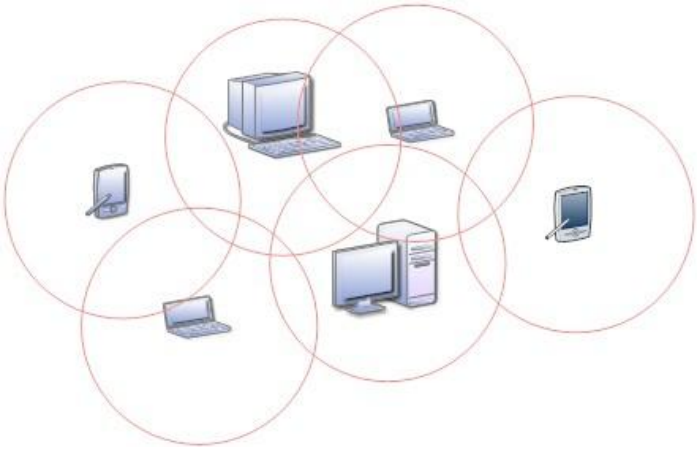
2.2 Les Réseaux sans fil Ad hoc

2.2.1 Définition

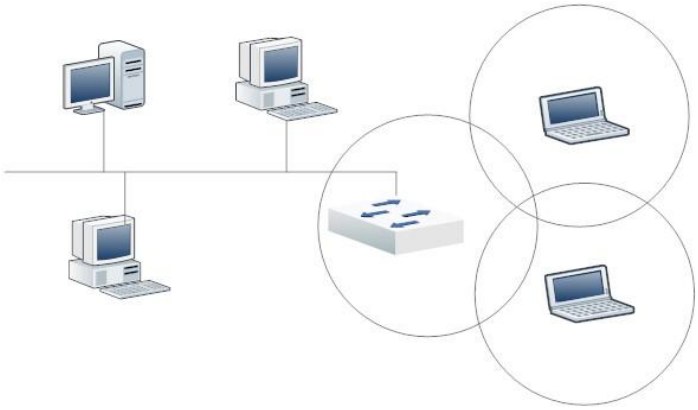
Les réseaux ad-hoc sont décrits et étudiés par le groupe de travail Mobile Ad-hoc NETworks (MANET) de "l'Internet Engineering Task Force (IETF)".

Une définition de ces réseaux est donnée formellement dans la RFC 2501 [Corson, 2002]. Il s'agit de réseaux sans fil qui composent de systèmes informatiques divers, plus ou moins complexes, appelés noeuds, par la suite, ayant la possibilité de communiquer de manière autonome par ondes radio. Les noeuds interagissent et peuvent coopérer pour s'échanger des services. Ces réseaux sont dits Ad hoc dans la mesure où ils ne nécessitent pas d'infrastructure fixe. Ils peuvent exister temporairement pour répondre à un besoin ponctuel

de communication. Le mode de fonctionnement Ad hoc se distingue du mode infrastructure dans lequel les noeuds du réseau communiquent entre eux via un point d'accès, aussi appelé base, qui peut être relié à un réseau fixe. La figure 2.1 montre la différence d'utilisation des réseaux sans fil en mode infrastructure fixe et en mode Ad hoc.



Mode Ad hoc



Mode infrastructure

Figure 2-1 Mode Ad hoc versus mode Infrastructure

Les réseaux sans fil Ad hoc s'appuient sur les technologies sans fil conçues à l'origine pour des réseaux locaux et domestiques :

- Les technologies IEEE 802.11a, 802.11b (*Wireless Fidelity*, *Wifi*), 802.11g, HiperLan/1 (remplacé par HiperLan/2). HomeRF (*SWAP*) : sont propres aux réseaux WLAN (*Wireless Local Area Network*).
- La technologie Bluetooth, pour les réseaux WPAN (*Wireless Personal Area Network*). Bluetooth fonctionne en mode point à point ou point à multipoint.

- Les technologies infrarouges (*IrDA. Infrared Data Association*), utilisées dans les télécommandes par exemple, peuvent aussi être considérées comme support des réseaux Ad-hoc. Mais ces technologies se limitent à des communications point à point.
- Les technologies Wifi. IEEE 802.11g. HiperLan. HomeRF et Bluetooth opèrent dans la bande ISM (*Industrial, Scientific and Medical*) à 2.4 GHz alors que 802.11a opère dans la région des 5 GHz.

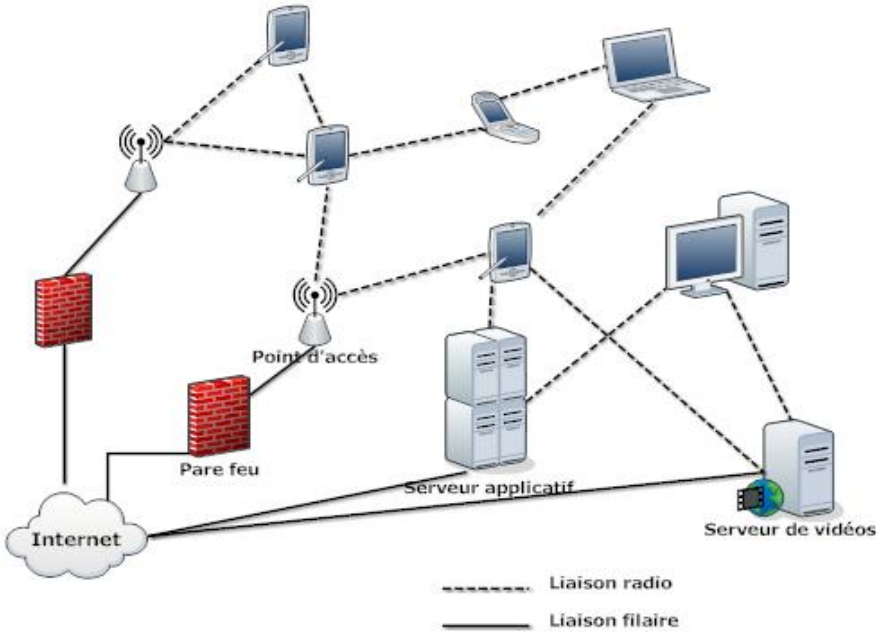


Figure 2-2 Un réseau Ad hoc

Les réseaux Ad hoc peuvent également être connectés au monde filaire (figure 2.2) par l'intermédiaire d'une ou plusieurs passerelles, que nous appellerons, en référence au monde cellulaire IP, des points d'accès (AP). De tels réseaux sont communément appelés réseaux hybrides [Theoleyre, 2006]. Chaque terminal du réseau Ad hoc, s'il possède une double interface filaire et sans fil peut donc agir en tant que passerelle pour les autres clients de la bulle Ad hoc. Les réseaux hybrides constituent les prémices de l'internet ubiquitaire de demain.

2.2.2 Contextes d'utilisation des réseaux Ad hoc

Les premières applications des réseaux Ad hoc concernaient les communications et les opérations dans le domaine militaire. Cependant, avec l'avancement des recherches dans le domaine des réseaux et l'émergence des technologies sans fil (ex : Bluetooth, IEEE 802.11 et HiperLan), d'autres applications civiles sont apparues. On distingue :

- ✓ Les services d'urgence : Opération de recherche et de secours des personnes, tremblement de terre, feux, inondation, dans le but de remplacer l'infrastructure filaire.
- ✓ Le travail collaboratif et les communications dans des entreprises ou bâtiments : Dans le cadre d'une réunion ou d'une conférence par exemple.
- ✓ Home network : Partage d'applications et communications des équipements mobiles.
- ✓ Applications commerciales : Pour un paiement électronique distant (taxi) ou pour l'accès mobile à l'Internet, où service de guide en fonction de la position de l'utilisateur.
- ✓ Réseaux de senseurs : Pour des applications environnementales (climat, activité de la terre, suivi des mouvements des animaux, etc.), ou domestiques (contrôle des équipements à distance).
- ✓ Réseaux en mouvement : Informatique embarquée et véhicules communicants.
- ✓ Réseaux Mesh : C'est une technologie émergente qui permet d'étendre la portée d'un réseau ou de le densifier.

En plus, dans un WLAN, un réseau Ad hoc fournit une solution pour étendre une couverture sans fil avec un moindre coût. Dans un WPAN (ex : UMTS), il permet d'accroître la capacité globale du réseau sans fil. En fait, plus de bande passante agrégée peut être obtenue en réduisant la taille des cellules et en créant des pico-cellules. Afin de supporter une telle architecture, les opérateurs disposent de deux options : déployer plus de stations de base (une station de base par cellule), ou utiliser un réseau Ad hoc pour atteindre la station de base. La deuxième solution est clairement plus flexible et moins coûteuse.

2.2.3 Propriétés et spécificités des réseaux Ad hoc

En général, un réseau Ad hoc mobile est considéré comme un système autonome dynamique composé de noeuds mobiles interconnectés par des liens sans fil, sans l'utilisation d'une infrastructure fixe et sans administration centralisée [Corson, 1999]. Les noeuds sont libres de se déplacer aléatoirement et s'organisent arbitrairement. Par conséquent, la topologie du réseau peut varier de façon rapide et surtout imprévisible. La route entre un nœud source et un nœud destination peut impliquer plusieurs sauts sans fil, d'où l'appellation de "réseaux sans fil multi-sauts". Un nœud mobile peut communiquer directement avec un autre nœud s'il est dans sa portée de transmission. Au delà de cette portée, les noeuds intermédiaires jouent le rôle de routeurs (relayeurs) pour relayer les messages saut par saut.

Les réseaux Ad hoc héritent des mêmes propriétés et problèmes liés aux réseaux sans fil. Particulièrement, le fait que le canal radio soit limité en termes de capacité, plus exposé

aux pertes (comparé au médium filaire), et sujet à des variations dans le temps. Le canal est confronté aux problèmes de "station cachée" et "station exposée". En outre, les liens sans fil sont asymétriques et pas sécurisés.

D'autres caractéristiques spécifiques aux réseaux Ad hoc conduisent à ajouter une complexité et des contraintes supplémentaires qui doivent être prises en compte lors de la conception des algorithmes et des protocoles réseaux, à savoir :

- ***L'absence d'une infrastructure centralisée*** : Chaque nœud travaille dans un environnement pair à pair distribué, et agit en tant que routeur pour relayer des communications, ou génère ses propres données. La gestion du réseau est ainsi distribuée sur l'ensemble des éléments du réseau.
- ***La mobilité des noeuds et maintenance des routes*** : La mobilité continue des noeuds, crée un changement dynamique de topologie. Par exemple, un nœud peut joindre un réseau, changer de position ou quitter le réseau. Ce déplacement a naturellement un impact sur la morphologie du réseau et peut modifier le comportement du canal de communication. Ajoutons à cela la nature des communications (longues et synchrones, courtes et asynchrones, etc.). Les algorithmes de routage doivent ainsi résoudre ces problèmes et supporter la maintenance et prendre en charge en un temps limité la reconstruction des routes tout en minimisant l'overhead généré par les messages de contrôle.
- ***L'hétérogénéité des noeuds*** **:** Un nœud mobile peut être équipé d'une ou plusieurs interfaces radio ayant des capacités de transmission variées et opérant dans des plages de fréquences différentes. Cette hétérogénéité de capacité peut engendrer des liens asymétriques dans le réseau. De plus, les noeuds peuvent avoir des différences en termes de capacité de traitement (CPU, mémoire), de logiciel, de taille (petit, grand) et de mobilité (lent, rapide). Dans ce cas, une adaptation dynamique des protocoles s'avère nécessaire pour supporter de telles situations.
- ***La contrainte d'énergie*** : Les équipements mobiles disposent de batteries limitées, et dans certains cas très limitées tels que les PDA, et par conséquent d'une durée de traitement réduite. Sachant qu'une partie de l'énergie est déjà consommée par la fonctionnalité du routage. Cela limite les services et les applications supportées par chaque nœud.
- ***La taille des réseaux Ad hoc*** : Elle est souvent de petite ou moyenne taille (une centaine de noeuds), le réseau est utilisé pour étendre temporairement un réseau filaire, comme pour une conférence ou des situations où le déploiement du réseau fixe n'est pas approprié

(ex : catastrophes naturelles). Cependant, quelques applications des réseaux Ad hoc nécessitent une utilisation allant jusqu'à des dizaines de milliers de noeuds, comme dans les réseaux de senseurs. Des problèmes liés au passage à l'échelle tels que : l'adressage, le routage, la gestion de la localisation des senseurs et la configuration du réseau, ***la sécurité***, etc., doivent être résolus pour une meilleure gestion du réseau.

- ***La faible sécurité :*** Il est facile 'd'espionner' un canal radio de manière passive. Les protections ne pouvant pas se faire de manière physique (il est en général difficile d'empêcher quelqu'un de placer discrètement une antenne réceptrice très sensible dans le voisinage), elles devront être mises en place de manière logique, avec de la cryptographie ou éventuellement des antennes très directionnelles. Mais le canal radio restera quoiqu'il en soit vulnérable à un brouillage massif (attaque de type denial of service). Dans les réseaux Ad hoc, non seulement les données sont vulnérables comme dans tout réseau radio, mais consécutivement au point précédent, il en est de même pour le trafic de contrôle et de gestion du routage [Karpijoki, 2000]. Les problématiques de la sécurité dans les réseaux Ad hoc sont donc très complexes, puisque l'on cherche à autoriser de nouveaux mobiles à participer au réseau, tout en évitant des noeuds "malins" qui détourneraient ou perturberaient le fonctionnement même du routage.
- ***La qualité de service :*** De nombreuses applications ont besoin de certaines garanties relatives par exemple au débit, au délai ou encore à la gigue. Dans ces réseaux Ad hoc, ces garanties sont très difficiles à obtenir. Ceci est du à la nature du canal radio d'une part (interférences et taux d'erreur élevés) et au fait que des "liens" entre des mobiles peuvent avoir à se partager les ressources (alors qu'en filaire, deux liens sont par définition indépendants). De ce fait, les protocoles de qualité de service habituels (par exemple IntServ / RSVP ou Diff-Serv) ne sont pas utilisables directement dans le monde Ad hoc et des solutions spécifiques doivent être proposées [Chaudet, 2002].

2.3 Les risques liés à la sécurité des réseaux Ad-hoc

2.3.1 L'Analyse de risque en sécurité

L'analyse de risque est nécessaire pour bien appréhender la problématique de la sécurité dans les réseaux sans fil Ad hoc. Elle suit les étapes suivantes :

1. Détermination des fonctions et données sensibles des réseaux Ad hoc à protéger.

2. Recherche des exigences de sécurité par le biais des critères de sécurité que sont l'authentification, l'intégrité, la confidentialité, l'anonymat et la disponibilité.

3. Étude des vulnérabilités.

4. Étude des menaces et quantification de leur probabilité d'occurrence ou de leur faisabilité.

5. Mesure du risque encouru en fonction des vulnérabilités mise en lumière et des menaces associées.

A partir de ces différents points d'entrée, il est possible de déterminer quelles sont les parties critiques, en termes de sécurité, que les concepteurs, les administrateurs, et les utilisateurs de réseaux sans fil Ad hoc doivent appréhender. La figure 2.3 retrace les différentes phases de ce processus :

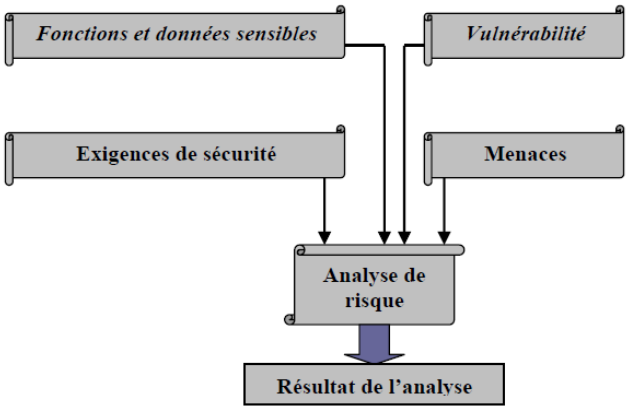


Figure 2-3 Les étapes de l'analyse de risque

Il faut noter qu'une généralisation des besoins en sécurité faisant abstraction des contextes d'utilisation a été nécessaire pour mener à bien cette analyse de risque. En effet, une application commerciale civile, par exemple, n'aura pas les mêmes contraintes qu'une application militaire. Un contexte militaire mettra en avant le fort besoin d'authentification, de furtivité et d'intégrité physique des éléments alors qu'une utilisation commerciale critique nécessitera de se focaliser sur la confidentialité des services. Selon les cas, il peut donc être indispensable d'étudier des solutions appropriées au contexte d'utilisation à travers une analyse approfondie prenant en compte des contraintes spécifiques.

2.3.2 Fonctions et données sensibles

Les fonctions sensibles des noeuds d'un réseau sans fil Ad hoc sont le routage, la configuration, la gestion d'énergie, et les mécanismes de sécurité. La plupart des données sensibles sont directement liées à ces fonctions puisqu'il s'agit :

- Des données relatives au routage (tables de routage et données de configuration des mécanismes de routage).
- Des mesures et données de configuration pour la gestion de l'énergie.

- Des données relatives à la sécurité (clés cryptographiques. mots de passe. Certificats, etc.).
- D'une manière générale tout ce qui concerne les données de configuration. Les informations personnelles des utilisateurs doivent aussi être considérées comme des données sensibles.

2.3.3 Exigences de sécurité des réseaux sans fil Ad hoc

Déterminer les exigences de sécurité d'un système nécessite d'appréhender l'ensemble des contraintes qui pèsent sur ce système. Cette étape permet par la suite de quantifier les critères de sécurité. Les spécificités des réseaux sans fil Ad hoc sont multiples et traitant de manière générale : les caractéristiques des noeuds, la gestion de l'énergie, les caractéristiques du réseau, les technologies sans fil sous-jacentes, la mobilité et la configuration.

2.3.3.1 Authentification / Intégrité / Confidentialité / Disponibilité

Coopérer au sein de tels réseaux présente un risque s'il n'y a aucun contrôle des participants. L'**authentification** des parties apparait donc comme la pierre angulaire d'un réseau sans fil Ad hoc sécurisé. En effet, comment assurer une quelconque confidentialité et intégrité des messages échangés si, dès le départ, on n'est pas sûr de communiquer avec la bonne entité.

Contrairement au réseau filaire, il n'est pas nécessaire de pénétrer dans un local physique pour accéder au réseau. Si l'authentification est mal gérée, un attaquant peut s'attacher au réseau sans fil et injecter des messages erronés.

L'**intégrité** des messages échangés est donc une exigence importante pour ces réseaux. L'intégrité des noeuds est, elle aussi, primordiale car les éléments d'un réseau Ad hoc sont moins sujets à surveillance. En effet, ils ne sont pas confinés dans un bureau mais transportés par leur propriétaire et peuvent donc être momentanément égarés. Un attaquant peut subtiliser un appareil, le corrompre avec un cheval de Troie par exemple, avant de le restituer discrètement à son propriétaire.

Une fois les parties authentifiées, la **confidentialité** reste un point important étant donné que les communications transitent via les airs et sont donc potentiellement accessibles à tout possesseur du récepteur adéquat.

La **disponibilité** est une propriété difficile à gérer dans les réseaux Ad hoc étant donné les contraintes qui pèsent sur ces réseaux sont :

- Topologie dynamique.
- Ressources limitées sur certains noeuds de transit.

- Communication sans fil pouvant être facilement brouillées ou perturbées. Les applications sans fil en mode Ad hoc ne devraient donc pas se focaliser sur ce critère.

2.3.3.2 Anonymat / Protection de la vie privée

Certaines applications peuvent nécessiter la discrétion sur l'identité des participants qui collaborent au réseau Ad hoc, par exemple un vote anonyme au cours d'une conférence. De plus, les différents gadgets électroniques qui formeront les noeuds des réseaux Ad hoc de demain, auront en toute probabilité, la possibilité de garder la trace de nos préférences afin de nous faciliter le quotidien et de nous offrir des services toujours plus appropriés. Cette tendance va pourtant à l'encontre de la protection de la vie privée de tout un chacun. Qui a envie de voir diffuser sur les ondes ses gouts et affinités ?

2.3.4 Vulnérabilités

La première vulnérabilité de ces réseaux est liée à la technologie sans fil sous-jacente. Quiconque possédant le récepteur adéquat peut potentiellement écouter ou perturber les messages échangés. Et ceci, même s'il se trouve dans un lieu public, à l'extérieur du bâtiment où se déroulent les échanges. Les noeuds eux-mêmes sont des points de vulnérabilités du réseau car un attaquant peut compromettre un élément laissé sans surveillance. L'absence d'infrastructure fixe pénalise l'ensemble du réseau dans la mesure où il faut faire abstraction de toute entité centrale de gestion pour l'accès aux ressources. Les mécanismes de routage sont d'autant plus critiques dans les réseaux Ad hoc que chaque entité participe à l'acheminement des paquets à travers le réseau. De plus, les messages de routage transitent sur les ondes radio.

2.3.5 Menaces

On distingue les menaces de type passif, où l'attaquant est limité à l'écoute et l'analyse du trafic échangé, et les menaces de type actif. Dans ce dernier mode, l'attaquant se donnera les moyens d'agir sur la gestion, la configuration et l'exploitation du réseau. Il peut injecter son propre trafic, modifier le fonctionnement d'un nœud, usurper l'identité d'un élément valide, rejouer des messages, modifier des messages transitant sur le réseau ou provoquer un déni de service. L'attaque passive prive le réseau de la confidentialité des messages échangés. Eventuellement, l'analyse du trafic représente un risque pour l'anonymat des participants et le respect de leur vie privée.

2.3.6 Résultat de l'Analyse de Risque

Après l'étude des besoins et exigences des réseaux sans fil Ad hoc en terme de sécurité, puis corrélation avec les risques issus des vulnérabilités et menaces s'appliquant à ces réseaux, nous avons pu dresser une liste des attaques fortement probables ou faisables et qui constituent un risque non négligeable en cas de réalisation.

Les dénis de services, *denial of services* (DoS), apparaissent comme les attaques les plus faciles à réaliser par un attaquant. La criticité de telles attaques dépend fortement du contexte d'utilisation mais n'est jamais complètement négligeable. Les modèles de dénis de services qui suivent se dégagent plus particulièrement dans le cas de réseau sans fil Ad hoc :

- Brouillage du canal radio pour empêcher toute communication.
- Tentative de débordement des tables de routages des noeuds servant de relais.
- Non-coopération d'un nœud au bon fonctionnement du réseau dans le but de préserver son énergie. "L'égoïsme d'un nœud est une notion propre aux réseaux Ad hoc". Un réseau Ad hoc s'appuie sur la collaboration sans condition de ses éléments. Un nœud refusant de jouer le jeu peut mettre en péril l'ensemble.
- Tentative de gaspillage de l'énergie de noeuds ayant une autonomie de batterie faible ou cherchant à rester autonome sans recharge le plus longtemps possible. Ces noeuds se caractérisent par leur propension à passer en mode veille le plus souvent possible. L'attaque consiste à faire en sorte que le nœud soit obligé de rester en état d'activité et ainsi de lui faire consommer toute son énergie. Cette attaque est référencée par Ross Anderson et Franck Stajano [Stajano, 2002], [Stajano, 1999] sous l'appellation *sleep deprivation torture attack*, un scénario de torture par privation du sommeil.
- Dispersion et suppression du trafic en jouant sur les mécanismes de routage.

Les attaques passives d'écoute et d'analyse du trafic constituent une menace certaine pour la confidentialité et l'anonymat.

L'usurpation de l'identité d'un nœud en leurrant les mécanismes de contrôle d'accès permet de nombreuses attaques actives rendant particulièrement critiques la protection des mécanismes de routage.

L'attaque physique d'un élément valide d'un réseau sans fil Ad hoc, entrainant la compromission du nœud, se révèle comme étant un point faible de ces réseaux. Enfin, il apparait clairement que les attaques sur les mécanismes de routage sont particulièrement critiques.

2.4 Le routage dans les réseaux Ad hoc

Les réseaux ad-hoc sont multi-sauts. Il peut donc arriver qu'un mobile veuille communiquer avec un autre qui n'est pas dans sa portée de communication directe. Les messages vont devoir être transmis de proche en proche jusqu'à la destination : c'est ce que l'on appelle le routage. La technique la plus basique est l'inondation, où chaque mobile réémet tous les paquets qu'il reçoit pour la première fois. Evidemment, l'inondation consomme beaucoup de ressources (bande passante et énergie) et n'est pas optimale. De nombreux protocoles de routage ont donc été proposés pour rendre les communications multisauts plus efficaces (moins de réémissions, chemins plus courts, etc.) que l'inondation basique.

Dans cette section, nous allons donc présenter certains des protocoles de routage développés dans le cadre du groupe de travail MANET de l'IETF. Ces protocoles travaillent au niveau IP et sont donc indépendants des couches physique et MAC. Le routage IP permet en particulier une inter-connectivité aisée avec toutes sortes d'autres réseaux ou matériels. Il est d'ailleurs possible d'utiliser ces protocoles pour fédérer en un seul réseau MANET des utilisateurs utilisant des matériels différents (cartes radios de technologies diverses, réseaux filaires, etc.). Les protocoles présentés sont parmi les plus représentatifs des diverses techniques utilisées pour le routage Ad hoc.

2.4.1 Routage hiérarchique ou plat

Les protocoles de routage pour les réseaux Ad hoc peuvent être classés suivant plusieurs critères. Le premier d'entre eux concerne le type de vision qu'ils ont du réseau et les rôles qu'ils accordent aux différents mobiles.

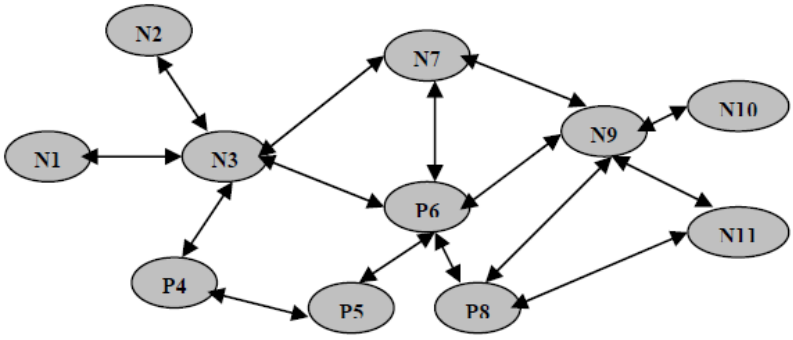


Figure 2-4 Routage à plat

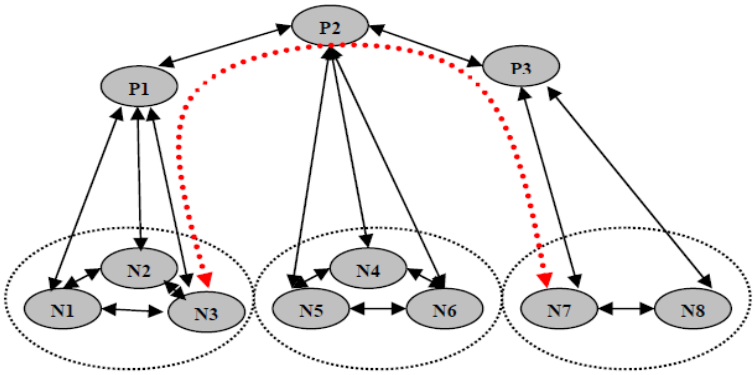


Figure 2-5 Routage hiérarchique

- Les protocoles de routage "à plat" considèrent que tous les noeuds sont égaux (figure 2.4). La décision d'un nœud de router des paquets pour un autre dépendra de sa position et pourra être remise en cause au cours du temps.

Les protocoles de routage hiérarchique fonctionnent en confiant aux mobiles des rôles qui varient de l'un à l'autre. Certains noeuds sont élus et assument des fonctions particulières qui conduisent à une vision en plusieurs niveaux de la topologie du réseau. Par exemple, un mobile pourra servir de passerelle pour un certain nombre de noeuds qui se seront attachés à lui. Le routage en sera simplifié, puisqu'il se fera de passerelle à passerelle, jusqu'à celle directement attachée au destinataire. Un exemple est donné sur la figure 2.5, où le nœud N3 passe par les passerelles P1, P2 et P3 pour atteindre N7. Dans ce type de protocole, les passerelles supportent la majeure partie de la charge du routage (les mobiles qui s'y rattachent savent que si le destinataire n'est pas dans leur voisinage direct, il suffit d'envoyer à la passerelle qui se débrouillera). Dans les réseaux où certains noeuds s'avèrent très sédentaires et disposent de suffisamment d'énergie (par exemple réseau d'ordinateurs portables mais où certains sont reliés au secteur, stations de base disposées là pour garantir la connectivité, etc.), ce type de routage présente certains avantages.

2.4.2 Etat de liens ou vecteur de distance

Une autre classification, héritée du monde filaire, est possible pour les protocoles de routage :

- **Les protocoles à état de lien :** Ils cherchent à maintenir dans chaque nœud une carte plus ou moins complète du réseau où figurent les noeuds et les liens les reliant. A partir de cette carte il est possible de construire les tables de routage. Un des avantages de ce type de protocole est leur capacité à pouvoir facilement trouver des routes alternatives lorsqu'un lien est rompu. Il est même possible d'utiliser simultanément plusieurs routes

vers une même destination, augmentant ainsi la répartition de la charge et la tolérance aux pannes dans le réseau. En contre partie, si le réseau est étendu, la quantité d'informations à stocker et diffuser peut devenir considérable.

- **Les protocoles à vecteur de distance :** Plutôt que de maintenir une carte complète du réseau (ce qui peut s'avérer extrêmement lourd), ces protocoles ne conservent que la liste des noeuds du réseau et l'identité du voisin par lequel passer pour atteindre la destination par le chemin le plus court.

2.4.3 Les différentes familles de protocoles de routage MANET :

Dans les travaux menés à l'IETF, plusieurs familles de protocoles se sont rapidement dégagées. Chaque protocole peut ainsi être classifié en tant que réactif, proactif, ou hybride.

2.4.3.1 Les protocoles réactifs :

Le principe des protocoles réactifs est de ne rien faire tant qu'une application ne demande pas explicitement d'envoyer un paquet vers un nœud distant. Cela permet d'économiser de la bande passante et de l'énergie. Lorsqu'un paquet doit être envoyé, le protocole de routage va rechercher un chemin jusqu'à la destination. Une fois ce chemin trouvé, il est inscrit dans la table de routage et peut être utilisé. En général, cette recherche se fait par inondation (un paquet de recherche de route est transmis de proche en proche dans tout ou partie du réseau).

L'avantage majeur de cette méthode est qu'elle ne génère du trafic de contrôle que lorsqu'il est nécessaire. Les principales contre parties sont que l'inondation est un mécanisme coûteux qui va faire intervenir tous les noeuds du réseau en très peu de temps et qu'il va y avoir un délai à l'établissement des routes.

2.4.3.2 Les protocoles proactifs :

Le principe de base des protocoles proactifs est de maintenir à jour les tables de routage, de sorte que lorsqu'une application désire envoyer un paquet à un autre mobile, une route soit immédiatement connue. Dans le contexte des réseaux Ad hoc les noeuds peuvent apparaître ou disparaître de manière aléatoire et la topologie même du réseau peut changer, cela signifie qu'il va falloir un échange continuel d'informations pour que chaque nœud ait une image à jour du réseau. Les tables sont donc maintenues grâce à des paquets de contrôle, et il est possible d'y trouver directement et à tout moment un chemin vers les destinations connues en fonctions de divers critères. On peut par exemple privilégier les routes comportant peu de sauts, celles qui offrent la meilleure bande passante, ou encore celles où le délai est le plus faible. L'avantage premier de ce type de protocole est d'avoir les routes immédiatement

disponibles quand les applications en ont besoin, mais cela se fait au coût d'échanges réguliers de messages (consommation de bande passante) qui ne sont certainement pas tous nécessaires (seules certaines routes seront utilisées par les applications en général).

2.4.3.3 Les protocoles hybrides :

Les protocoles hybrides combinent les approches réactive et proactive. Le principe est de connaître notre voisinage de manière proactive jusqu'à une certaine distance (par exemple trois ou quatre sauts), et si jamais une application cherche à envoyer quelque chose à un nœud qui n'est pas dans cette zone, d'effectuer une recherche réactive à l'extérieur. Avec ce système, on dispose immédiatement des routes dans notre voisinage proche, et lorsque la recherche doit être étendue plus loin, elle en est optimisée (un nœud qui reçoit un paquet de recherche de route réactive va tout de suite savoir si la destination est dans son propre voisinage. Si c'est le cas, il va pouvoir répondre, et sinon il va propager de manière optimisée la demande hors de sa zone proactive). Selon le type de trafic et les routes demandées, ce type de protocole hybride peut cependant combiner les désavantages des deux méthodes échange de paquets de contrôle réguliers et inondation de l'ensemble de réseau pour chercher une route vers un nœud éloigné.

2.4.4 Description de quelques protocoles de routage représentatifs :

Les protocoles décrits par la suite sont issus du groupe de travail MANET de l'IETF. Ces protocoles sont représentatifs de diverses techniques et sont les plus avancés sur la voie d'une normalisation.

2.4.4.1 AODV (Ad hoc On Demand Distance Vector):

AODV [Perkins, 2003] est un algorithme de routage à la demande, c'est-à-dire qu'il ne construise de routes entre noeuds que lorsqu'elles sont demandées par les noeuds sources, et ce pour réduire le nombre de diffusions de messages. AODV utilise les principes des numéros de séquence afin de maintenir la consistance des informations de routage. Les numéros de séquence permettent d'utiliser les routes les plus récentes. Il utilise une requête de route dans le but de créer un chemin vers une destination. La route peut ne pas exister si la destination n'est pas connue au préalable, ou si le chemin existant vers la destination a expiré ou il est devenu défaillant. Cependant, AODV maintient les chemins d'une façon distribuée en gardant une table de routage, au niveau de chaque nœud de transit appartenant au chemin cherché. Afin de maintenir des routes cohérentes, une transmission périodique du message "HELLO" est effectuée. Si au bout d'un certain temps aucun message "HELLO" n'est reçu à partir d'un nœud voisin, le lien en question est considéré défaillant. Le protocole AODV ne présente pas

de boucle de routage, et offre une convergence rapide quand la topologie du réseau Ad hoc change. Le protocole AODV est un protocole réactif, uniforme, de type distance vector.

2.4.4.2 DSR (Dynamic Source Routing Protocol) :

Le protocole DSR [Johnson, 2004] est basé sur le principe de diffusion à la demande pour calculer une route vers une destination. Il utilise un routage par la source, et se base principalement sur deux mécanismes coopératifs : la découverte de route et la maintenance de route. Il permet aussi l'existence de plusieurs routes vers la destination. A partir des informations de routage qui sont incluses dans les paquets de données, les noeuds appartenant à la route, ainsi que leurs noeuds voisins, peuvent les collecter et les mettre dans leurs caches pour une utilisation ultérieure. Chaque nœud dans le réseau envoyant ou relayant un paquet est responsable de confirmer son acheminement vers le prochain nœud en recevant un acquittement. Si un nœud détecte une cassure de route, un message d'erreur de route est retourné à la source. Lors de la réception d'un message d'erreur de route, la source supprime la route défaillante de son cache. Si un chemin alternatif est disponible, il peut être employé pour des données restantes à la destination, autrement, une nouvelle découverte de route est lancée. Comme AODV, DSR bufférise les paquets IP dans le nœud de source quand la découverte de route est effectuée. Ce protocole est un protocole réactif, uniforme, de type link state.

2.4.4.3 OLSR (Optimized Link State Protocol)

OLSR [Clausen, 2003] est un protocole de routage proactif. Il est considéré comme une optimisation du protocole à état des liens filaire pour les réseaux mobiles Ad hoc. Son innovation réside dans sa façon d'économiser les ressources radio lors des diffusions. Ceci est réalisé grâce à l'utilisation du concept des relais multi-points dans lequel chaque nœud choisit un sous ensemble de ses voisins qu'il appellera "MPR" (multi-point relais) pour retransmettre ses paquets en cas de diffusion. En se basant sur la diffusion sur les MPRs, tous les noeuds du réseau sont atteints avec un nombre réduit de répétitions.

Comme dans le paradigme proactif, des messages de contrôle périodiques doivent être utilisés pour le maintien des tables de routage et de voisinage. Dans OLSR, principalement deux types de messages sont introduits : "*Hello*" et "*TC*" (Topology Control). Périodiquement, chaque nœud diffuse localement un message Hello contenant des informations sur son voisinage et l'état des liens. Ceci permet à chaque nœud de prendre connaissance de son voisinage à un et deux sauts. L'ensemble MPR est alors construit dans chaque nœud de façon à contenir un sous-ensemble de voisins à un saut qui couvre tous les voisins à deux sauts. Afin

de construire les tables nécessaires au routage des paquets, chaque nœud génère périodiquement un paquet TC contenant la liste de ses voisins l'ayant choisi comme MPR. Le message TC est diffusé dans l'ensemble du réseau. Seuls les voisins MPR rediffusent un paquet TC reçu pour éviter l'inondation. Cette technique prometteuse réduit considérablement l'overhead généré par le trafic de contrôle. A la réception d'un message TC, la table de topologie peut être construite. Basé sur la table de topologie, chaque nœud peut calculer la table de routage qui permet d'acheminer les paquets vers n'importe quelle destination dans le réseau. OLSR est un protocole non uniforme, proactif de type link state.

2.4.4.4 TBRPF (Topology Dissemination Based on Reverse-Path Forwarding):

TBRPF [Ogier et al, 2002] est un protocole de routage proactif à état de lien. Chaque nœud exécutant TBRPF calcule un arbre de source fournissant des routes à tous les noeuds accessibles. Il se base sur l'information partielle de topologie stockée dans sa table de topologie, en utilisant une modification de l'algorithme de Dijkstra. Pour réduire l'overhead, chaque nœud rapporte seulement une partie de son arbre de source aux voisins. TBRPF emploie une combinaison de mises à jour périodiques et différentielles pour tenir tous les voisins au courant de la partie rapportée de son arbre de source. Chaque nœud a également l'option pour rapporter l'information additionnelle de topologie (jusqu'à la topologie complète), pour fournir la robustesse améliorée dans les réseaux fortement mobiles. TBRPF effectue la découverte de voisins en utilisant des messages HELLO différentiels, qui rapportent seulement des changements sur l'état des voisins. Par conséquent, les messages HELLO sont beaucoup plus petits que ceux utilisés dans d'autres protocoles de routage à état de lien tels que OSPF (Open Shortest Path First).

2.4.4.5 ZRP (Zone-Based Hierarchical Link State Routing Protocol):

ZRP [19] est un exemple de protocole hybride qui combine des approches proactives et réactives, essayant de ce fait de rassembler les avantages des deux approches. ZRP définit autour de chaque noeud une zone qui contient les noeuds voisins à un nombre donné de sauts du nœud. Des algorithmes proactifs et réactifs sont employés par le nœud pour acheminer les paquets, respectivement, dans et en dehors de la zone.

2.4.4.6 Autres protocoles :

De nombreux autres protocoles de routage ont été proposés pour les réseaux Ad hoc. [Royer, 1999] en décrit un certain nombre en sus de ceux déjà mentionnés. Dans la catégorie des protocoles construisant une topologie hiérarchique on peut citer *Cluster-head Gateway*

Switch Routing (CGSR) présenté dans [Chiang et al, 1997]. Certains autres protocoles nécessitent l'emploi de matériels externes. Par exemple Temporaly-Ordered Routing Algorithm (TORA) [Park, 1997] à besoin que les mobiles soient synchronisés. D'autres ([Giordano, 2001], [Camara, 2000]) utilisent le système GPS pour estimer la direction géographique de la destination et ne faire intervenir qu'une sous-partie du réseau dans la phase de construction des routes. Alors que beaucoup de protocoles cherchent à minimiser le nombre de sauts (minimum shortest path), certains protocoles enfin s'attachent à prendre d'autres critères en considération. ABR (*Associativity-Based Routing*) [Toh,1996] par exemple privilégie les liens les plus stables (mobiles qui restent longtemps dans le voisinage les uns des autres). SSR (Signal Stability Routing) [Dube, 1997] travaille à partir des informations de niveau de signal et [Kang, 2002] cherche à maximiser la durée de vie du réseau en agissant sur la puissance d'émission de chaque mobile séparément.

2.4.5 Le routage de paquets :

Afin de comprendre les attaques sur les protocoles de routage, il est nécessaire de comprendre leur fonctionnement global. Lorsqu'un nœud dans un réseau veut émettre un message vers un autre nœud, il regarde dans sa table de routage si une route existe pour ce noeud. Si elle n'existe pas, il initie une découverte de route, ***route discovery***, en diffusant sur le réseau, dans les airs pour les accès sans fil, un message de type ***route request***. Le message de route request contient l'adresse du nœud émetteur, l'adresse du nœud destinataire, un marqueur permettant d'identifier la découverte de route et une liste initialement vide à remplir par les noeuds intermédiaires. Lorsqu'un noeud intermédiaire reçoit ce paquet, s'il n'en est pas le destinataire et si sa table de routage n'indique pas de chemin pour le noeud recherché, il diffuse à son tour le paquet de type route request en rajoutant son adresse à la liste de nœuds intermédiaires. Dans le cas où le noeud intermédiaire possède dans sa table de routage un chemin pour le noeud destinataire, la majorité des protocoles prévoit que le nœud intermédiaire renvoie directement un message de type **route reply** à l'émetteur en indiquant ce chemin. Lorsqu'un paquet de requête atteint son destinataire, ce dernier émet un paquet de réponse du type route reply. Ce paquet transite par les noeuds intermédiaires de la liste. La figure 2.6 schématise l'évolution des messages lors de la découverte de route.

Lorsque la réponse atteint l'initiateur de la découverte de route, ce dernier met à jour sa table de routage avec cette nouvelle route, qui consiste en la liste des noeuds intermédiaires avec un cout associé. Le cout sert aux noeuds à effectuer un choix entre deux routes menant à la même destination. Il peut être basé sur le nombre de noeuds intermédiaires traversés ou sur des critères plus complexes comme le débit, la fiabilité des liaisons ou la taille des paquets. Si

l'initiateur reçoit ultérieurement une indication comme quoi cette destination peut être jointe avec un cout plus faible par un autre chemin, la table de routage sera mise à jour avec la route ayant le cout le plus faible. Une fois une route établie, un protocole de routage doit aussi mettre en œuvre un mécanisme de maintenance des routes pour gérer les évènements comme la coupure d'un lien entre deux noeuds par lesquels transitent des messages.

Lorsqu'un noeud reçoit un paquet de données pour une destination vers laquelle il ne peut plus émettre, il renvoie un message d'erreur de type **route error** vers la source du paquet de données. La route doit alors être supprimée de la table de routage. Des optimisations existent permettant à un noeud d'écouter les routes changées par les autres noeuds et de mettre à jour sa table de routage en conséquence.

2.4.6 Les Attaques Liées aux Protocoles de Routage :

Si aucun contrôle n'est fait sur la provenance et l'intégrité des messages de routage du réseau Ad hoc, un noeud malicieux pourra facilement causer des perturbations au réseau. Cela lui sera d'autant plus facile que les réseaux sans fil Ad hoc n'ont pas de barrière physique pour se protéger et que tous les éléments peuvent potentiellement participer au mécanisme de routage.

Si un noeud malicieux a la capacité d'usurper l'identité d'un noeud valide du réseau, il peut lors du mécanisme de découverte de route répondre au noeud initiateur avec un message de type route replay en annonçant un chemin avec un cout minimal, vers le noeud demandé. Le noeud émetteur mettra alors sa table de routage à jour avec cette fausse route. Les paquets de données du noeud émetteur vers le noeud destinataire transiteront par le noeud malicieux qui pourra tout simplement les ignorer. Cette attaque est appelée ***trou noir***, ***(black hole)*** Les paquets sont captés et absorbés par le noeud malicieux. La figure 2.7 illustre cette attaque.

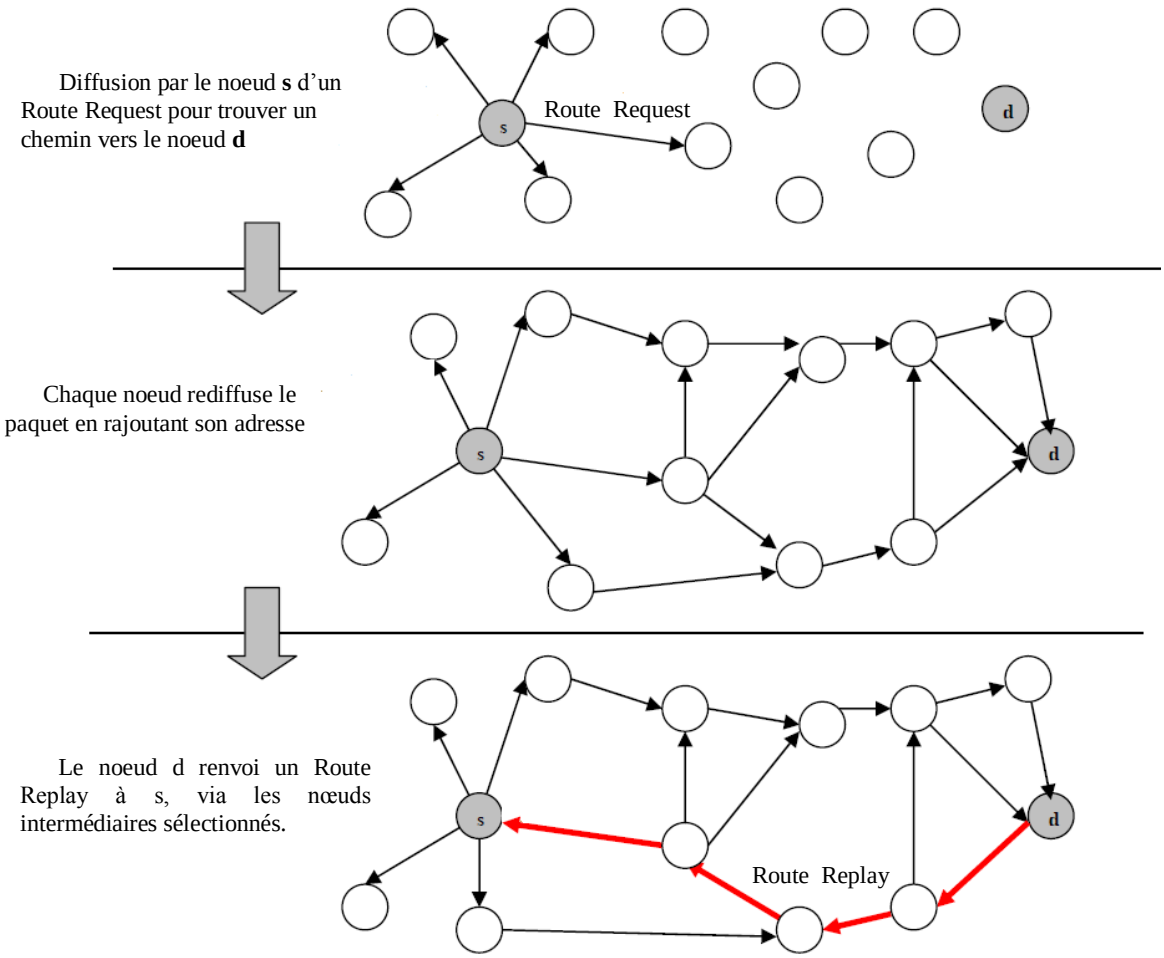


Figure 2-6 Découverte de route initiée par le protocole de routage

Une variante est appelée ***grey hole***, seuls certains types de paquets sont ignorés par le noeud malicieux. Par exemple, les paquets de données ne sont pas retransmis alors que les paquets de routage le sont. Un attaquant peut aussi créer des boucles infinies dans le réseau ou imposer aux paquets de faire des détours consommant la ressource radio inutilement. Un noeud malicieux ayant usurpé l'identité d'un noeud valide peut aussi générer des messages d'erreurs de type route error, de manière aléatoire, pour perturber le fonctionnement du mécanisme de maintenance des routes.

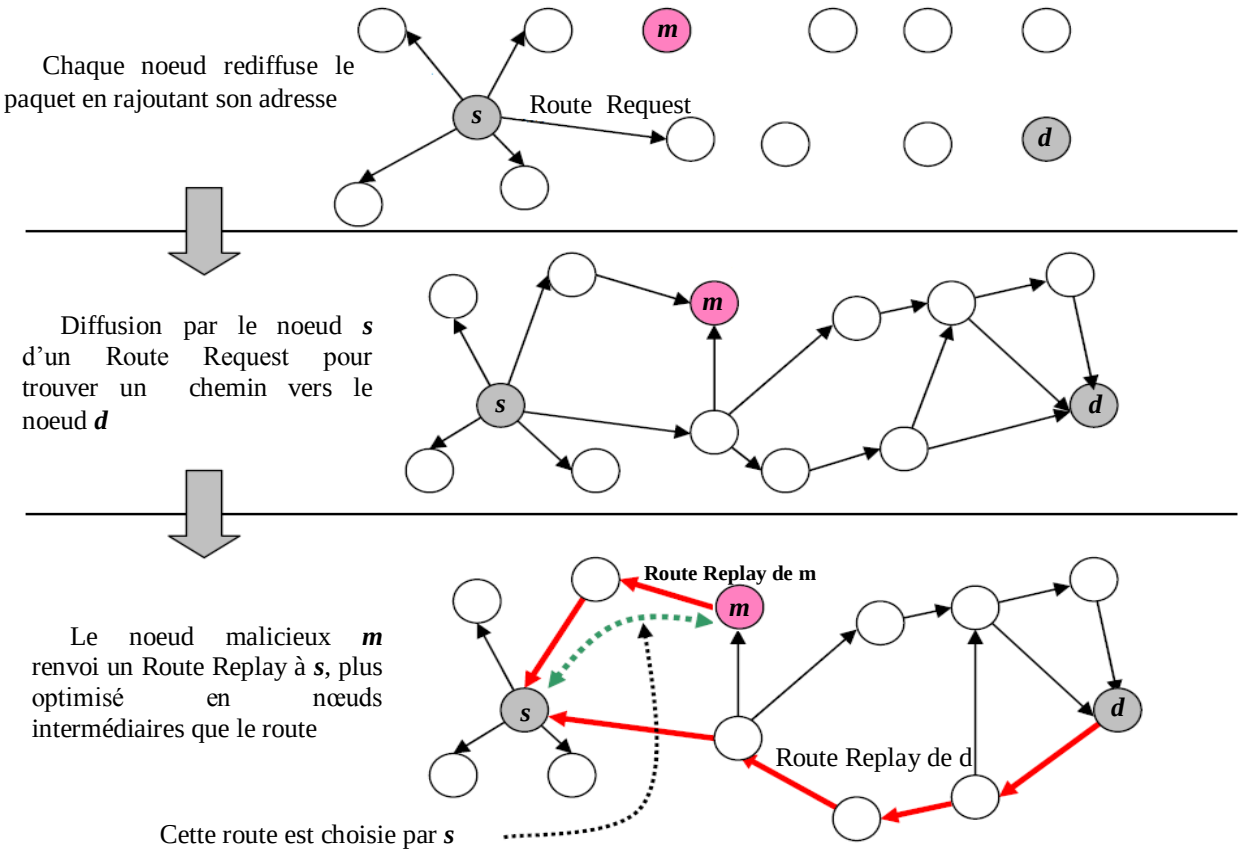


Figure 2-7 Attaque black hole

2.5 Conclusion :

Un réseau ad-hoc (MANET) se caractérise par sa simplicité ainsi que par sa facilité de déploiement en cas d'urgence ou de travaux temporaires. Mais de nouveaux problèmes apparaissent. En effet, l'absence d'une infrastructure centralisée rend les protocoles de communication dans les MANETs très complexes. Ainsi, même si les réseaux Ad hoc constituent une solution tout à fait prometteuse aux problèmes actuels liés à la mobilité des utilisateurs et des réseaux eux-mêmes, leur développement est freiné aujourd'hui par l'absence de mécanismes de sécurité suffisamment efficaces pour subvenir aux besoins actuels en protection des données tels que ceux des applications multimédia.

Après avoir étudié les réseaux ad hoc qui sont essentiellement caractérisés par l'absence d'infrastructure centralisée, la topologie dynamique, et par l'absence de sécurité, nous aborderons dans le prochain chapitre la problématique de déploiement du protocole de signalisation SIP dans ce type de réseaux en présentant les différentes approches proposées pour la décentralisation du SIP, ainsi que les mécanismes de sécurité utilisés dans ces approches.

Chapitre 3 : Etat de l'art
Décentralisation du protocole SIP dans les réseaux Ad-hoc

3.1 Introduction

Dans les deux chapitres précédents on a étudié le protocole de signalisation SIP, et les réseaux ad hoc. SIP s'appuie sur un réseau d'entités centralisées pour aider les terminaux dans leurs opérations. Les principales fonctions de SIP sont l'enregistrement et la localisation des multiples participants dans une session. L'absence d'infrastructure dans les réseaux ad hoc, nécessite la réalisation de ces fonctions par les terminaux du réseau ad hoc, tout en prenant en considération les contraintes de ce type de réseaux sans fil, qui sont la topologie dynamique, la bande passante limitée, les faibles ressources en énergie et en capacité de calcul, et la sécurité physique limitée. Dans ce chapitre nous présentons et nous évaluons les travaux de décentralisation du protocole SIP, dans les réseaux ad hoc.

3.2 Problèmes de décentralisation du SIP dans les réseaux Ad hoc

Pour établir de façon efficace une communication entre deux ou plusieurs parties, une signalisation adéquate est nécessaire pour négocier les paramètres et les caractéristiques des médias utilisés dans cette communication. Le fonctionnement de la signalisation pour l'établissement de sessions multimédias est également appelé gestion de session. Néanmoins, actuellement, tous les protocoles de gestion de sessions nécessitent la présence d'une infrastructure de réseau préexistant, car la plupart des opérations sont traitées par des entités centralisées.

Afin de supporter la vision proposée et le besoin d'une future communication, il est nécessaire de déployer les systèmes de gestion de sessions d'une manière distribuée. De cette façon, les échanges de médias basés sur IP peuvent également être possibles dans les environnements ad-hoc, ou en général, dans les environnements où aucun support d'infrastructure n'est disponible ou nécessaire. En outre, pour permettre une communication entièrement omniprésente, il est nécessaire que les applications et les services modifiés pour l'usage dans les réseaux ad-hoc, soient interopérables avec les applications existantes dans le monde d'internet.

Dans l'architecture du SIP deux éléments logiques jouent un rôle clé, les serveurs d'enregistrement et les serveurs proxys. Ils sont souvent regroupés dans le même serveur. De plus le protocole SIP suppose que chaque noeud du réseau connaisse l'adresse de son proxy responsable. Dans une telle architecture, les agents utilisateurs s'enregistrent auprès de leurs proxys en leur envoyant leurs adresses SIP d'enregistrements (voir section 5 du chapitre 1). Une fois enregistré chaque noeud voulant communiquer avec d'autres noeuds, exprime sa

volonté de communiquer auprès de son proxy. Il envoi alors sa demande au proxy qui se charge de localiser et transmettre cette demande à l'interlocuteur.

Le scénario décrit ci-dessus n'est manifestement pas applicable dans les réseaux ad hoc, plusieurs problèmes entravent l'utilisation du protocole SIP dans les MANETs, un problème majeur est que ces environnements n'acceptent pas des serveurs et des éléments fixes pour diverses raisons. Que se passe-t-il lorsque le serveur se déplace hors de portée ou lorsqu'il n'est plus joignable avec le reste du réseau? Ces situations peuvent se produire souvent dans les MANET car les nœuds mobiles se caractérisent par une topologie dynamique, et cela peuvent influencer le fonctionnement normal du réseau. Par conséquent, les serveurs centralisés doivent être évitées. De toute façon la plupart des services connus dans les réseaux d'infrastructure ne s'appliquent pas dans les MANETs, car elles s'appuient sur des serveurs centralisés.

Un autre problème qui peut entraver la décentralisation de SIP, c'est les ressources limités des nœuds mobile, ces derniers communiquent entre eux en partageant un médium radio, par conséquent la bande passante qu'utilise un noeud est modeste. Les mécanismes de diffusion de messages consomment de la bande passante et beaucoup d'énergie, alors que l'autonomie des noeuds est très réduite. La décentralisation du service de localisation et d'enregistrement impose que chaque noeud stocke les correspondances entre les adresses des noeuds et leurs identifiants SIP. Cependant, les faibles capacités de stockage et de calcul des nœuds présentent une importante contrainte pour décentraliser le service de localisation.

Le protocole SIP d'un autre coté utilise le DNS (Domain Name System) pour obtenir des informations sur le serveur proxy SIP qu'est responsable à un nom de domaine donné. Ceci est impossible dans les MANETs, car en général, aucun service DNS n'est disponible. Ainsi, l'absence de DNS pose un autre problème qui empêche le déploiement de SIP dans les MANETs. Deux approches ont été déployées pour réaliser la fonctionnalité du DNS dans les MANETs : la première consiste à imposer une certaine topologie au réseau en assignant les tâches de localisation à certains noeuds du réseau [Stuedi, 2007]. Mais cette solution présente beaucoup de problèmes car il est très difficile de maintenir cette topologie dans les réseaux mobiles. La deuxième approche consiste à former des super-noeuds (clusterheads) qui joueront le rôle du DNS dans les MANETs [Engelstad and Zheng, 2005]. L'élection et le maintien des super-nœuds présentent un sérieux problème de ce genre d'approches.

Tous les problèmes mentionnés ci-dessus, conduit à un autre problème, qui est le problème de la sécurité qui survient fortement, et qui constitue le principal objectif de ce mémoire. Les

réseaux mobiles ad hoc sont plus touchés par le paramètre de sécurité, que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé, ainsi que tout travail qui ne prend pas en considération ce problème, il ne sera pas exploitable dans des milieux comme celui du militaire.

3.3 Les différents travaux de décentralisation de SIP

Dans tous les travaux étudiés on a trouvé que les auteurs ont choisi le protocole de signalisation SIP, pour le déployer dans les réseaux ad hoc, L'idée principale commune entre eux est la réutilisation des protocoles existants, ainsi de modifier la sémantique de ces protocoles selon les besoin, au lieu de créer un nouveau protocole de signalisation propre aux réseaux ad hoc, et cela pour des raisons de compatibilité avec les applications déjà existantes.

Plusieurs travaux ont été proposées pour déployer SIP dans les réseaux ad hoc, nous les avons classifiées en trois classes (figure 3.1). La première classe de ces travaux s'inspire des réseaux P2P (Peer to Peer), puisque la découverte des pairs dans les réseaux P2P est sémantiquement similaire à la découverte des terminaux SIP dans les MANETs [Borg, 2003]. Les deux réseaux P2P et ad hoc ont des caractéristiques similaires qui sont :

- l'absence des pairs qui agit comme des serveurs.
- Les principaux enjeux dans ces deux réseaux consistent à savoir comment trouver efficacement les données ou les routes demandées.
- Leur topologie change fréquemment.

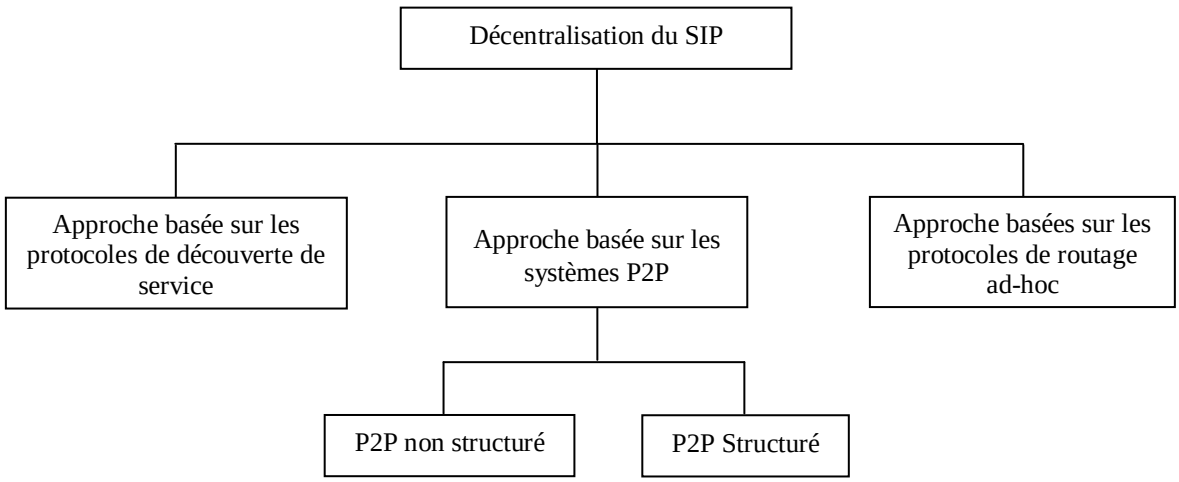


Figure 3-1 Classification des travaux de décentralisation de SIP

Selon leur topologie qu'ils utilisent, les systèmes P2P se divisent en deux sous classes, P2P pur ou bien non structuré, et le P2P structuré (DHT Distribued hashed table). Dans la première sous classe tous les pairs sont à la fois client et serveur. Et chaque pair peut communiquer directement avec l'ensemble de ses voisins par l'intermédiaire de pairs notoirement connus ou par une requête broadcastée sur le réseau pour trouver les pairs déjà connectés. Dans l'autre sous classe, les pairs s'organisent selon un sens ou une structure logique, par exemple un anneau, un cercle ou une grille.

Dans la deuxième classe de ces travaux les auteurs ont essayé d'intégrer la découverte des terminaux SIP avec un protocole de découverte de services, du fait qu'une similarité qui existe entre le problème de découverte des terminaux SIP dans les MANETs, et celui de découverte de services ou ressources dans les environnements distribués, comme les réseaux ad hoc, les réseaux d'agents mobiles, et les grilles. La découverte de services est définie pour résoudre le problème de localisation automatique des différents services dans un réseau. Les services sont des entités offertes à l'utilisateur par les nœuds du réseau. A titre d'exemple, le service peut être une imprimante, un scanneur d'images ou un logiciel de comptabilité.

La troisième classe des travaux est basée sur les protocoles de routage de la couche réseaux ad-hoc. Dans cette classe la découverte de terminaux SIP est découplée du protocole de routage ou elle est intégrée avec un protocole de routage comme les protocoles qui utilisent les topologies virtuelles sous-jacents basées sur le principe de regroupement.

Dans ce qui suit nous détaillerons les différentes classes mentionnées ci-dessus, et nous présenterons des exemples pour chaque classe pour bien comprendre les principes de chacune de ces classes.

3.3.1 Travaux basés sur les systèmes P2P :

3.3.1.1 P2P non structuré

Dans ce type de système, les noeuds sont disposés de manière aléatoire et chaque noeud localise et se connecte à n'importe quel pair disponible dans le réseau sans aucun mécanisme de sélection des pairs auxquels il devrait se connecter. Tous les participants ont le même statut. Pour rejoindre le réseau, un participant P doit d'abord diffuser un message spécial sur le réseau physique qui contient son URI SIP et son adresse IP. Les pairs directement proches de P et qui reçoivent ce message renvoient leurs liaisons (URI SIP avec l'adresse IP). Vu qu'il n'y a pas de serveurs, les requêtes de P seront diffusées à ces voisins et ses derniers les diffusent à leurs voisins immédiats et ainsi de suite, un tel arrangement est facile à gérer et à maintenir. Par contre ce genre de système rend la recherche très difficile puisque sa structure

est aléatoire. Le travail le plus important qui se situe dans ce type c'est le travail de Leggio [Leggio et al, 2005]. Leggio a proposé trois solutions pour décentraliser le SIP. Nous abordons ces solutions dans la section suivante.

3.3.1.1.1 Decentralized SIP (dSIP) :

Dans dSIP [Leggio et al, 2005] les auteurs ont proposé une solution qui permet aux nœuds des réseaux ad hoc d'utiliser les fonctions SIP de manière décentralisée. L'idée est d'intégrer un ensemble de fonctions de base d'un serveur proxy et d'un serveur d'enregistrement dans chaque nœud de réseau ad hoc, et construire une architecture logicielle qui peut contrôler toutes les opérations. L'architecture logicielle du Decentralized SIP est composée de plusieurs modules, la figure 3.2 représente la structure et les connexions entre ces modules.

- ***La bibliothèque SIP***

  La plus basse couche de l'architecture fournit les fonctions de base de SIP, telles que l'analyse de message et le contrôle de syntaxe. Au sommet de la bibliothèque, le module User Agent et le module serveur.

- ***Le module User Agent SIP (UA)***

Réalise les fonctions de base qu'un UA (User Agent) doit faire, comme la construction des messages SIP.

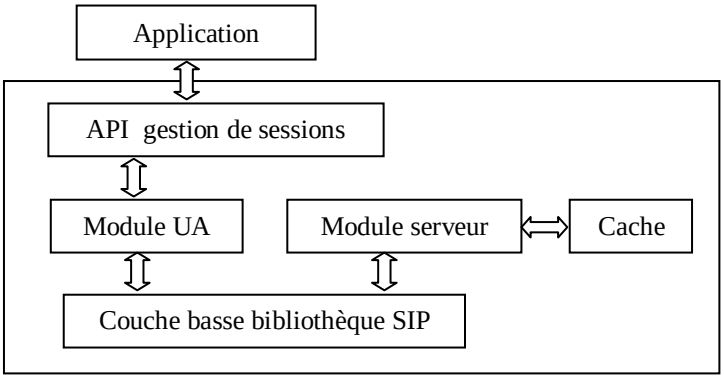


Figure 3-2 L'architecture logicielle du Decentralized SIP

- ***Le module serveur SIP***

Ce module implémente les fonctions des serveurs SIP, les opérations de manipulation sont typique aux serveurs proxy et serveur d'enregistrement.

Les trois modules discuter dessus, constituent la base de l'architecture Decentralized SIP, le seul module qui a été ajouté et modifié pour décentralisée les opérations est le module serveur et son cache, qui contient les liaisons des utilisateurs actuellement présents dans le

réseau ad-hoc. Le module serveur SIP interagit avec son cache, qui est le correspondant logique de service de localisation SIP. Le module UA et le serveur sont des modules indépendants et communiquent par sockets UDP.

- ***L'API (Application Programming Interface) gestion de session (SM)***

Fournit des méthodes pour initier, modifier et terminer les sessions SIP, ainsi que toutes les modifications nécessaires aux messages Decentralized SIP. L'API (SM) décide quelle fonctionnalité doit être utilisée ad hoc, ou centralisé. Dans le premier cas, ou le nœud se trouve dans un réseau ad hoc, l'API passe au module UA l'adresse de « loopback » comme adresse IP du serveur d'enregistrement. Le message REGISTER sera transmis vers le serveur local, qui va le transmettre en broadcast. Tandis que dans le deuxième cas l'API SM passe au module UA l'adresse IP du serveur d'enregistrement externe préconfiguré, dans ce cas le message REGISTER sera envoyé sans la participation du module serveur local. Seulement le module user agent est utilisé, ce qui garantit une totale compatibilité avec le standard SIP. L'approche dSIP permet également à un utilisateur d'établir des sessions SIP avec des nœuds à l'intérieur et l'extérieur du réseau ad hoc en même temps.

A) L'enregistrement dans le réseau :

Pour qu'un nœud s'enregistre dans le réseau, il communique sa présence à tous les autres nœuds. L'enregistrement distribué se fait par l'envoi d'un message SIP REGISTER en broadcast qui contient l'URI SIP de ce nœud associé à son adresse IP (liaison). Ce message est traité par les modules serveur des nœuds qui le reçoivent. La liaison de l'utilisateur qui vient s'enregistrer est stockée, et un message SIP 200 OK contenant la liaison de l'utilisateur recevant est retourné au nœud émettrice, Les nœuds récepteurs du REGISTER enregistrent la liaison de l'utilisateur dans leur cache. Les entrées dans le cache ont une durée de validité limitée, mis en place par un champ d'en-tête Expires qui peut être contenu dans le message REGISTER. Les nœuds, dont leur enregistrement est expiré, il devrait les actualiser en envoyant une nouvelle requête de diffusion REGISTER, Les récepteurs ne doivent pas répondre aux requêtes REGISTER d'actualisation. La figure 3.3 représente le scénario d'enregistrement de trois nœuds A, B et C.

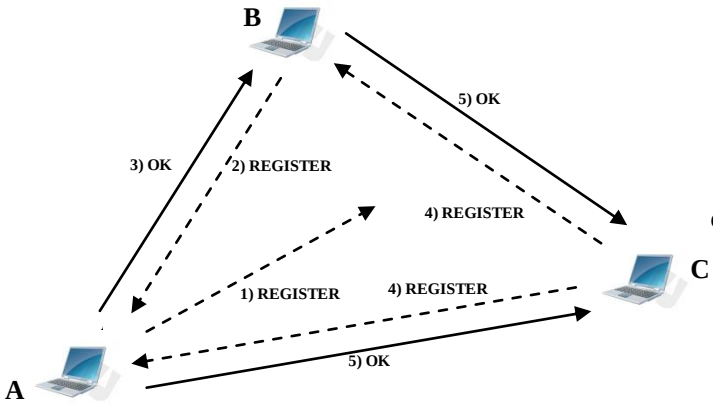


Figure 3-3 La registration dans dSIP

Lorsque le noeud C veut adhérer au MANET, il diffuse un message *REGISTER* contenant son adresse IP vers tous les noeuds du réseau. Chaque noeud qui reçoit ce message, lui répond par un *200 OK* et lui transmet ses informations de contact. Grâce à cette procédure, chaque noeud du réseau met à jour son cache.

Etablissement de session

Quand un utilisateur décide d'inviter un autre à une session dSIP, un message INVITE est généré par le module UA de l'appelant selon les règles de base de SIP. Les différents messages échangés entre les deux nœuds sont illustrés dans la figure (figure 3.4).

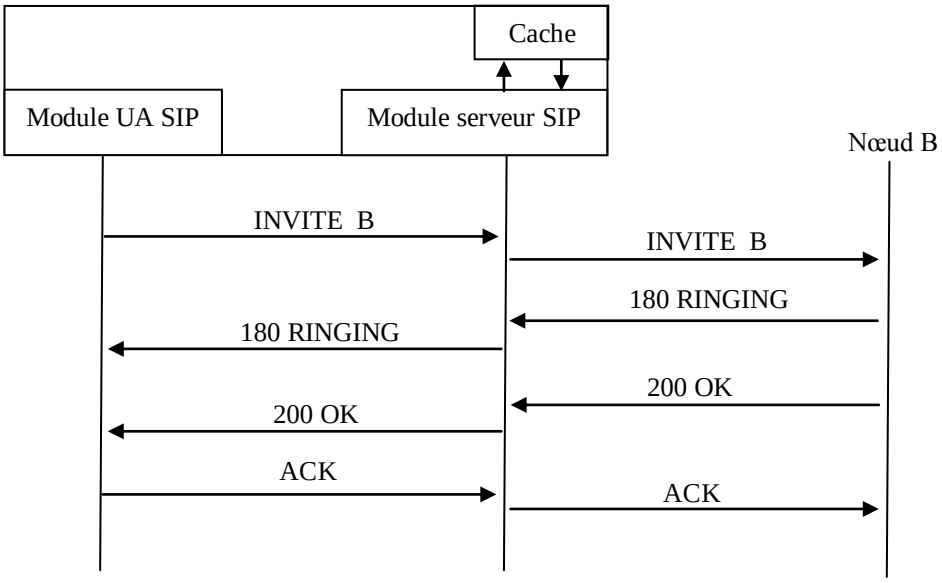


Figure 3-4 Les différents messages échangés pendant l'établissement de session

Toutes les demandes des utilisateurs dans le cas ad-hoc, sont transmises par le module UA au module serveur local. Le module proxy local reçoit le message, regarde dans le cache la

liaison pour l'URI spécifiée, et envoie le message INVITE à l'utilisateur désiré et ainsi la session SIP est crée.

Si la liaison de la cible URI demandée n'est existe pas dans le cache (figure-3.5), du fait qu'elle est expirée, le serveur diffuse un message REGISTER d'actualisation, en précisant l'URI SIP de l'utilisateur concerné dans l'URI de la requête REGISTER. Seul le nœud spécifié, va répondre au message d'actualisation, s'il est présent dans le réseau.

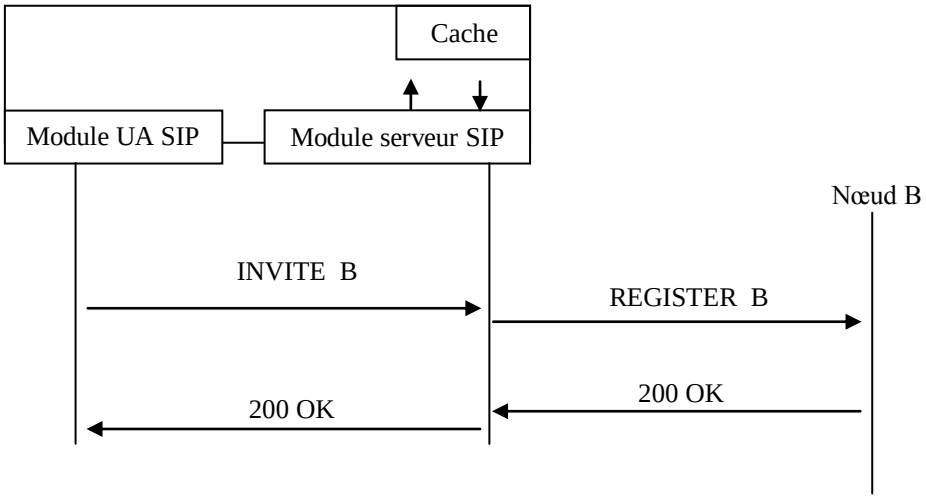


Figure 3-5 Enregistrement dans dSIP

La solution dSIP (Decentralized SIP) est conçue pour des réseaux de petite taille. La principale limitation de dSIP est que chaque nœud doit stocker la liaison de tous les autres nœuds pour avoir une vue complète de tous les utilisateurs du réseau. Cette hypothèse doit être révisée lors de l'utilisation de dSIP dans des réseaux de grande taille qui peuvent avoir des centaines de nœuds. Il n'est pas raisonnable de concevoir le mécanisme de découverte des terminaux de sorte que chaque nœud arrivant devrait recevoir un message de réponse de tous les autres nœuds dans le réseau. L'algorithme de routage utilisé dans dSIP provoque le problème de tempête de diffusion. Une solution à ce problème a été proposée par Leggio par l'utilisation d'une version modifiée de l'algorithme PCache [Miranda et al, 2006], [Baldoni et al, 2006] où les nœuds diffusent et reçoivent les messages par des mécanismes probabilistes et déterministes.

3.3.1.1.2 SIPCache

SIPCache [Leggio et al, 2006] c'est une solution qui combine le dSIP avec l'algorithme PCache [Miranda et al, 2006], [Baldoni et al, 2006], et qui forme un service de localisation fonctionnelle dans les MANETs, aussi elle optimise les opérations de dSIP et l'étend pour qu'il fonctionne dans les réseaux ad-hoc à multi-hop. SIPCache permet la distribution des

liaisons entre tous les nœuds du MANET, et élimine la nécessité de stockage de toutes les liaisons par tous les nœuds.

A) L'algorithme PCache

PCache [Miranda et al, 2006], [Baldoni et al, 2006] c'est un algorithme de gestion des informations dans les réseaux ad-hoc, y compris la diffusion des données et la mise en cache efficace. L'algorithme fournit deux opérations distinctes: la diffusion et la récupération d'objets de données (data item) mis en cache. L'implémentation de ces opérations est orchestrée de telle sorte qu'un nombre limité de messages est nécessaire pour récupérer tous les objets de données dans le réseau. Le but de PCache est de fournir une répartition adéquate des objets de données de sorte que chaque noeud est capable de trouver une proportion importante de la totalité des objets dans son cache ou dans les caches des voisins à un saut.

1) Objet de données

On distingue deux type d'objets dans le cache d'un nœud, des objets propriétaires (owned), qui appartiennent à ce nœud et que ce nœud les annoncent, et les objets complémentaires (remote), qui sont des objets annoncés par d'autres nœuds.

2) Structure du cache

Chaque nœud dans le système PCache dispose d'un cache de taille limitée et prédéfinie, qui stocke une partie d'objets de données annoncés par tous les noeuds. Chaque objet de données est composé d'une clé, une valeur, une durée d'expiration et un numéro de version. La figure 3.6 représente la structure d'un objet dans le cache.

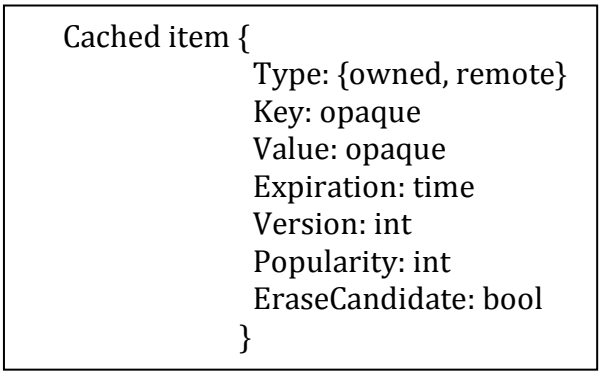


Figure 3-6 Structure d'un objet PCache dans le cache

3) Contenu du message PCache

Les messages PCache partagent un champ d'en-tête commune, qui décrit le type de message (diffusion, requête ou réponse), un délai de vie (TTL, Time To Live), qui se décrémente en traversant chaque nœud. Le message contient aussi des informations supplémentaires concernant les éléments qu'il porte et leur relation avec l'état du cache des autres nœuds. La figure 3.7 représente un message PCache.

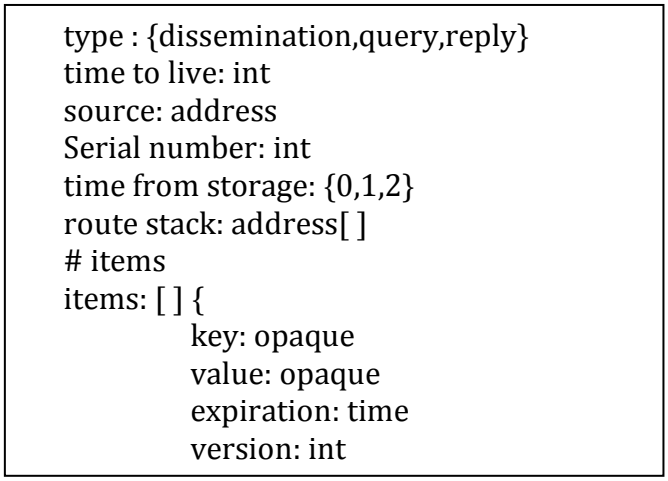


Figure 3-7 Structure d'un message PCache

Le champ d'entête « Source » contient l'adresse du nœud initiateur du message. Et le « Serial number » est utilisé pour identifier de manière unique un message PCache. Pour identifier les doublons, les nœuds conservent un enregistrement des messages reçus récemment.

4) L'algorithme de diffusion PCache

Cet algorithme utilise à la fois un mécanisme pour limiter le nombre de retransmissions pendant l'inondation de messages. (Semblables à ceux dans [Haas et al, 2002]), et un protocole qui optimise le flooding de messages en fonction de la puissance de signal des messages reçus [Levis et al, 2004].

5) Processus de diffusion (process of dissemination)

La diffusion d'un nouveau objet se fait par l'initiation d'un message de diffusion, dans ce message le champ d'en-tête « Time From Storage » indique la distance (en nombre de saut) de l'expéditeur au plus proche nœud connu qui stocke l'objet indiqué, Par conséquent, le

nœud source remet le champ TFS à zéro pour indiquer que les objets sont stockés dans sa mémoire cache locale.

Chaque noeud recevant un message de diffusion le met en attente pendant une période de temps proportionnelle à la puissance de réception. Au cours de la période d'attente, le noeud compte le nombre de retransmissions écouté et calcule « mintfs », qui est la plus petite valeur des « Time From Storage » de toutes les tentatives du message original. À la fin de la période d'expiration, « mintfs » indiquera la distance en sauts de la source à la plus proche nœud (s) qui a enregistré une copie de l'objet. Lorsque la période de maintien expire, le nœud utilise le nombre de retransmissions écouté, « mintfs » et un générateur de nombre aléatoire pour décider l'une des trois actions suivantes:

a) Si le nœud est à l'écoute de deux ou plusieurs retransmissions et la valeur de « mintfs » est 0 ou 1, il ignore le message.
b) Si le premier critère n'a pas été appliqué, le nœud stocke l'objet de données dans le cache et rediffuse le message. l'action s'exécute avec une probabilité égale à : $e^{mintfs\ -2}$.
c) Si les deux conditions précédentes n'ont pas été atteintes, le message sera rediffusé, mais les données ne seront pas stockées dans le cache.

6) Processus de découverte (Retrieval Process)

La découverte se fait, par l'envoie d'un message de découverte d'un objet (Request for a data). Un nœud recevant ce message commence par la recherche de la clé dans son cache local. Si l'objet n'existe pas, le noeud prépare un autre message de découverte, en mettant la clé dans le message, Le message est d'abord diffusé avec TTL égale à un. Si aucune réponse n'est reçue dans une certaine limite de temps prédéfini, la requête sera modifiée par l'augmentation de la valeur de TTL. Le protocole impose une limite sur le nombre de tentatives qui s'exécutent.

Un nœud recevant un message de découverte d'objet, et qui ne trouve pas la valeur dans son cache local il exécute l'algorithme de diffusion décrit précédemment. Si un nœud reçoit un message de découverte et trouve la clé demandée dans son cache, il envoie une réponse en unicast à la source de message. Le message de réponse (Query Reply) est envoyé après un délai aléatoire, pour empêcher la collision des réponses multiples.

B) Le mappage de message dSIP aux messages PCache

L'interaction entre SIP et PCache a été réalisée par le mappage direct de messages SIP dans le format défini pour PCache. Les messages SIP sont distribués dans le réseau conformément à la procédure définie par l'algorithme PCache, et leurs champs d'en-tête sont traités en tant que des champs d'en-tête PCache. Un message de diffusion PCache est mappé dans un message SIP REGISTER. La correspondance entre SIP et les messages PCache est montré dans le tableau 3.1.

Tableau 3-1 Le mappage de message SIP-PCache

	SIP	**PCache**
Type de messages	REGISTER INVITE 200 OK	Dissemination Query Query Reply
Nom de champs d'entête	Max-Forwards From Call-ID Cseq Via Content-Length Body	TTL Source Serial Number Time from Storage Route Stack # Items Items

1) L'enregistrement dans SIPCache

L'enregistrement SIPCache se fait dans le cas d'un nouveau nœud qui rejoint le réseau, ou dans le cas où un nœud actualise sa liaison afin d'éviter l'expiration. Les procédures suivies par les nœuds correspondent aux étapes suivies lors de la diffusion d'un message PCache. Les mises à jour des caches également suivent les règles de l'algorithme PCache. L'objectif du processus d'enregistrement SIPCache, est la diffusion efficace de toutes les liaisons des utilisateurs dans le MANET, on note que l'algorithme PCache ne cherche pas à assurer que chaque nœud reçoit une copie de tous les liaisons, mais plutôt, la diffusion des messages est réalisée par une approche combinée probabiliste et déterministe.

Selon l'algorithme PCache, les messages d'enregistrement SIPCache transporte la liaison de l'utilisateur qui vient s'inscrit dans le réseau, et dans le corps des messages, les liaisons complémentaires qui correspondent à d'autres nœuds.

2) ***Etablissement de session SIPCache***

L'établissement de session se fait par l'envoie d'une requête SIPCache (SIPCache Request). Une requête SIPCache permet à un nœud de trouver l'adresse IP qui correspond à un URI SIP recherché. Les considérations présentées pour le message REGISTER sont

valides également au message Request SIPCache. La requête SIPCache porte aussi dans son corps les liaisons complémentaires, ce qui permet la répartition équitable de toutes les liaisons dans le réseau. Le message de réponse peut être envoyé par l'utilisateur recherché, ou par n'importe quel utilisateur qui a une liaison non-expirée pour l'utilisateur recherché. Dans ce dernier cas, la liaison de l'utilisateur qui répond à cette requête est ajoutée comme l'un des objets complémentaires dans le message de réponse.

C) La sécurité dans dSIP et SIPCache

Les schémas dSIP et SIPCache fournissent des mécanismes de sécurité, dans les deux phases d'enregistrement et d'établissement de session. Ils reposent sur le mécanisme SIP Identity [Peterson and Jennings, 2006] pour garantir l'authentification et l'intégrité des messages, L'idée clé de ce mécanisme, est que chaque entité utilise le certificat de son domaine au lieu de son propre certificat, pour signer les messages. Ce mécanisme a été modifié pour qu'il s'adapte à l'environnement ad-hoc.

Le mécanisme modifié, suppose la pré-distribution et la vérification préalable des certificats, et impose que chaque nœud signe ses messages par son propre certificat. Ce mécanisme protège l'identité des utilisateurs qui partagent leurs certificats avant d'avoir connecté au réseau. A titre d'exemple, deux collègues pour qu'ils puissent communiquer dans l'environnement dSIP, ils doivent au préalable échanger leurs certificats et les stocker dans leurs dispositifs. Ce mécanisme assure l'authentification et l'intégrité de tous les messages échangés dans le dSIP et SIPCache tel que, les réponses SIP, INVITE, et REGISTER. L'authenticité des messages REGISTER dSIP est très importante, ce qui donne aux destinataires d'un message REGISTER la possibilité de vérifier l'identité de l'utilisateur qui vient s'inscrit dans le réseau. On note que le dSIP et le SIPCache n'utilisent aucun mécanisme pour assurer la confidentialité des messages et l'anonymat des utilisateurs.

D) **Discussion sur dSIP et SIP Cache**

SIPCache repose sur PCache [Leggio et al, 2006], qui est un algorithme destiné à réduire la distribution et la récupération des objets de données stockées dans les mémoires caches. L'application de PCache à dSIP, permet de distribuer les informations de contacts, ou les liaisons des utilisateurs SIP dans le réseau, de sorte que n'importe quel nœud est capable de récupérer l'information d'un utilisateur recherché. En fusionnant les fonctionnalités de dSIP avec PCache, SIPCache prolonge le cadre d'utilisation de dSIP qui est optimisé pour les petits réseaux ad-hoc, vers des réseaux à grande échelle. L'inconvénient de SIPCache est qu'il utilise un algorithme probabiliste et complexe d'un point de vue de réalisation et

d'implémentation. Le dSIP et SIPCache reposent sur le mécanisme SIP Identity pour assurer l'authentification et l'intégrité et n'emploie aucun mécanisme pour la confidentialité.

3.3.1.2 P2P Structuré (distributed hash table)

Dans les systèmes P2P structurés, au dessus du réseau physique sous-jacent, les nœuds sont reliés par un réseau recouvrant construit sous certaines contraintes, répondant à plusieurs propriétés et connectant les noeuds selon une structure particulière donnée (Anneau (Chord) [Stoica et al, 2001], ou Espace de coordonnées cartésiennes (CAN) [Ratnasamy et al, 2001]).

La DHT (distributed hash table) est une propriété particulière déployée dans les systèmes P2P structurés, qui permet de contrôler la structure du réseau recouvrant et le placement des ressources dans ce réseau. Le principe est que chaque ressource reçoit une clé (par exemple le nom de fichier ou sa valeur de hachage). Chaque pair va maintenir à jour une table de hachage faisant la liaison entre la ressource et le pair. Cette table est distribuée sur l'ensemble des pairs qui conservent les valeurs correspondant à leur domaine. A l'insertion, le couple (Clé, valeur) est placé sur le pair responsable de la clé et non pas sur le pair l'ayant inséré. Pour récupérer une clé, un nœud doit interroger le nœud responsable de la clé. Ce dernier lui fournit le nœud à interroger pour obtenir la donnée associée à la clé. Avec ce système, le nœud responsable est généralement trouvé en log(n) sauts dans un réseau comportant N participants. Ces architectures permettent d'accéder à n'importe quelle donnée en log(n) sauts.

Par conséquent, les chercheurs ont commencé à proposer des approches qui combinent la nature décentralisée du P2P structuré avec l'efficacité du protocole SIP. Le projet SIPpeer de l'université de Columbia [Singh and Schulzrinne, 2005], et le projet SOSIMPLE de William & Mary College [Bryan et al, 2005] ont été les premières tentatives d'étude des systèmes de communication P2PSIP basés sur le protocole Chord.

A partir de 2005, la recherche P2PSIP a attiré beaucoup d'attention à la fois dans le milieu universitaire et industrielle (Cisco, Nokia, Huawei, Ericsson, etc.). Quelques recherches et propositions ont été initiées de plus en plus et ont été discutées au groupe de travail P2PSIP de l'organisation IETF (RELOAD [Jennings, 2008], [Zheng and Vladimir, 2009], [Bryan et al, 2008]).

Nous considérons à ce niveau le travail SOSIMPLE, mais avant de l'aborder, et pour une bonne compréhension de son principe, il est important de citer le principe de fonctionnement de l'algorithme Chord [Stoica et al, 2001].

A) L'algorithme Chord

Chord [Stoica et al, 2001] est l'un des algorithmes les plus populaires de la DHT, il repose sur une topologie en anneau. Chord utilise une fonction de hachage h, appliquer à la fois aux adresses IP des pairs et aux chaînes ASCII identifiant une ressource disponible sur le réseau. Le résultat est un nombre aléatoire codé sur m bits. La fonction de hachage couramment utilisée, par Chord est : SHA-1 (Fonction utilisée en cryptographie). L'identifiant d'un noeud (*peerID*) est obtenu en hachant l'adresse IP, et la clé d'une ressource (*RessourceID*) est obtenue en hachant son nom. Le principe consiste ensuite à stocker, de façon totalement distribuée, l'information (*RessourceID*, adresse IP) pour chaque ressource, qui associe au nom d'une ressource l'adresse IP du pair qui possède la ressource.

Les nœuds sont répartis dans un anneau de 2^m nœuds, chaque nœud tient une table de repérage (table finger) à jour avec au plus *m* nœuds sauvegardés (Tableau 3.2).

Tableau 3-2 Table de repérage

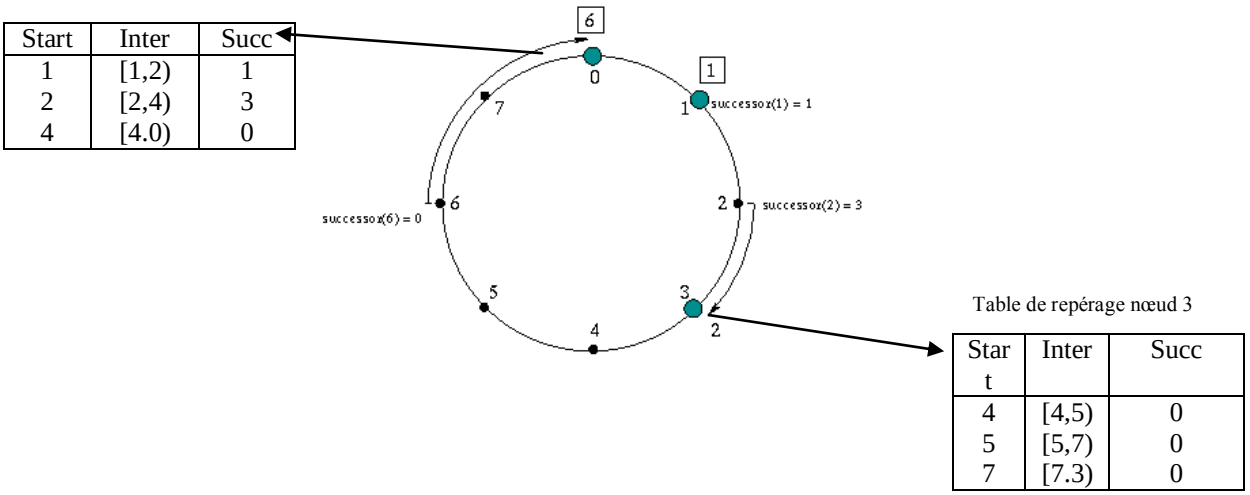
Notation	**Définition**
finger[k].start	$(n+2^{k-1}) \bmod 2^m,\ 1 \le k \le m$
.interval	*[finger[k].start, finger[k+1].start)*
.node	*first node* $\ge n.$ *finger[k].start*
successor	*Le noeud suivant dans le cercle des identifiants* *finger [1].node*
predecessor	*Le nœud précédent dans le cercle des identifiant*

La plupart des identifiants des noeuds ne correspondent pas à des nœuds réels. Chord définit donc une fonction successeur(i) qui retourne l'identifiant *peerID* du premier noeud réel $\ge$ i modulo 2^m. L'information sur la ressource ***nom*** est alors stockée sur le noeud d'identifiant *peerID* égale à successeur(*RessourceID*). L'index est ainsi globalement réparti aléatoirement sur plusieurs noeuds.

Exemple

La figure 3.8 montre un exemple simple d'un réseau Chord composé de trois nœuds m=3, dont les identifiants sont 0, 1 et 3. L'ensemble des clés (ou plus précisément, les clés de ressources) est {1, 2, 6}, et ils sont affectés aux trois nœuds {0.1.3}.

En raison que le successeur de la clé 1 est le nœud qui a le *peerID* égale à 1, la clé 1 est assignée au nœud 1.



Start	Inter	Succ
1	[1,2)	1
2	[2,4)	3
4	[4.0)	0

Table de repérage nœud 3

Start	Inter	Succ
4	[4,5)	0
5	[5,7)	0
7	[7.3)	0

Figure 3-8 Exemple d'un réseau

De même, assignant la clé 2 au premier nœud trouvé déplaçant vers (le sens horaire) la droite de 2 sur le cercle d'identifiants *peerID*, donc le successeur de la clé 2 est le nœud qui a le *peerID* égale à 3. Pour la clé 6, le successeur est le nœud 0 qu'on le trouve par enroulement autour du cercle, ainsi la clé 6 est affectée au noeud 0.

Recherche d'une ressource dans Chord

Quand un nœud N ne connaît pas le successeur d'une clé *RessourceID*, il envoie une requête "Find successor" à un nœud intermédiaire dont son *peerID* est plus proche de *RessourceID*. Le noeud N trouve le noeud intermédiaire en cherchant son entrée dans sa table. Supposons que le nœud F est le plus proche et qui précède la clé recherché *RessourceID*, le noeud F à son tour recherche dans sa table d'entrée l'entrée la plus proche précédant la clé *RessourceID*, et envoie ce message à N. de ce fait N apprends sur les nœuds les plus proches de *RessourceID* jusqu'à l'arrivé au Nœud qui le stocke.

Considérons l'exemple illustré ci-dessus (figure 3.8), et supposons que le noeud 3 souhaite trouver le successeur de l'identifiant 1, puisque 1 appartient à l'intervalle circulaire [7, 3), donc le noeud 3 vérifie le successeur dans sa table de repérage qui donne le noeud 0, le nœud 3 demande du nœud 0 le successeur de 1. À son tour, le noeud 0 déduit de sa table de repérage le successeur de 1 qui est le nœud 1 lui-même, et retourne le *peerID* 1 au nœud 3.

3.3.1.2.1 SOSIMPLE

SOSIMPLE combine le protocole SIP [Rosenberg, 2005] et le protocole d'extensions SIP de la messagerie instantanée SIMPLE [Campbell et al, 2002], avec la propriété d'auto-organisation des tables de hachage distribuée (DHT), le protocole SOSIMPLE est caractérisé par la compatibilité avec l'infrastructure existante du protocole SIP/SIMPLE.

Structure de SOSIMPLE

Les nœuds dans SOSIMPLE sont organisés dans des tables DHT gérées par l'algorithme Chord [Stoica et al, 2001]. Chaque nœud possède un *Nœud-ID* et une table de repérage (finger table). Le Nœud-ID est obtenu par le hachage de l'adresse IP. Les tables de repérage contiennent 16 entrées, et les nœuds l'utilisent pour localiser les autres nœuds dans le réseau.

SOSIMPLE utilise les messages SIP pour maintenir la DHT, enregistre les utilisateurs. Et pour l'établissement des sessions, deux façons d'utilisation de message REGISTER, une nommée l'enregistrement des utilisateurs (User Registration) qui est comme l'enregistrement classique utilisé par le SIP, et l'autre nommée l'enregistrement des nœuds (Node Registration), qui sert à gérer la DHT, comme l'adhésion, le départ, et le maintien de réseau recouvrant.

L'enregistrement des nœuds

L'adhésion au système se fait par l'échange d'un certain nombre de messages REGISTER. Le nouveau nœud qui veut s'adhérer au réseau, utilise son adresse IP, pour calculer son *Node-ID*. Une fois le nœud rejoint le réseau recouvrant, il sera responsable d'une partie des utilisateurs, et de se fait il stocke leurs informations.

Un exemple d'adhésion d'un nouveau nœud est donné dans la figure 3.9. Quand un nouveau nœud souhaite se joindre à la superposition il faut d'abord localiser un nœud déjà dans la superposition, considéré comme un nœud bootstrap (nœud de démarrage). Le noeud joignant calcule son Node-ID, 503 dans l'exemple, et l'envoie dans un REGISTER(1) au nœud du bootstrap, qui a le Node-ID égale à 023.

Supposant que le nœud bootstrap n'est pas le nœud responsable de cette région, il répond avec le message SIP 302 (2) (Moved Temporarily response) en ajoutant le Node-ID du nœud le plus proche de Node-ID 503, dans ce cas le noeud B, avec le Node-ID 445. Le nœud joignant répète le processus, par l'utilisation de ce Node-ID 445 comme le nouvelle bootstrap (3-4).

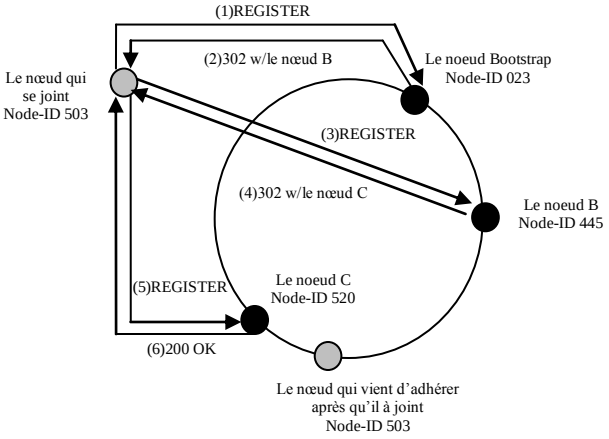


Figure 3-10 Exemple d'adhésion d'un nouveau nœud dans

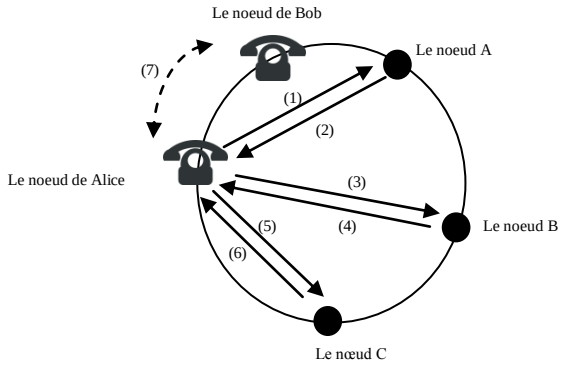


Figure 3-9 Etablissement de session entre Alice et Bob

Enfin, le noeud joignant atteint le nœud qui est actuellement responsable sur cette section de la superposition, dans ce cas le nœud C, avec Node-ID 520. Le nœud C répond par réponse SIP 200 OK, y compris des informations détaillées sur les voisins proches dans les en-têtes (5-6), qui permet au noeud joignant à s'insérer dans la superposition.

L'enregistrement des utilisateurs

L'enregistrement d'un utilisateur est similaire à l'enregistrement d'une ressource dans le système Chord. Dans l'exemple montré dans la figure 3.10, lorsque Alice veut enregistrer son URI SIP, il commence par le hachage de son nom d'utilisateur URI SIP pour produire une ressource-ID. Ensuite Alice échange les messages de REGISTER (1.3.5). Lorsque le nœud C reçoit ce message, il s'aperçoit qu'il est chargé de stocker l'enregistrement d'Alice. Le Nœud C entre une correspondance entre le nom d'utilisateur d'Alice et son adresse IP dans sa table d'inscription et répond au nœud d'Alice avec un 200 OK (6), à ce moment, Alice est enregistrée dans le système.

Etablissement de session

Considérant le même exemple dans la figure 3-10, Alice tente d'appeler Bob, elle commence par le hachage de l'URI SIP de Bob (Bob@domain), pour obtenir le ressource-ID qui correspond à Bob, une fois la ressource-ID est calculée le noeud d'Alice cherche dans sa table de repérage, le nœud avec le Node-ID le plus proche de la ressource-ID de Bob, dans ce cas, le nœud A. Le module agent utilisateur de « Alice » envoie un message INVITE au nœud A (1), supposons que le nœud A n'est pas responsable de cette ressource-ID, dans ce cas il répond à Alice par le message SIP 302 (Moved Temporarily), y compris dans les en-têtes, le

nœud le plus proche de ressource-ID de Bob, dans ce cas le noeud B, (2). Le nœud d'Alice essaie de nouveau le nœud B et reçoit à une réponse SIP 302, qui contient le Node-ID de nœud C (3-4). Le nœud C, il cherche dans sa table de repérage ainsi détermine qu'il est responsable de l'enregistrement de Bob, et examine pour voir si l'enregistrement de Bob est présent. S'il n'existe pas, il répond à Alice par un message 404 Not Found (6), qui indiquant que l'utilisateur n'est pas dans le système. Si Bob est enregistré, le nœud C envoie une réponse 302 Moved Temporarily, offrant un contact directement à Bob (6).

Une fois Alice connaît l'adresse IP de l'agent utilisateur (UA) de Bob, un appel entre les UAS peut être établi directement sans l'utilisation de réseau P2P recouvrant (7), le noeud d'Alice met en cache les informations reçues de nœud C, et peut les utiliser pour les futures communications avec Bob.

B) La sécurité dans SOSIMPLE et P2PSIP

La sécurité est l'un des plus grands défis dans les systèmes P2PSIP. Le coût de maniabilité réduit et la nature décentralisée de P2P, provoque des menaces de sécurité. L'approche P2PSIP hérite tous les problèmes de sécurité qui existe dans les réseaux P2P, et qui comporte l'attaque d'identité (identity attack). L'attaque de réseau recouvrant (overlay attack), attaque de données (data attack), toutes ces menaces de sécurité ont été identifiées par le groupe de travail P2PSIP dans [Matuszewski, 2007].

Une solution a été proposée dans [Marwaha, 2010] pour sécuriser les messages SOSIMPLE, l'auteur a proposé l'utilisation de l'algorithme RSA pour générer des paires des clés et un certificat X.509 [Housley, 2002] pour chaque nœud, ces clés sont utilisées pour sécuriser les données dans la superposition.

Le groupe de travail P2PSIP dans [Jennings, 2007] a proposé des mécanismes de sécurité pour les P2PSIP, Le processus d'adhésion (Enrollement process) proposé consiste à fournir aux nœuds les informations nécessaires pour qu'il rejoint au superposition. Ce processus se déroulera en dehors de la superposition elle-même par un canal sécurisé, La solution proposée permet de fournir un certificat X.509 [Housley, 2002] signé par l'autorité de gestion de la superposition et une paire de clés à chaque nœud, ces clés seront utilisées pour sécuriser les données et les messages. Dans ce système, les informations d'identification sont remplacées par le certificat X.509.

Discussion sur le SOSIMPLE et les P2PSIP

L'utilisation de la DHT peut engendrer des temps d'établissement de session très importants. En effet, pour qu'un utilisateur contacte un voisin directe, il doit transiter par plusieurs noeuds et nécessite l'échange de plusieurs messages alors que cette opération nécessite l'échange seulement de deux messages et aucun passage par des nœuds intermédiaires et par conséquent un temps très faible. De plus, le mécanisme d'enregistrement d'un noeud dans le réseau consomme de l'énergie et du temps et requière une gestion particulière des tables des noeuds pour les maintenir à jour. La procédure d'élection des super noeuds est très couteuse en termes d'énergie. En outre, les départs en masse présentent un sérieux problème et oblige la reconstruction de la topologie.

Le temps d'enregistrement et d'établissement de session restent importants dans cette solution. De plus, l'algorithme Chord impose un seul sens de recherche et cela peut causer le parcours de toutes les tables de hachage du réseau afin de trouver un noeud prédécesseur.

3.3.2 Les travaux qui utilisent les protocoles de découverte de service

La découverte de services est définie pour résoudre le problème de localisation automatique des différents services dans un réseau. Les services sont des entités offertes à l'utilisateur par les noeuds du réseau.

Plusieurs plateformes de découverte de services peuvent être utilisées pour aider la localisation des terminaux SIP. Le dispositif n'a besoin que d'une API pour que les modules SIP et les applications externes peuvent émettre des requêtes de découverte de service. Les travaux qu'ont été proposés sont SIPhoc [Stuedi, 2007], UPnPSIP [Chang, 2004], et sSIP [Leggio et al, 2005]. Il existe aussi d'autres plateformes de découverte de services qui pourraient s'adapter à l'architecture SIP qui sont : Jini, défini par Sun Microsystems [Jini, 2003], Bluetooth Service Discovery défini par Bluetooth Forum [James, 2000].

3.3.2.1 SIPHoc

SIPHoc [Stuedi, 2007] utilise la plateforme MANET SLP (Service Location Protocol) pour la découverte dynamique des services SIP. Et suppose qu'au moins un nœud possède une connexion internet qui permet également aux autres nœuds de se connecter. Un noeud se connecte à internet grâce à deux modules fournisseur de passerelle et fournisseur de connexion.

L'architecture logicielle d'un noeud SIPHoc est donnée dans la figure 3.11. Un noeud SIPHoc comporte deux types de composants, le premier type pour implémenter SIP dans les MANETs et l'autre type est pour permettre au nœud de se connecter à Internet.

Les composants réalisant les fonctionnalités de SIP dans les MANETs sont MANET SLP et SIPHoc proxy.

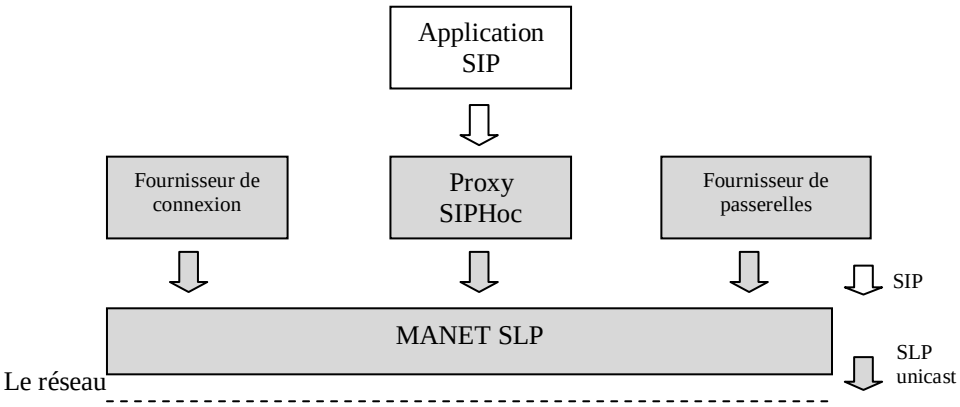


Figure 3-11 Architecture logicielle d'un noeud SIPHoc

MANET SLP

C'est une couche de SLP (Service Location protocole) [Guttman, 1999] pour les MANETs qui fournit une interface SLP ordinaire, et fournit également la découverte distribuée des services dans les réseaux ad hoc. La plateforme MANET SLP, utilise deux requêtes SLP *REGISTER* et SLP *LOOKUP*. SLP REGISTER permet aux nœuds MANET d'enregistrer leurs services tandis que SLP LOOKUP pour la rechercher d'un service. La figure 3.12 montre L'architecture de MANET SLP.

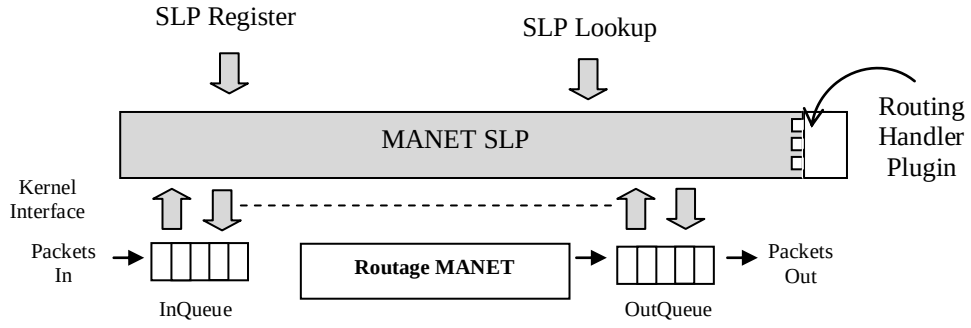


Figure 3-12 L'architecture de MANET SLP

La plateforme MANET SLP utilise une technique qui permet de greffer les informations de services avec les messages de routage. Ceci est fait par un module gestionnaire de routage qui capture les messages de routage, et procède par l'encapsulation de l'information de service dans ces messages. Le module reçoit en entrée les paquets de routage et génère des paquets modifiés qui incluent l'information de services.

SIPHoc proxy

Un proxy SIPHoc joue le rôle d'un serveur d'enregistrement SIP, pour un ensemble d'utilisateurs de certains domaines. Il stocke seulement dans sa table locale de localisation de service, les informations qui concernent les utilisateurs d'un domaine particulier, Chaque proxy SIPHoc utilise le module MANET SLP pour annoncer son adresse de contact pour ses utilisateurs enregistrés. Si un SIPHoc proxy reçoit un message INVITE et ne trouve pas l'utilisateur recherché dans sa table de localisation de service, il consulte la couche MANET SLP et transmet le message INVITE au proxy responsable de cet utilisateur.

Les composants de la connexion des nœuds à Internet

Le fournisseur de passerelle transforme le noeud qui a accès à Internet en passerelle. Ce processus se déclenche dès que le noeud se connecte à Internet.

Le fournisseur de connexion gère la connexion du noeud à Internet lorsqu'il y a une passerelle dans le MANET.

Exemple

La figure 3.13 illustre un exemple de fonctionnement de SIPHoc. Nous supposons deux utilisateurs, Alice et Bob. Les adresses IP des deux machines d'Alice et Bob sont 192.168.220.1 et 192.168.220.2, respectivement. Les proxys SIPHoc utilisent le port 5060, et les applications SIP sur le port 5062. ProxyA est le proxy utilisé par Alice et ProxyB celui de Bob.

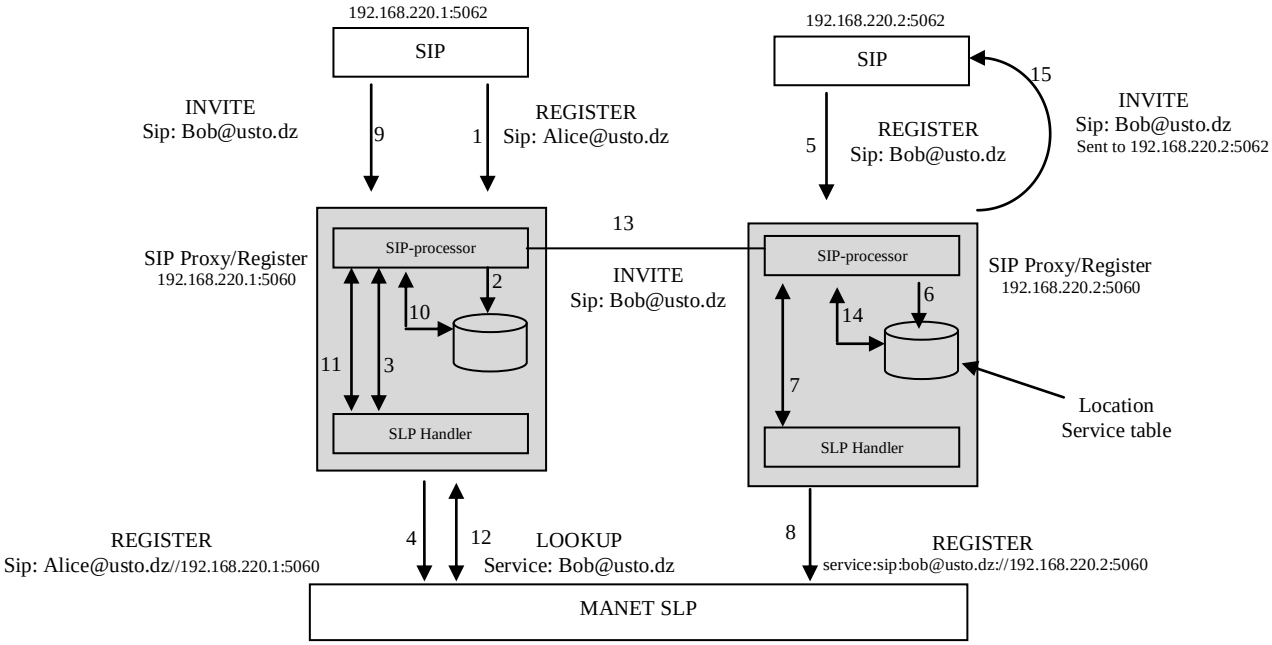


Figure 3-13 Exemple de fonctionnement de SIPHoc

B. L'enregistrement

Pour s'inscrire auprès SIPHoc, les deux utilisateurs envoient leur URI SIP associés à leurs adresse de contact aux proxys (1 et 5, figure 5), pour Alice (sip: alice@usto.dz avec 192.168.220.1:5062), et pour Bob (SIP: bob@usto.dz avec 192.168.220.2:5062). Le SIPHoc proxy de chacun d'eux sauvegarde l'entrée correspondante dans les tables de localisation de services locales (2,6). Egalement le SIPHoc proxy de chaque noeud contacte le module MANET SLP en lui envoyant l'enregistrement afin de le publier dans le MANET en tant que service présent dans le réseau (étapes 3, 4, 7 et 8).

C. Etablissement de Session

Lorsque Alice veut établir une session avec *Bob*, elle envoie un *INVITE* contenant l'URI de *Bob* à son SIPHoc proxy (étape 9). Ce dernier vérifie si *Bob* existe dans sa table de localisation locale (étape 10). S'il ne le trouve pas, il demande, s'il existe, au module MANET SLP de le localiser dans le réseau (étape 11 et 12). Une fois le SIPHoc proxy de *Bob* localisé, il reçoit l'*INVITE* envoyé par le SIPHoc proxy d'Alice (étape 13). Le proxy B cherche l'ID de Bob dans sa table de localisation (étape 14) et transmet l'invitation à l'application multimédia de Bob et la session peut commencer.

3.3.2.2 SIP avec UPnP

L'architecture de SIP-UPnP [Chang, 2004] se compose de trois modules (figure 3.14), la pile UPnP, l'agent d'utilisateur SIP et l'application VoIP, Le module VoIP interagit avec les utilisateurs SIP. L'agent utilisateur SIP est le module qui gère la gestion des messages SIP, et la Pile UPnP fonctionne comme un module de découverte de service, qui collecte les informations nécessaires pour le module agent utilisateur SIP.

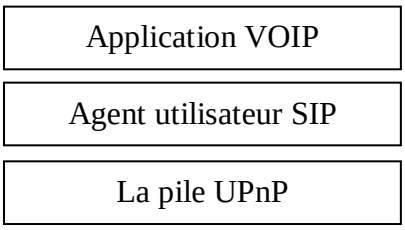


Figure 3-14 L'architecture de système

L'application VoIP :

Le module VoIP est l'interface utilisateur de l'application. Les utilisateurs interagissent directement avec ce module pour utiliser les fonctions, telles que l'initialisation et la terminaison des appels téléphoniques, envoyer et recevoir des messages texte, la recherche des périphériques et les services UPnP. Il offre également une interface aux utilisateurs pour qu'ils configurent leurs téléphones SIP, et leurs profils UPnP, tels que l'URI SIP, et les noms UPnP.

L'agent utilisateur SIP

Il communique directement avec le module sous-jacent UPnP. Le module agent utilisateur SIP est équivalent à la couche de présentation et de session du modèle standard OSI. La fonction principale de ce module est de mettre en place les sessions multimédias entre deux nœuds Ad-hoc. Les nœuds peuvent fonctionner comme des agents utilisateurs, serveur proxy ou serveur de redirection selon divers scénarios.

La pile UPnP

La Pile UPnP est la plus basse module dans l'architecture. Afin de permettre la communication entre les nœuds mobiles, tous les nœuds mobiles sont implémenter à la fois comme des points de contrôle (Controller) et des périphériques racine (Root).

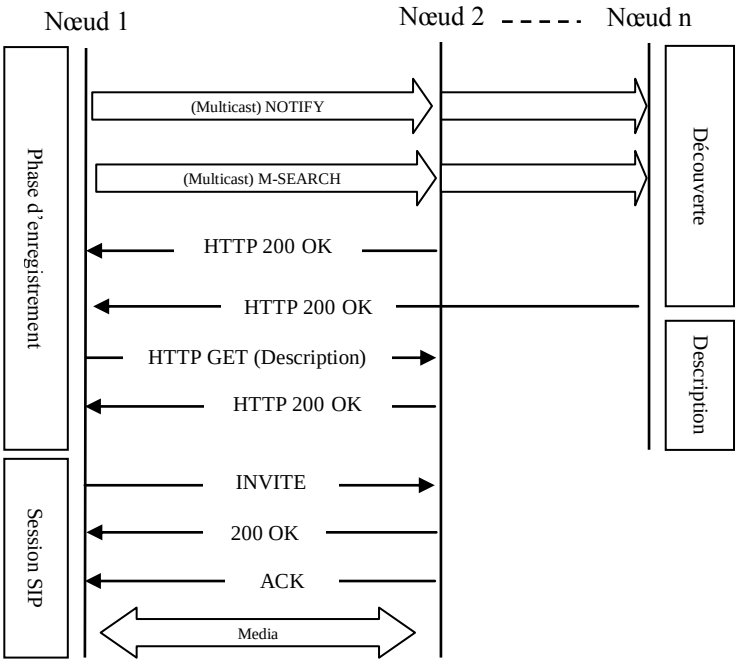


Figure 3-15 Enregistrement et établissement de session

Selon le document qui décrit l'architecture des périphériques UPnP [Unpn Forum, 2008], chaque nœud dispose deux fichiers XML pour présenter les propriétés du périphérique racine UPnP et les services UPnP.

L'enregistrement dans le réseau

Lorsqu'un noeud veut adhérer le réseau, il annonce sa présence à tous les autres nœuds. Il nécessite aussi les informations de tous les autres noeuds présents dans le réseau. Pour cela il envoie une requête en multicast conformément au protocol UPnP, et tous les autres nœuds répondent au nœud joignant.

Dans l'exemple illustré dans la figure 3.15, le nœud 1 emploie la commande de protocole UPnP « NOTIFY » pour notifier tous les nœuds dans le réseau. Il s'appuie sur un mécanisme multicast pour délivrer ce message. Cependant le nœud « 1 » n'a pas encore acquis les informations des autres nœuds, alors il utilise un autre message « M-SEARCH » pour acquérir ces informations.

Après l'acquisition des informations de bases, le nœud « 1 » a besoin de savoir le détail des services fournis par les nœuds, comme les variables accessibles et les actions. Ces détails de descriptions peuvent être obtenus par les mécanismes Control et Event de protocole UPnP. Dans l'exemple le nœud « 1 » envoie ses informations de descriptions qui contiennent son URI SIP et son adresse IP, par la méthode GET, et obtient les réponses des autres nœuds, y compris leurs URIs SIP et leurs adresses IP.

L'établissement de session

Apres la phase d'enregistrement tous les nœuds ont une vue complète sur le réseau et possèdent les liaisons nécessaires, donc dans l'exemple ci-dessus, l'utilisateur « 1 » envoie le message INVITE vers le l'utilisateur « 2 » sans l'utilisation du protocole UPnP.

3.3.2.3 SIP avec le Service Location Protocole (sSIP)

La solution sSIP [Leggio et al, 2005] combine le Decentralized SIP (dSIP) avec une plateforme de découverte de services basés sur le protocole SLP [Guttman, 1999]. Le principe de sSIP est de substituer les requêtes et les réponses SIP générées par dSIP par les requêtes et réponses SLP et les services à découvrir seront les enregistrements SIP. La figure 3.16 illustre l'architecture de sSIP. Son architecture est similaire au dSIP avec l'ajout d'un nouveau module qu'est l'API de découverte de service.

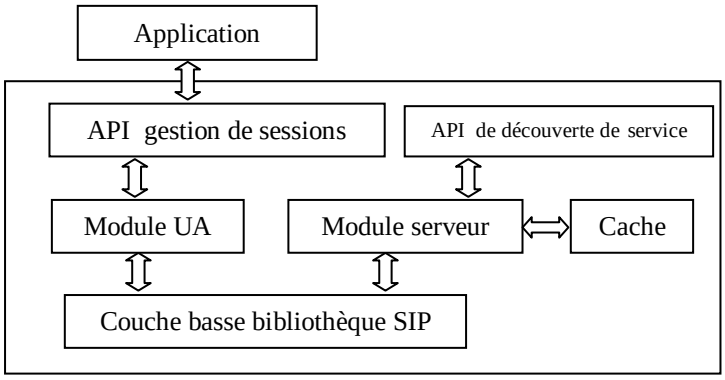


Figure 3-16 l'architecture logiciel de sSIP

Cet API permet au module proxy d'utiliser la plateforme Service Discovery, pour récupérer les liaisons. Elle peut également être utilisée par l'application elle-même pour trouver d'autres services dans le réseau ad-hoc.

Dans sSIP le service de localisation est exploité et maintenu par l'échange de requêtes et réponses SLP. Chaque noeud qui rejoint le réseau ad hoc envoie un message *SLP Query* contenant son AOR (Adresse Of Record). Ce message est envoyé à tous les noeuds du réseau qui vont mettre à jour leurs caches et confirment l'enregistrement par un message *SLP Reply* qui contient leurs informations. De plus des messages de rafraichissements SLP sont diffusés périodiquement pour garder les caches à jour. Le fonctionnement de SIP dans sSIP est identique à dSIP seulement que les messages REGISTER et INVITE sont encapsulés dans des messages SLP Query.

3.3.2.4 Sécurité dans l'approche des protocoles de découvertes de services

SIPhoc repose sur les mécanismes de sécurité offerts par le standard SLP. Le sSIP utilise une version modifiée de mécanismes de sécurité du protocole SLP. Dans le standard SLP, il est possible d'authentifier l'entité qui annonce un service. L'architecture SLP définit deux types d'entités, l'une appelée agent serveur (SA, Server Agent), et l'autre appelée agent utilisateur (UA, User Agents). Lors de l'envoi d'un nouveau service par un agent utilisateur, l'entité agent serveur signe l'URL et les listes des services annoncés. Le SA ajoute la signature correspondant à la fin du message. Les informations d'authentification liées au SLP sont réalisées par ce qu'est appelé le bloc d'authentification.

Le bloc d'authentification basique de SLP est illustré dans la figure 3.17. Ce bloc est ajouté à un message SLP pour garantir l'intégrité du message et assure l'identité de l'expéditeur.

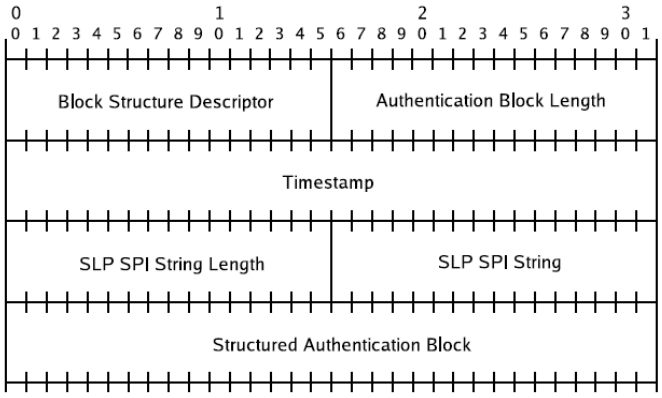


Figure 3-17 Bloc d'authentification SLP ([Guttman, 1999])

La structure basique de bloc d'authentification SLP a été modifiée dans sSIP pour apporter un soutien de sécurité plus avancés. Les modifications sont illustrées dans la figure 3.18. Le principal problème dans la structure de base du bloc d'authentification SLP est que seule l'entité qui fournit un service peut être authentifiée. De plus, SLP n'offre aucun moyen pour protéger l'intégrité de tous les messages. Le bloc d'authentification amélioré est plus grand que celui du standard SLP, et introduit donc un plus overhead que le dSIP.

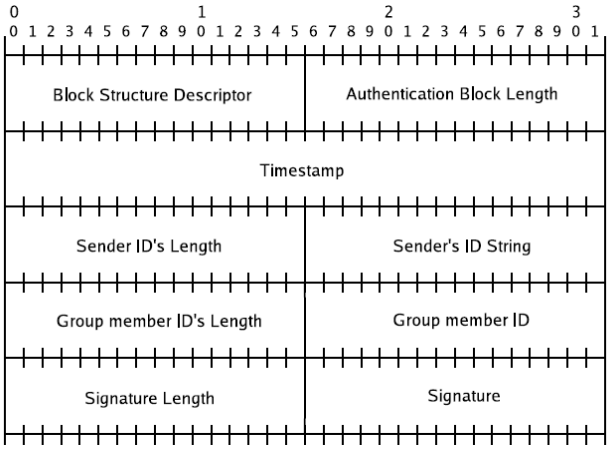


Figure 3-18 Bloc d'authentification SLP amélioré ([Liimatainen, 2005])

Discussion

Le problème principal des solutions qui utilisent la découverte de service reste l'interopérabilité; les noeuds implémentant une certaine architecture logicielle peuvent ne pas pouvoir localiser d'autres noeuds implémentant une autre architecture. A titre d'exemple un nœud qui utilise le protocole UPnP ne peut pas trouver le nœud qui fonctionne avec le protocole SLP. L'avantage de ces solutions qu'ils utilisent les fonctionnalités des protocoles de localisation de services et qu'ils ne modifient pas le protocole SIP, il considère que les URIs SIP des nœuds et leurs adresses IP comme des services annoncés par les protocoles de découvertes des services. Un autre point fort de ces solutions, c'est la réutilisation de tous les mécanismes de sécurité propres aux protocoles de localisation de services afin de sécuriser la phase de découverte des terminaux SIP.

3.3.3 L'approche des protocoles de routage Ad-hoc

L'idée d'utilisation de protocoles de routage MANETs pour la découverte des terminaux SIP, a été introduite par Banerjee [Banerjee et al, 2006]. Banerjee propose deux approches d'utilisation de protocoles de routage, une faiblement couplée LCA «Loosely Coupled Approche » et l'autre fortement couplée TCA « Tightly Coupled Approach ». Dans le LCA la découverte des terminaux SIP est découplée du protocole de routage et se fait par une technique similaire au protocole de routage AODV [Perkins, 2003]. Tandis que dans le TCA la découverte des terminaux SIP est intégrée avec le protocole de routage CBRP (Cluster Based Routing Protocol) [Jiang et al, 1999]. Une technique de regroupement est utilisée par ce protocole. Le regroupement permet de créer une topologie virtuelle, et divise le réseau physique en un ensemble de groupes (cluster). Chaque Cluster est géré par un chef de groupe (Cluster Head), qui sont relies entre eux par des nœuds passerelles. Le TCA considère les

clusters comme des serveurs SIP et de ce fait le SIP proxy, le serveur d'enregistrement et le serveur de redirection sont intégrés dans les clusters.

3.3.3.1 Tightly Coupled Approach

L'approche TCA [Banerjee et al, 2006] intègre la découverte des terminaux SIP avec le protocole de routage CBRP [Jiang et al, 1999]. Ce protocole crée une topologie virtuelle qui comporte des clustershead formant un Backbone réseau, qui s'occupe du routage de messages SIP, et les paquets de données. Le diagramme fonctionnel de TCA est donné dans la figure 3.19.

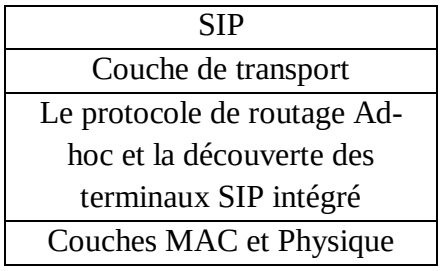


Figure 3-19 Diagramme fonctionnel de TCA

Construction de la topologie virtuelle

Le protocole CBRP [Jiang et al, 1999] utilise une approche totalement distribuée dans la construction de clusters. Les nœuds ayant des degrés plus élevés sont considérés comme des clusterheads potentiels, et les autres noeuds sont au maximum à un saut de leur clusterheads. Chaque clusterhead est connecté à tous les autres clusterheads soit directement ou à travers un ensemble de nœuds spéciaux désignés comme passerelles.

Le protocole génère un nombre minimal de clusterheads dans le réseau pour réduire le nombre de sauts, et minimise la consommation de l'énergie des nœuds, L'ensemble d'un nombre minimal de clusterheads et passerelles du backbone réseau forment un ensemble minimal dominant (Minimal Dominating Set (MDS)). L'ensemble des termes utilisés dans TCA sont représentés dans la table 3.3.

Tableau 3-3 Les termes utilisés dans le protocole de clustering

Terme	Signification
Node	Nœud de réseau ad hoc qui peut jouer le rôle d'un agent utilisateur, un serveur d'enregistrement, un service de localisation ou un serveur proxy.
Node ID	c'est un identifiant qui identifie de manière unique un nœud dans le réseau. L'adresse du nœud interne ou l'adresse IP est généralement utilisée comme identifiant.
Degree	Représente le nombre de nœuds adjacents à un nœud donné
Cluster	Groupe de nœuds avec un chef de groupe (clusterhead)
Clusterhead	C'est un nœud qui sélectionne lui même comme un chef de groupe (Clusterhead), un clusterhead possède toutes les informations des membres de groupe
Cluster member	Un nœud dans un groupe (cluster) qui n'est pas un clusterhead est un membre
Adjacency Table	Une table d'adjacence pour un nœud contient une liste de tous les nœuds voisins avec leurs types, à savoir si elles sont clusterhead ou membres.
Cluster Adjacency Table	Contient la liste de tous les clusterheads qui sont à 2 sauts de distance

La sélection de clusters head

En amorçage, chaque nœud diffuse un message HELLO à ses voisins à un saut. Un nœud après qu'il reçoit les messages HELLO de ses voisins à un saut, il calcule son degré et construit sa table d'adjacence. Le format de la table d'adjacence et du paquet HELLO sont respectivement représentés dans les figures 3.20 et 3.21. Initialement, le degré et les champs ClusterHeadFlag sont mis à 0, et les tableaux d'adjacence sont conservés vide, après une période de temps spécifiée par (HELLO_PERIODE). Chaque nœud rediffuse de nouveau le message HELLO, cette fois avec son degré et sa table d'adjacence complète.

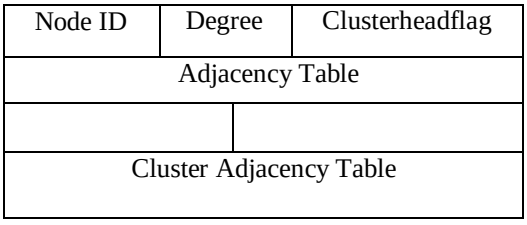


Figure 3-20 Format du message HELLO

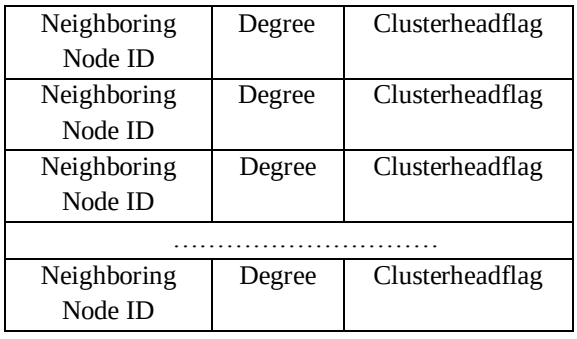


Figure 3-21 Format de la table d'adjacence

Après avoir reçu au moins trois messages HELLO, chaque nœud peut exécuter l'algorithme de sélection des clustershead checkClusterhead représenté dans la figure 3-22. et choisit lui même en tant que clusterhead si l'une des conditions suivantes est satisfaite :

Condition 1. Le nœud a le plus grand degré dans son voisinage à 1-saut.

Condition 2. Le nœud a le plus grand degré dans son voisinage à 2-sauts.

Les fonctions et les variables utilisées dans l'algorithme checkClusterhead sont résumées dans la Table 3-4 Une fois les clustersheads sont choisis, ils agissent en tant que Proxy SIP et serveurs d'enregistrement SIP.

Tableau 3-4 Les notations utilisées l'algorithme checkClusterhead

Cluster (j)	Cluster head du nœud j
Degree(j)	Degré du noeud j
Clusterheadflag(j)	ClusterHeadFlag du noeud j 1 pour cluster head 0 un membre de cluster ou pas encore assigné -1 signifié que le nœud possède le plus grand degré mais ces ressources ne suffisent pas.
N1(j)	Ensemble de nœud à un saut du nœud j

```
1:  checkClusterhead(i);
2:  for (j ∈ N1(i) ∪ {i}) do
3:     cluster(j) = j;
4:     for (k ∈ N1(j)) do
5:        if (degree(k) > degree(j)) then
6:           cluster(j) = k;
7:        end if
8:     end for
9:     if (cluster(j) == i) then
10:       clusterheadflag(i) = 1;
11:    end if
12: end for
```

Figure 3-23 L'algorithme de sélection de clusterheads

Formation de Cluster

Une fois les clustersheads sont choisis, ils maintiennent la connectivité aux clusterheads voisins à travers le processus de sélection de passerelle, et informent leur clustermembres par l'envoie des messages HELLO. Les membres du cluster envoient alors un message SIP REGISTER au clusterhead responsable et s'enregistre auprès du serveur d'enregistrement.

Sélection de passerelle (Gateway)

Après la sélection des clusterheads, chacun d'eux maintient la connectivité avec ses clusterheads voisins à travers des nœuds passerelles. Chaque noeud qui n'est pas cluster head doit exécuter l'algorithme de sélection de passerelle afin de savoir s'il est passerelle ou pas. Le principe de cet algorithme repose sur l'utilisation des tables de voisinage de cluster. La Figure 3.23 montre le format de cette table, chaque nœud membre doit vérifier s'il existe dans sa table de voisinage de cluster deux qui ne sont pas à portés et qu'il n'y a aucun autre noeud qui peut être passerelle entre ces deux clusterheads.

Cluster Head Node ID	Gateway Node ID
Cluster Head Node ID	Gateway Node ID
Cluster Head Node ID	Gateway Node ID
……	
Cluster Head Node ID	Gateway Node ID

Figure 3-24 Le format de la table d'adjacence clusters

Les fonctions de serveur d'enregistrement et le proxy SIP

Un nœud s'enregistre auprès du serveur d'enregistrement SIP correspondant en envoyant un message SIP REGISTER. Le service de localisation associée au serveur d'enregistrement dans le clusterhead, sauvegarde la liaison entre les URI-SIP et les adresses des membres de cluster. La topologie virtuelle induite par les clustersheads, permet aux serveurs d'enregistrement de fonctionner exactement de la même manière qu'un réseau avec infrastructure.

L'établissement de session SIP

Lorsqu'un noeud veut établir une session avec un autre noeud du réseau, il génère un message INVITE qui contient l'ID-SIP du noeud destination et l'envoie à son clusterhead. Le clusterhead cherche dans sa table de correspondance et envoie l'INVITE à ses clusterheads adjacents après avoir changé le champ Record-Route du message INVITE. La même procédure est répétée jusqu'à atteinte du noeud destinataire. Après la réception de l'INVITE, le noeud destinataire génère un message SIP 200 OK qui traversera la route spécifiée dans le champ Record-Route.

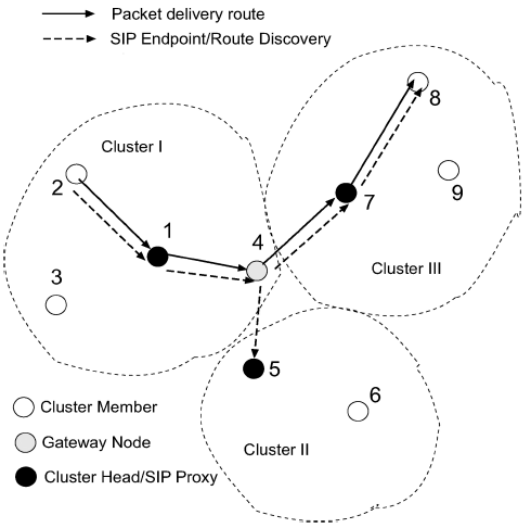


Figure 3-25 Formation de clusters et la découverte de routes

La figure 3.24 illustre l'établissement de session entre deux noeuds 2 et 8. Le nœud « 2 » envoie un message SIP INVITE, à son clusterhead « le nœud 1 ». Le clusterhead 1 alors sélectivement diffuse le message INVITE au clusterhead voisin à travers le nœud passerelle correspondant « nœud 4 ». A l'arrivé du message au clusterhead du cluster III « noeud 7 » il cherche le "Request-URI" parmi les URIs des nœuds qui ont enregistré. Ainsi, il envoie le message INVITE au nœud de destination « 8 ». Tout au long de la transmission du message INVITE, le chemin composé de la série de proxys traversant est enregistré dans le champ «Record-Route». Un message SIP OK est alors renvoyé vers le nœud « 2 » suivant la liste des proxys dans le champ «Record-Route» dans l'ordre inverse.

3.3.3.2 LCA: Loosely coupled approach

LCA [Banerjee et al, 2006] fonctionne au dessus du protocole de routage ad hoc et emploie une technique similaire à celle que le protocole AODV [Perkins, 2003] utilise pour la découverte de routes vers une adresse IP de destination. La figure 3.26 illustre la disposition des différents composants de LCA dans le modèle OSI.

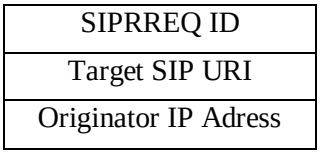


Figure 3-26 Le message SIPRREQ

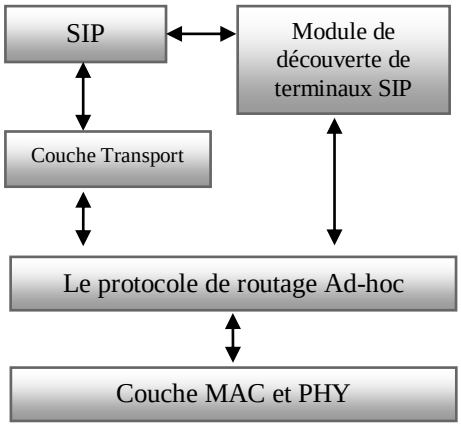


Figure 3-27 Loosely Coupled Approach

Pour localiser un utilisateur donné qui a comme identifiant SIP son « URI SIP », LCA utilise deux types des messages, le message « SIPRREQ » et « SIPRREP », dérivés respectivement des messages «RREQ : Route REQuest » et «RREP : Route REPly » du protocole AODV. Le format du message SIPRREQ est représenté dans la figure 3.25.

Le message SIPRREQ contient un numéro de séquence « SIPRREQ ID » qu'un nœud l'incrémente à chaque envoie d'un nouveau message SIPRREQ, le champ « Target SIP URI » représente l'identifiant SIP « URI SIP » de l'utilisateur recherché « destination ».

La paire < SIPRREQ ID, Originator IP Adress > identifie de manière unique un message SIPRREQ.

Etablissement de session LCA

Lorsqu'un utilisateur veut établir une session SIP avec un autre, son module de découverte de terminaux SIP diffuse un message SIPRREQ contenant l'identifiant URI SIP de l'utilisateur de destination. Afin d'éviter le problème de tempête de diffusion « Broadcast Storm », le nœud source utilise la technique « Expanding Ring » qui limite le nombre de sauts de messages SIPRREQ, au début le TTL du message SIPRREQ est initialisé à TTL_START et après un délai d'attente appelé RING_TRAVERSAL_TIME, s'il n'y a pas de réponse, le TTL est incrémenté de TTL_INCREMENT. Ceci et continue jusqu'à ce que TTL atteint TTL_THRESHOLD, où il est mis à NET_DIAMETER, et dans ce cas la requête SIPRREQ est retransmise en utilisant l'algorithme du back-off pour éviter la congestion du réseau. Les valeurs typiques des paramètres de TTL utilisées dans le processus de découverte peuvent être obtenues à partir des recommandations du protocole AODV [Perkins, 2003].

L'adresse IP du nœud destinataire est déterminée lorsque la requête SIPRREQ atteint le noeud cible ou un noeud intermédiaire contenant une correspondance récente entre l'identifiant « URI SIP » du nœud recherché et son adresse IP. Une réponse SIPRREP est ensuite envoyée en mode unicast vers le noeud source contenant l'adresse IP de la destination. Le processus de découverte est illustré dans la figure 3.28.

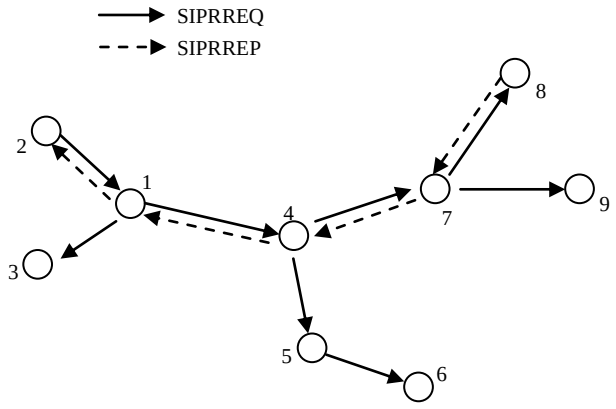


Figure 3-28 Exemple de découverte

SIPRREQ ID
Target SIP URI
Target IP Address
Originator IP Adress

Figure 3-29 Le message SIPRREP

Le format du message SIPRREP est illustré dans la figure 3.27, le champ « Target Node Address » est l'adresse IP qui correspond au « Target SIP URI ». Un nœud génère un message SIPRREP dans l'un des deux cas suivants :

i) le nœud est destinataire.

ii) le noeud possède la liaison «Target SIP URI, Target IP Address » du nœud de destination d'un message SIPRREQ, dont le numéro de séquence « SIPRREQ ID » est supérieur ou égal au numéro de séquence « SIPRREQ ID » figurant dans le message SIPRREQ reçu.

Lors de la génération du SIPRREP, le nœud destinataire copie le « Target SIP URI », le « SIPRREQ ID », et « Originator IP Address » à partir du message SIPRREQ. Les nœuds intermédiaires utilisent le « SIPRREQ ID » et gardent la liaison correspondante pour une future utilisation. Après sa création, le message SIPRREP est envoyé en unicast directement vers la source. Ainsi, lorsque le SIPRREP atteint la source, il connaît maintenant l'adresse IP de destinataire, il remet le processus de routage des messages SIP et les paquets de médias au protocole de routage AODV.

Discussion sur TCA et LCA

Le TCA est conçu pour les réseaux relativement stables. Cependant, dans les réseaux où la topologie change fréquemment, le coût de la construction et maintenance de la topologie virtuelle devient élevé, ce qui conduit à la consommation de ressources des nœuds et de la bande passante. De plus il rend le temps d'établissement de session très important, et de ce fait cette solution devient inadaptée pour ce type de réseaux. D'autre part l'hétérogénéité des nœuds limite l'utilisation de cette solution, de sorte que pour qu'un nœud joue le rôle d'un clusterhead il doit y avoir les capacités de stockage et de calcul. Ce que distingue le TCA aux autres travaux, est la topologie virtuelle similaire à celle des réseaux avec infrastructure qu'il utilise, et qui rend l'exploitation de serveurs SIP et leur fonctionnalité identique à la spécification du protocole SIP.

Le LCA n'utilise pas d'enregistrement SIP et donc aucun noeud ne connait ses voisins. Lorsqu'un noeud veut établir une session avec un autre noeud du réseau, une nouvelle procédure de recherche est activée par la diffusion d'une requête de recherche SIPRREQ selon la technique « Expanding Ring ». Cependant, cette recherche peut prendre beaucoup de temps. Lorsque tous les nœuds se déplacent aléatoirement LCA, dans ce cas est plus performant que le TCA en termes de temps de latence.

3.4 Comparaison entre les différentes classes de décentralisation du SIP

Une variété de solutions de décentralisation du SIP ont été proposées dans la littérature. Nous avons essayé de les classer selon le principe qu'elles utilisent en trois classes principales (Voir Tableau-3.3): les solutions qui utilisent le principe du P2P, les solutions qui utilisent les protocoles de découverte de services, et les solutions qui utilisent les protocoles de routage. Les solutions de P2P ont été divisées en deux sous classes de solutions :

Les solutions P2P non structuré sont destinées à fonctionner dans les réseaux de très petite taille, de plus, la recherche dans ces réseaux n'est pas déterministe. Leur seul avantage est qu'aucun mécanisme de gestion ou de maintien de la topologie n'est déployé et la phase d'enregistrement dans ces réseaux est très rapide. Les deux solutions qu'on a discuté dans cette sous classe dSIP et SIPCache utilisent le mécanisme de sécurité SIP Identity qui assure l'authentification et l'intégrité, d'un autre coté aucun mécanisme de sécurité n'a été utilisé pour assurer la confidentialité.

Dans la sous classe P2P structuré, le coût de maintien de la topologie en terme de bande passante est très élevé et le mécanisme d'enregistrement est complexe. Ce type de solution fonctionne sur des réseaux de grande taille et leur recherche est déterministe. Le SOSIMPLE et P2PSIP utilisent le RSA pour générer des paires de clés et un certificat X.509 pour chaque nœud. Ce qui permet aux nœuds de garantir l'authentification et l'intégrité mais la confidentialité reste absente.

La deuxième classe souffre de problème d'interopérabilité qui empêche les nœuds qui emploient deux plateformes de découverte de service différentes de se communiquer, ainsi que ces solutions ne s'appliquent pas aux réseaux avec une grande taille. L'avantage de ces solutions est qu'elles se reposent sur les mécanismes de sécurités fournie par les protocoles de découverte de services afin d'assurer l'authentification et l'intégrité.

La troisième classe de solution est celle qui utilise les protocoles de routage Ad hoc. Deux solutions qui utilisent ce principe ont été étudiées le LCA et le TCA. Cette dernière présente plusieurs avantages, comme le temps d'établissement de session relativement rapide, et la recherche déterministe. Elle supporte aussi les réseaux de très grande taille. Cependant, pour les réseaux relativement mobiles, le temps d'établissement de session devient très lent et la consommation de la bande passante très élevée. Le LCA supporte les réseaux de très grande taille, il devrait être adopté, quand il est nécessaire d'établir rapidement des sessions dans des réseaux qui sont caractérisé par la mobilité élevé des nœuds. Autrement, TCA fonctionne bien

Tableau 3-5 Comparaison entre les différentes solutions

	Approche P2P			Approches Découverte de service		Approches Routage Ad hoc	
	P2P non structuré		P2P structuré			Découplé	Intégré
	dSIP	SIPCache	SOSIMPLE	sSIP	SIPHoc	LCA	TCA
Principe	l'intégration de fonctions d'enregistrement dans chaque nœud, ainsi chaque noeud connait l'information de contact de tous les autres noeuds du réseau	- Combine le dSIP avec l'algorithme PCache -Optimise le dSIP et l'étend sur de grands réseaux	-Combine le SIP/SIMPLE protocole avec Chord -Les nœuds sont organisés dans des tables DHT gérées par l'algorithme Chord	Combine le dSIP avec une plateforme de découverte de services basés sur le protocole SLP, le principe de sSIP est de substituer les messages générés par dSIP par les messages SLP.	Utilise la plateforme MANET SLP pour la découverte dynamique des services SIP	Emploie une technique similaire à celle que le protocole AODV utilise pour la découverte de routes	La découverte des terminaux SIP est intégrée avec le protocole de routage CBRP qui utilise une technique de regroupement (clustering)
Avantages	-Temps d'établissement de session rapide -Facile à gérer et à maintenir	- Temps d'établissement de session rapide - Taille du réseau supporté est grande -Recherche déterministe	- Recherche déterministe - Taille du réseau supporté est grande	-SLP consomme moins de bande passante que le SIP - Temps d'établissement de session rapide	-Temps d'établissement de session rapide. -Taille du réseau supporté est grande	-Etablissement de session rapide dans les réseaux avec une mobilité élevée -Taille du réseau supporté est grande	-Fonctionne bien pour les réseaux avec une faible mobilité des nœuds -Taille du réseau supporté est grande
Inconvéni	- Taille du réseau supporté est petite -Recherche aléatoire et non déterministe	- Mécanisme d'enregistrement complexe et probabiliste	- Temps d'établissement de session moyen - Mécanisme d'enregistrement complexe.	-Problème d'interopérabilité -Taille du réseau supporté est petite	- Nécessite la présence d'une connexion internet -Problème d'interopérabilité	pas d'enregistrement SIP -Encombre la bande passante	-Problème d'hétérogénéité des noeuds -La mobilité élevée implique un temps d'établissement lent
Sécurité	- L'authentification et l'intégrité sont assurées par le mécanisme SIP Identity -pas de mécanisme pour la confidentialité	- L'authentification et l'intégrité sont assurées par le mécanisme SIP Identity -pas de mécanisme pour la confidentialité	- Utilise le RSA pour générer des paires de clés et un certificat X.509 pour chaque nœud. - pas de mécanisme pour la confidentialité	- Utilise une version modifiée de mécanisme de sécurité du protocole SLP - pas de mécanisme pour la confidentialité	- Utilise le mécanisme de sécurité du protocole SLP - pas de mécanisme pour la confidentialité	Pas de mécanisme de sécurité	Pas de mécanisme de sécurité

pour les réseaux avec une faible mobilité des nœuds. En ce qui concerne la sécurité, les deux solutions LCA et TCA n'ont pas abordé la question de sécurité.

3.5 Conclusion

Dans ce chapitre on a étudié les différentes solutions de décentralisation du protocole SIP dans les réseaux Ad hoc. Nous les avons classifiées en trois grandes classes. Les deux premières classes utilisent des mécanismes de sécurité pour assurer l'authentification et l'intégrité dans la phase d'établissement de sessions. L'autre classe qui emploie les techniques des protocoles de routage n'utilise aucun mécanisme de sécurité, En tenant compte des limites constatées sur la solution LCA, nous proposons l'amélioration et la sécurisation de cette solution par l'utilisation de la signature basée sur l'identité. Le chapitre suivant fait l'objet de la solution que nous proposons pour sécuriser le LCA.

Chapitre 4 : Notre Contribution Secure_LCA

4.1 Introduction

La solution LCA (Loosely Coupled Approach) [Banerjee et al, 2006] utilise une technique similaire au protocole AODV (Ad-Hoc On-Demand Distance Vector) [Perkins, 2003] pour établir des sessions multimédia. Chaque entité dans LCA joue le rôle d'un routeur et prend part de la responsabilité d'acheminement des messages de signalisation. Cette manipulation rend le LCA très vulnérable aux attaques. Dans ce chapitre nous proposons un schéma sécurisé nommé Secure_LCA [Douara, 2012a],[Douara, 2012b]. Dans le schéma proposé, on a utilisé la signature basée sur l'identité afin de garantir l'authentification et l'intégrité des messages.

4.2 Vulnérabilités du protocole LCA et les attaques possibles

Le schéma de LCA n'offre aucun système de sécurité. Toutes les entités peuvent participer au processus de signalisation donc il n'y a pas de barrières pour un nœud malicieux de causer des perturbations dans le trafic circulant. Le schéma de LCA suppose que tous les nœuds sont légitimes et assume l'absence de nœuds malveillants dans le réseau ad hoc. Évidemment, cette hypothèse n'est pas applicable dans ce type de réseau, qui est caractérisé par le support sans fil ouvert. Puisque LCA utilise une technique similaire au protocole AODV, donc il hérite toutes les vulnérabilités et les attaques possibles, de ce dernier.

Les attaques peuvent être classées en attaques passives, où l'attaquant est limité à l'écoute et l'analyse du trafic échangé, et les attaques actives, dans ce dernier mode l'attaquant peut injecter son propre trafic, modifier le fonctionnement d'un nœud, usurper l'identité d'un élément valide, rejouer des messages, modifier des messages transitant sur le réseau ou provoquer un déni de service. L'attaque passive prive le réseau de la confidentialité des messages échangés. Eventuellement, l'analyse du trafic représente un risque pour l'anonymat des participants et le respect de leur vie privée. Dans ce mémoire, nous considérons principalement les attaques suivantes :

a) L'usurpation de l'identité qui consiste à se faire passer pour quelqu'un d'autre en utilisant son identité. L'attaquant se présente en utilisant l'identité d'un nœud légitime et peut ainsi communiquer avec les nœuds du réseau sans être rejeté.
b) La modification des champs des messages de contrôle (l'intégrité) où un nœud malicieux peut altérer les différents champs d'un message SIPRREQ. Chaque modification rapportée a un effet particulier sur le processus de découverte des nœuds.

c) La fabrication des messages SIPRREQ, l'attaquant dans ce cas injecte des faux messages afin de perturber le processus de découverte déjà établis.

4.3 Notre Contribution « Secure LCA »

Le schéma de LCA n'utilise aucun mécanisme de sécurité. Dans la phase d'établissement de session. Nous proposons l'amélioration de LCA par des solutions qui permettent de rendre le LCA résistant aux attaques. Dans notre schéma, le principal problème de sécurité consiste à assurer l'identité des utilisateurs SIP dans le réseau ad-hoc. Par exemple, quand une réponse SIPRREP au message SIPRREQ est renvoyée, il est très important de s'assurer que la réponse provient d'un utilisateur légitime. En d'autres termes, il est très important de vérifier que l'émetteur est vraiment l'utilisateur indiqué dans le message SIPRREQ.

Dans les parties suivantes nous présentons une évaluation, de différents mécanismes de sécurité SIP, et aussi les systèmes de cryptographie, afin de décider quel mécanisme ou quel système correspond le mieux à l'architecture LCA et à l'environnement Ad-hoc. Ensuite nous présentons la façon dont nous avons modifié et appliqué une telle solution choisie, pour être déployée dans le LCA.

4.3.1 Evaluation des mécanismes de sécurité SIP

Le SIP utilise quatre mécanismes de sécurité pour assurer la confidentialité, l'intégrité, l'anonymat et l'authentification au travers de la signalisation. Les mécanismes sont : HTTP digest authentication [Franks et al, 1999], TLS [Dierks and Allen, 1999], IPsec [Kent and Atkinson, 1998], et S/MIME [Ramsdell, 2004]. Parmi ces méthodes, l'authentification HTTP Digest a été écartée car elle repose sur les secrets partagés. En général, un message destiné à atteindre plusieurs destinataires, ne peut pas être protégé par l'authentification HTTP, car il serait enclin à des failles de sécurité. D'autre part l'envoie de plusieurs messages en unicast avec des challenges différents à tous les nœuds n'est pas pratique, car les nœuds ad-hoc ne savent pas les adresses des autres nœuds dans le réseau.

TLS a été ignoré car il ne prend pas en charge l'envoie en broadcast et en multicast, est repose sur le protocole TCP. IPsec est similaire au TLS, il peut être employé essentiellement dans SIP entre des entités SIP qui ont une configuration et une association de sécurité assez statique, ce qui nécessite une structure de gestion complexe de clés, donc il n'est pas applicable au LCA.

S/MIME dans un contexte SIP permet trois utilisations : la transmission d'un certificat, la signature et le chiffrement. Le chiffrement de tout le message SIP de bout-en-bout pour des besoins de confidentialité n'est pas approprié dans LCA à cause des intermédiaires du réseau

ad-hoc qui ont besoin de voir certains champs des en-têtes afin d'acheminer correctement les messages. Il est toujours possible pour un émetteur d'inclure son certificat dans le message. L'ajout d'un certificat à des corps S/MIME est nécessaire lorsque le destinataire ne possède pas le certificat de l'émetteur stocké localement. Ce qui augmente d'une manière très significative la taille des messages SIP. Les nœuds nécessitent également d'associer à chaque URI une clé publique ce qui n'est pas forcément facile dans un environnement ad-hoc. Dans notre environnement du réseau, le principal objectif de conception est de minimiser le surcharge du réseau, et S/MIME ne satisfait pas à cette exigence.

De toute façon, la protection de messages augmente la consommation de bande passante et les ressources des nœuds. La solution de sécurité proposée pour LCA doit supporter la diffusion en broadcast. Le schéma d'authentification basée sur la signature utilise des certificats et des listes de révocation de certificats CRL (Certificate Revocation List), Les deux nécessitent un stockage et un coût de communication élevés. Pour cela l'utilisation d'une solution à base de clés publiques et de certificats est non pratique dans le LCA. Même l'utilisation d'un secret partagé pratiquement n'est pas faisable, donc n'importe quelle solution basée sur les clés partagées est éliminée.

L'analyse et l'évaluation mentionnées ci-dessus, qui décrit la difficulté de l'utilisation des mécanismes de sécurités SIP, nous a conduit à déployer le système de cryptage basé sur l'identité au LCA, La cryptographie basée sur l'identité réduit ainsi considérablement la complexité et le coût de l'établissement et la gestion des clés publiques connue sous le nom d'infrastructure à clés publiques. Les paragraphes suivants expliquent comment le système fonctionne ainsi que la manière dont nous allons l'exploiter dans LCA.

4.3.2 Cryptographie basée sur l'identité

Le concept de la "cryptographie basée sur l'identité" a été introduite par Adi Shamir en 1984 [Shamir, 1984]. Contrairement aux schémas conventionnels, les schémas basés sur l'identité IBE (Identity Based Encryption), offrent la possibilité de choisir librement la clé publique. Les certificats électroniques, réputés pour leur complexité, sont alors remplacés par des informations facilement mémorisables, telle que l'adresse URI SIP, comme clé publique. Dans la mesure où les schémas basés sur l'identité ne requièrent aucun mécanisme de certification, ils permettent de simplifier considérablement l'implémentation sécurisée des systèmes de communication.

Par leur construction, les schémas basés sur l'identité (IBE) parviennent à résoudre les problèmes liés à l'authentification, à la gestion et à la révocation des clés publiques,

néanmoins ils nécessitent la présence d'une autorité ultra-puissante PKG (Private Key Generator) qui fournit à chaque utilisateur la clé privée correspondante à sa clé publique. La confiance accordée à cette autorité doit être sans faille, car elle est intrinsèquement capable de regénérer la clé privée de tout utilisateur, et par conséquent capable de réaliser sans autorisation des signatures ou des déchiffrements.

Depuis leur introduction, les schémas basés sur l'identité ont fait l'objet de recherches intensives, pourtant il aura fallu attendre 2001, et les travaux de Cocks [Cocks, 2001] et de Boneh et Franklin [Boneh and Franklin, 2001], pour avoir des cryptosystèmes qui répondent relativement aux conditions de sécurité et d'efficacité. Même s'il repose sur un problème mathématique réputé difficile à résoudre, à savoir l'extraction des racines modulaires, le schéma de Cocks est très peu efficace car il chiffre les messages bit par bit. Jusqu'à maintenant plusieurs schémas basés sur l'identité ont été proposés mais dans ce mémoire, nous utilisons le schéma de BLMQ [Barreto, 2002] qui est pleinement opérationnel et qui réduit le nombre des opérations d'accouplement nécessaire pour la signature et la vérification.

4.3.2.1 Fonctionnement des IBE

Pour illustrer concrètement le fonctionnement des IBE, nous allons supposer qu'il a été convenu de choisir comme clé publique l'adresse URI SIP des utilisateurs. Pour envoyer un message à Bob, Alice va chiffrer son message en utilisant la clé publique Bob@proxysip.com une fois le message chiffré reçu, Bob s'authentifie auprès de l'autorité Private Key Generator, en procédant de la même manière qu'auprès d'une autorité de certification, afin d'obtenir sa clé secrète et de pouvoir ainsi déchiffrer le message qu'il vient de recevoir. La réalisation d'un schéma de chiffrement basé sur l'identité se déroule en quatre étapes figure 4.1:

- L'initialisation: le PKG crée une paire de clés : master (privé) et publique notées respectivement *skPKG* et *pkPKG*. Le *pkPKG* est fourni à toutes les parties intéressées et utilisé comme étant un paramètre constant du système pour une longue période.

- L'extraction de clé privée : Bob s'authentifie auprès du PKG et obtient la clé privée *skIDBob* associée à son identité *IDBob*.

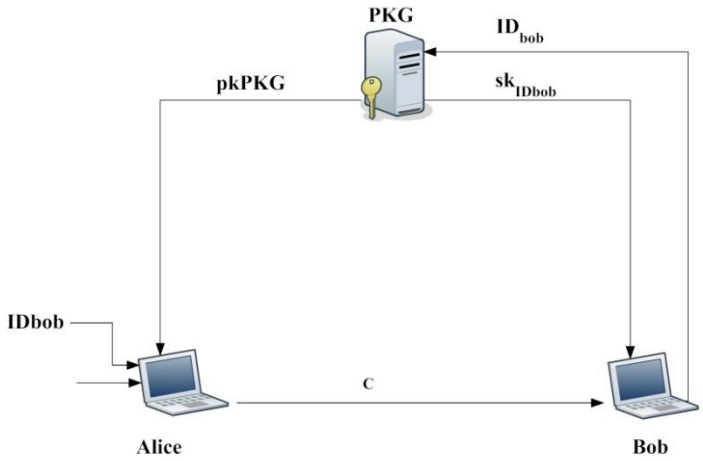


Figure 4-1 Fonctionnement de PKG

- Chiffrement : utilisant l'identité de Bob ID_{Bob} et le *pkPKG* obtenu à partir du PKG, Alice chiffre le message en clair M et obtient le message chiffré C.
- Déchiffrement : recevant le message C, chiffré par Alice, Bob le déchiffre en utilisant sa clé privée *skIDBob* pour reconstituer le message M.

4.3.3 L'application de la signature basée sur l'identité au LCA

L'absence d'un champ qui indique l'URI SIP de source impose au destinataire d'accepter l'appel entrant sans connaitre l'identité de l'appelant, pour identifier l'émetteur de messages. Nous ajoutons un champ qui spécifier son URI SIP, qui sert aussi comme une clé publique. Pour éliminer l'usurpation de l'identité et empêcher la modification des messages, l'émetteur signe le message en utilisant la signature basée sur l'identité. Avant de décrire le schéma de Secure_LCA, nous présentons un préliminaire mathématique:

Soient $G1$ et $G2$ deux groupes cycliques d'ordre p, un grand nombre premier. Le groupe $G1$ est muni d'une loi de groupe additive et $G2$ d'une loi de groupe multiplicative.

Les applications bilinéaires utilisées sont des applications définies par $e : G1 \times G1 \rightarrow G2$ et qui vérifient les propriétés suivantes :

- Bilinéaire : $e(aP, bQ) = e(P, Q)^{ab}$ pour tous entiers a, b et tous $(P, Q) \in G_1 \times G_1$.
- Non-dégénérée : $e(P, Q) = 1_{G2}$, pour tout $Q \in G_1$ si et seulement si $P = 0_{G1}$ Ainsi si P est un générateur de $G1$, l'élément $e(P, P)$ est alors un générateur de G_2.

Calculable : pour tout $P, Q \in G_1$, $e(P, Q)$ peut être calculé efficacement.

Dans la pratique, G_1 correspond à un sous groupe d'une courbe elliptique $E(\mathrm{F}q)$ constituée par l'ensemble des points de la courbe d'ordre p, noté par $E[p]$, et $G2$ correspond au groupe des racines-$p^{ème}$ de l'unité, noté μ_p.

Dans ce qui suit, on adapte les notations suivantes :

Tableau 4-1 Notation

Symbole	Signification
Q_s	Point dans la courbe elliptique qui correspond a l'URI du nœud S .
S_{URI}	Clé publique du nœud S
d_s	Clé privé du noud S
S	Nœud émetteurs
N	Nœud destinataire
$Sign_{d_s}[M]$	Signer M avec la clé privée du nœud source en utilisant l'algorithme de signature
$Verif_{S_{URI}}(Sign_{d_S}[M])$	Vérifier la signature du message M avec la clé publique de source en utilisant l'algorithme de vérification
$\sigma_{SIPRREQ}$	Signature de message SIPRREQ
P_{pub}	Clé publique de PKG

4.3.3.1 Le schéma de Secure_LCA

Nous proposons dans notre schéma que le PKG soit géré par une autorité digne de confiance qui s'occupe par le calcul de paramètres du système, la pré-distribution des adresses URIs SIP, les paramètres généraux du système, et les clés privées associées. Chaque utilisateur possède donc une adresse URI SIP unique qui a la forme d'une adresse mail (ex : user@proxysip.com). On note que dans le cas où un utilisateur possède une connexion internet, il a toujours la possibilité de s'enregistrer auprès d'un serveur SIP externe en utilisant son URI SIP. Le mécanisme de signature utilisé dans Secure_LCA se décompose en 4 algorithmes (Mise en place, extraction, signature et vérification) :

1) Mise en place : l'autorité doit préparer et déterminer les paramètres généraux du système. Cet algorithme sert à construire des groupes, des fonctions de hachage et il calcul aussi le master key et la clé publique qui constitueront le système de signature, nous résumons ci-dessous les différentes étapes de calcul de ces paramètres :

1: prendre comme entrée un paramètre de sécurité K.

2: générer un nombre premier q de K, deux groupes $(G_1, +)$, $(G_2, *)$ d'ordre q, un couplage $\hat{e}: G_1 \times G_1 \rightarrow G_2$, et un générateur $P \in G_1$.

3: Choisir aléatoirement la clé maître $z \in \mathbb{Z}_q^*$.

4: Calculer $P_{pub} = zP$.

5: Choisit deux fonctions de hachage cryptographiques $H_1 : \{0,1\}^* \rightarrow G_1^*$, et $H_2: \{0,1\}^* \times G_1 \rightarrow \mathbb{Z}_q^*$

L'espace des messages clairs est $m = \{0,1\}^n$ et l'espace des messages chiffrés est $\check{C} = G_1^* \times \{0,1\}^n$.

Les paramètres du système sont $params = \langle q, G_1, G_2, \hat{e}, n, P_{pub}, H_1, H_2 \rangle$, le master key est $z \in \mathbb{Z}_q^*$.

2) Extraction : cet algorithme est utilisé par le PKG pour dériver la clé privée d'un utilisateur S de sa clé publique S_{URI} . Pour une chaîne de caractères donnée $URI \in \{0,1\}^n$:-

- Le PKG transforme URI en un point Q_s de la courbe elliptique tel que $Q_s = H_1(URI) \in G_1^*$.
- Génère la clé secrète de l'utilisateur associé à URI : $d_S = z.Q_s$.

3) Signature : l'émetteur S signe un message M avec sa clé privée d_s par le calcul de $U = rQ_s$ où $Q_s = H_1(S_{URI})$, et $V = (r + h)\, d_s$ où r un entier choisi aléatoirement et $h = H_2(M, U)$. La signature est : (U, V).

4) Vérification : le destinataire N peut vérifier la validité de la signature (U, V) en vérifiant si $\hat{e}(P, V) = \hat{e}(P_{pub}, U + hQ_s)$.

4.3.3.1.1 L'établissement de session dans Secure_LCA

Lorsque un nœud S a besoin de la liaison d'un nœud destination N, il initie un message de demande de route (SIPRREQ) représenté dans la figure 4.2, le même que celui de LCA avec l'ajout de deux nouveaux champs ($Source\ SIP\ URI$, et $\sigma_{SIPRREQ}$).

$SIPRREQ\ ID$
$Target\ SIP\ URI$
$Source\ SIP\ URI$
Source IP Adress
$\sigma_{SIPRREQ}$

Figure 4-2 Format de message SIPRREQ de Secure_LCA

Le premier champ $Source\ SIP\ URI$ représente l'URI SIP du nœud S notamment sa clé publique S_{URI} , et ensuite le nœud S signé également avec sa clé privé d_s le message $SIPRREQ$ avec l'algorithme de signature pour cela il calcule $Sign_{d_s}[SIPRREQ]$, et ajoute la signature $\sigma_{SIPRREQ}$ au paquet $SIPRREQ$, et puis le nœud S envoie le message en broadcast,

n'importe quel nœud intermédiaire, par exemple X, a la réception il vérifie la signature $\sigma_{SIPRREQ}$ en calculant $Verif_{S_{URI}}(\sigma_{SIPRREQ})$ ce n'est pas valide ou si elle a déjà vu ce paquet en reconnaissant la combinaison du $\langle SIPRREQ\ ID, SOURCE\ IP\ ADRESS \rangle$ il rejette le paquet, sinon et si la signature est valide le nœud X compare le champ $Target\ URI\ SIP$ avec son $URI\ SIP$ s'il est différent il retransmet le paquet en broadcast dans le cas contraire, le nœud X est le destinataire N, le nœud N répond en unicast au S par la génération d'un message $SIPRREP$, il recopier les champs suivants le «$Target\ SIP\ URI$ », « $SIPRREQ\ ID$ » et « $Originator\ IP\ Address$ » du message $SIPRREQ$, est signé le message $SIPRREP$ avec sa clé privée d_N par le calcul de $Sign_{d_s}[SIPRREP]$ et ajoute la signature $\sigma_{SIPRREP}$ au message $SIPRREP$, Le format du message $SIPRREP$ est représenté dans la figure 4.3.

$SIPRREQ\_ID$
$Target\ SIP\ URI$
$Source\ SIP\ URI$
Source IP Adress
Target IP Adress
$\sigma_{SIPRREP}$

Figure 4-3 Format du message SIPRREP de Secure LCA

L'émetteur à la réception du message $SIPRREP$ vérifié la signature de ce dernier. Si elle est valide il l'accepte et il peut procède par l'envoie d'un message INVITE au destinataire N, et dans le cas contraire il le rejette.

4.3.3.1.2 Analyse de la sécurité de Secure_LCA

Le schéma Secure_LCA, présente un certains nombre d'avantages et d'inconvénients. En effet, le champ $SOURCE\ URI\ SIP$ qu'on a ajouté sert à identifier l'émetteur et de ce fait, le destinataire peut savoir de qui il vient l'appel, Cependant il peut vérifier l'existence de ce dernier dans son carnet d'adresse et décide de répondre ou de rejeter l'appel.

Le problème de l'usurpation de l'identité a été résolu par l'introduction des signatures numériques. Ces dernières assurent l'authentification de bout en bout et l'intégrité des messages transmis. Dans le cas d'une attaque externe où un nœud ne possède pas une paire de clé valide il ne peut pas fabriquer des messages ou usurper l'identité des nœuds légitimes ou falsifier des messages, aussi les nœuds légitimes ne peuvent pas modifier le contenu du champ d'un message ou le supprimer car l'attaquant ne peut pas générer une signature valide sans la possession de clé privée de la source du message.

4.4 Conclusion

La solution de déploiement du SIP dans les réseaux Ad hoc comme LCA est sujet à plusieurs types d'attaques. A travers la modification ou la fabrication des messages de contrôle ou l'usurpation de l'identité, un attaquant peut facilement perturber le processus de découverte des liaisons des nœuds et influencer la signalisation. La nouveauté dans la solution que nous avons proposé dans Secure_LCA est l'utilisation de la signature basée sur l'identité qui utilise les adresses des nœuds URIs SIP comme des clés publiques pour assurer l'authentification et l'intégrité d'une manière très simplifiée, et qui permettent à la solution d'être plus efficace et plus rapide. Ainsi le schéma proposé donne un niveau de confiance élevé à un coût très réduit.

Conclusion générale

La sécurité des réseaux mobiles et spécialement des réseaux mobiles ad hoc n'ont jamais cessé de susciter des préoccupations du fait qu'ils sont exposés à des menaces supplémentaires par rapport aux réseaux filaires. En général, ces menaces viennent du fait que les communications sans fil sont transmises par ondes radios et peuvent être écoutées par des personnes non autorisées.

Les techniques de chiffrement les plus utilisées dans les systèmes filaires ne sont pas toujours convenables pour les systèmes sans fils vu leurs caractéristiques limitées (puissance de calcul, capacité de stockage, bande passante).

L'utilisation des schémas cryptographiques nécessite l'utilisation d'un service de gestion de clés. Pour cela, il existe des procédures basées sur une infrastructure à clé publique permettant la gestion de clés. Cette solution est à moitié satisfaisante car elle suppose l'existence d'une infrastructure centralisée alors que le concept de réseaux ad hoc appelle à une infrastructure distribuée. Par conséquent, les mécanismes de sécurité devront être distribués.

Il est présenté dans ce travail les différentes approches de décentralisation de SIP dans les réseaux Ad hoc. La plupart de ces solutions utilisent des certificats numériques pour garantir l'authentification et l'intégrité dans la phase d'établissement de session. Ces mécanismes de sécurité sont complexes et nécessitent une grande puissance de calcul. Les solutions de décentralisation de SIP qui utilisent les techniques de protocoles de routage sont proposées sans aucun mécanisme de sécurité. Pour cela, dans ce mémoire, nous avons exploité l'efficacité et la rapidité de la signature basée sur l'identité pour proposer une solution au problème d'authentification et d'intégrité de LCA. La signature basée sur l'identité réduit considérablement la complexité et le coût de l'établissement et la gestion des clés. Par leur construction, les schémas basés sur l'identité parviennent à résoudre les problèmes liés à l'authentification, à la gestion et à la révocation des clés publiques.

Le schéma Secure_LCA que nous avons proposé montre un gain important en consommation d'énergie et en temps d'exécution, ce qui correspond mieux aux caractéristiques limitées des réseaux ad hoc.

Bibliographie

[Althouse, 1999]: E. Althouse, “Extending the Littoral Battlespace (ELB), Advanced Concept Technology Demonstration (ACTD)”, NATO Information Systems Technology Panel Symposium on Tactical Mobile Communications, Juin 1999.

[Baldoni et al , 2006]: R. Baldoni, G. Cortese, F. Davide, and A. Melpignano ed. Epidemic Dissemination for Probabilistic Data Storage, chapter of Global Data Management. IOS, July 2006.

[Banerjee et al, 2006]: N.Banerjee, A. Acharya et S. K. Das, "Enabling SIP-Based Sessions in Ad Hoc Networks", Winet, Avril 2006.

[Barreto, 2002]: P. S. L. M. Barreto, H. Y. Kim, B. Lynn and M. Scott, “Efficient Algorithms for Pairing-Based Cryptosystems,” Proc. Advances in Cryptology -- Crypto'02, pp.354-368, 2002.

[Boneh and Franklin, 2001]: D. Boneh, M. Franklin: Identity-Based Encryption from the Weil Pairing, Proceedings of CRYPTO 2001, LNCS 2139, Springer-Verlag, 2001.

[Borg, 2003]: J.Borg “A Comparative Study of Ad Hoc & Peer to Peer Networks, Master’s thesis, University College London, AUGUST 2003.

[Bryan et al, 2005]: D. A. Bryan, B. B. Lowekamp and C. Jennings, SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System First International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA'05) IEEE, Orlando, USA, 2005, pp. pp. 42-49.

[Bryan et al, 2008]: D. Bryan, P.Mathews, E. Shim, D. Willis, S. Dawkins, “Concepts and Terminology for Peer to Peer SIP,” draftietf-p2psip-concepts-02, July, 2008.

[Camara, 2000]: D. Camara and A.F. Loureiro, “A novel Routing Algorithm for Ad hoc networks”, In Proc. HICSS, Hawaii, 2000.

[Campbell et al, 2002]: B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema et D. Gurle, "session initiation protocol (SIP) extension for instant messaging", RFC 3428, Décembre 2002.

[Chang, 2004]: Chang Lin-huang, Chuang Ping-da, and Chen Yu-Jen. An ad-hoc voip system implementation using UPnP. In International Computer Symposium, December 2004.

[Chaudet, 2002]: C. Chaudet and I. Guérin Lassous, “Bruit: Bandwidth reservation under interferences influence”. In European Wireless 2002 (EW 2002), pages pp. 466–472, Florence, Italy, Février 2002.

[Chiang et al, 1997]: C. Chiang, H. Wu, W. Liu, and M. Gerla. “Routing in Clustered Multihop, Mobile Wireless Networks”, In Mobile Wireless Networks, the IEEE Singapore International Conference on Networks, 1997.

[Clausen, 2003]: T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR): Request for Comments”, 3626, Octobre 2003.

[Cocks, 2001]: C. Cocks, An Identity Based Encryption Scheme Based on Quadratic Residues, Cryptography and Coding - Institute of Mathematics and Its Applications International Conference on Cryptography and Coding {Proceedings of IMA 2001, LNCS 2260, pages 360{363, Springer-Verlag, 2001.

[Corson, 1999]: S. Corson and J. Macker, “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”, Request for Comments 2501,

IETF, Janvier 1999.

[Corson, 2002]: S. Corson, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501.

[Dierks and Allen, 1999]: T. Dierks and C. Allen. The TLS Protocol Version 1.0. RFC 2246 (Proposed Standard), January 1999. Updated by RFC 3546.

[Douara, 2012a]: B.N. Douara, K. Belkadi, S. MAMMAR, "SIP Security in Wireless Ad-hoc Network ", International Conference on Informatics and Applications, University Sultan Zainal Abidin, Malaysia, ICIA2012, June 3-5, Malaysia.

[Douara, 2012b]: B.N. Douara, K. Belkadi, S. MAMMAR, «Secure Multimedia Session Establishment in Wireless Ad-Hoc Networks ", 8th International Conference on Digital Content, Multimedia Technology and its Applications, IDCTA2012, June 26 - 28, Jeju, Korea.

[Dube, 1997]: R. Dube, C. Rais, K. Wang, and S. Tripathi, "Signal stability based adaptive routing (SSA) for Ad hoc mobile networks. In Signal stability based adaptive routing (SSA) for Ad hoc mobile networks", IEEE Personal Communication, Février 1997.

[Singh and Schulzrinne, 2005]: K. Singh and H. Schulzrinne, Peer-to-peer internet telephony using SIP, Proceedings of the international workshop on Network and operating systems support for digital audio and video, ACM, Stevenson, Washington, USA, 2005, pp. 63-68.

[Fifer et al, 1987]: W. Fifer, F. Bruno, "The low-cost packet radio", Proceeding of the IEEE 75 (1), pp. 33-42, 1987.

[Franks et al, 1999]: J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Draft Standard), June 1999.

[Freebersyser et al, 2001]: J.A. Freebersyser, B. Leiner, "A DoD perspective on mobile Ad hoc networks", Ad Hoc Networking, Addison Wesley, pp. 29-51, 2001.

[Giordano, 2001]: S. Giordano, I. Stojmenovic and L. Blazevie, "Position based routing algorithms for Ad hoc networks: a taxonomy", Juillet 2001. http://www.site.uottawa.ca/-ivan/routing-survey.pdf.

[Guttman, 1999] : E. Guttman, C. Perkins, J. Veizades, and M. Day. Service Location Protocol, Version 2. RFC 2608 (Proposed Standard). Updated by RFC 3224. Juin 1999.

[Haas et al, 2002]: Z. J. Haas, J.Y. Halpern, and L. Li. Gossip-based ad hoc routing. In INFOCOM '02: Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, volume 3, pages 23.27, June 2002.

[Haas, 2002]: Z. Haas, M. Pearlman and P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", http: //www.draft-ietf-manet-zone-zrp-02.txt, Juillet 2002.

[Housley, 2002]: R. Housley, W. Polk, W. Ford, and D. Solo. RFC 3280 – Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF, April 2002.

[James, 2000]: James Kardach" Bluetooth Architecture Overview, Mobile Computing Group, Intel Corporation, 2000.

[Jennings, 2007]: C.Jennings. Security mechanisms for Peer to Peer SIP. http://tools.ietf.org/wg/p2psip/draft-jennings-p2psip-security-00.txt. February 2007.

[Jennings, 2008]: C. Jennings, B.Lowekamp, E. Rescorla, S. Baset, H. Schulzrinne,

"REsource LOcation And Discovery (RELOAD)," draft-bryan-p2psip-reload-04, June, 2008.

[Jiang et al, 1999]: M. Jiang, J. Li and Y.C. Tay, "Cluster Based Routing Protocol (CBRP)," IETF Internet Draft draft-ietf-manet-cbrp-spec-01.txt, August 1999.

[Jini, 2003]: Jini™ Architecture Specification, Sun Microsystems, Version 2.0, juin 2003.

[Johnson, 2004]: D. Johnson, D.A. Maltz, Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", Internet Draft: http: //www.draft-ietf-manetdsr-10.txt, 19 Juillet 2004.

[Kang, 2002]: I. Kang and R. Poovendran, "On Lifetime Extension and Route Stabilization of Energy- Efficient Broadcast Routing over MANET", In Proceedings of INC 2002, Plymouth, 2002.

[Karpijoki, 2000]: V. Karpijoki. "Security in Ad Hoc Networks", Seminar on network security, In Proceedings of the Helsinki University of Technology, 2000.

[Kent and Atkinson, 1998]: S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401 (Proposed Standard), November 1998. Updated by RFC 3168.

[Leggio et al, 2005]: S. Leggio, J. Manner, A. Hulkkonen, et K. Raatikainen, "Session initiation protocol deployment in ad-hoc networks: a decentralized approach", In 2nd International Workshop on Wireless Ad-hoc Networks (IWWAN), London, May, 2005.

[Banerjee et al, 2006]: N.Banerjee, A. Acharya et S. K. Das, "Enabling SIP-Based Sessions in Ad Hoc Networks", Winet, Avril 2006.

[Leiner et al, 1996]: B. Leiner, R. Ruth and A.R. Sastry, "Goals and challenges of the DARPA GloMo program", IEEE Personal Communications, pp. 34-43, 1996.

[Levis et al, 2004]: P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In Proc. of the 1st USENIX/ACM Symp. On Networked Systems Design and Implementation, 2004.

[Marwaha, 2010]: Marwaha Richa, "Security in Peer-to-Peer SIP VoIP". Master's Projects http://scholarworks.sjsu.edu/etd_projects/160 Septembre 2010.

[Matuszewski et al, 2007]: M. Matuszewski, J-E. Ekberg and P. Laitinen. Security requirements in P2PSIP, February 2007. http://tools.ietf.org/wg/p2psip/draft-matuszewski-p2psip-security-requirements-00.txt.

[Miranda et al, 2006]: H. Miranda, S. Leggio, L. Rodrigues, and K. Raatikainen. A poweraware broadcasting algorithm. In 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2006 11-14 September, Helsinki, Finland.

[Ogier et al, 2002]: R. Ogier, F. Templin and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)", IETF RFC 3684.

[Park, 1997]: V. D. Park and M. Scott Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", In INFOCOM (3), pages 1405–1413, 1997.

[Perkins, 2003]: C.E. Perkins, E.M. Belding-Royer et S. Das. "Ad Hoc On Demand Distance Vector (AODV) routing", RFC 3561. July 2003.

[Peterson and Jennings, 2006]: J. Peterson and C. Jennings. Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP). RFC 4474 (Proposed Standard), August 2006.

[Ramsdell, 2004]: B. Ramsdell. Secure/Multipurpose Internet Mail Extensions (S/MIME)

Version 3.1 Message Specification. RFC 3851 (Proposed Standard), July 2004.

[Ratnasamy et al, 2001]: S.Ratnasamy, P.Francis, M.Handley, R.Karp, and S.Shenker. A scalable Content-Addressable Network. In ACM SIGCOMM 2001, San Diego, CA (USA), August 2001.

[Royer, 1999]: E. Royer and C. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks", In IEEE Personal Communications, Avril 1999.

[Shamir, 1984]: A. Shamir. Identity-Based Cryptosystems and Signature Schems. Crypto84, vol. 196 , LNCS, pp. 47-53, 1984.

[Singh and Schulzrinne, 2005]: K. Singh and H. Schulzrinne, Peer-to-peer internet telephony using SIP, Proceedings of the international workshop on Network and operating systems support for digital audio and video, ACM, Stevenson, Washington, USA, 2005, pp. 63-68.

[Stajano, 1999]: F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks", 7th International Workshop on Security Protocols, pp. 172-194, 1999.

[Chaudet, 2002]: C. Chaudet and I. Guérin Lassous, "Bruit: Bandwidth reservation under interferences influence". In European Wireless 2002 (EW 2002), pages pp. 466–472, Florence, Italy, Février 2002.

[Stoica et al, 2001]: I. Stoica, R. Morris, D.Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: A scalable Peer-to-Peer lookup service for internet. In ACM SIGCOMM 2001, San Diego CA, August 2001.

[Stuedi, 2007]: P. Stuedi, M. Bihr, A. Remund, et G. Alonso, "Siphoc: Efficient sip middleware for ad hoc networks" stuedi2007siphoc, LECTURE NOTES IN COMPUTER SCIENCE, Springer, 2007.

[Theoleyre, 2006]: F. Theoleyre, "Une auto-organisation et ses applications pour les réseaux Ad hoc et hybrides", Thèse de doctorat, Institut national des sciences appliquées de Lyon-France, Septembre 2006.

[Toh, 1996]: C-K Toh, "A Novel Distributed Routing Protocol to Support Ad-Hoc Mobile Computing". In IEEE 15th Annual Int'l, Phoenix Conf. Comp. and Commun, 1996.

[UPnP Forum, 2008]: Universal Plug and Play Forum. UPnP TM device architecture, October 2008.

[Westcott et al, 1984]: J. Westcott and G. Lauer, "Hierarchical routing for very large networks", Proceeding of the IEEE MILCOM 1984, pp. 214-218, 21-24 Octobre 1984.

[Zheng and Vladimir, 2009]: X. Zheng, O.Vladimir, "Improving Chord lookup protocol for P2PSIP-based Communication Systems," International Conference on New Trends in Information and Service Science (3rd NISS), June, 2009.