République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bouguarra de Boumerdès

Faculté des Sciences
Département d'Informatique

Mémoire

Pour l'obtention du diplôme de
Magistère en Informatique
Option

Spécification des logiciels et traitement de l'information

Présenté et soutenu par

M. OUTIOUA Mustapha Idir

Proposé par

Dr. Hakima Chaouchi : Maître de Conférence à l'Institut National des Télécommunications, France

Titre

Une architecture optimisée pour le support de services d'utilisateurs mobiles

"Optimized IP based architecture for mobile users' service support"

Soutenu le 26 juin 2008 devant le jury composé de

Pr. Mohamed Mezghiche	Professeur	UMBB	Président
Dr. Hakima Chaouchi	Maître de conférence	INT-F	Promotrice
Dr. Amar Balla	Maître de conférence	INI	Examinateur
Dr. Rachid Chalal	Maître de conférence	INI	Examinateur

Résumé :

Par son succès, l'Internet a permis l'émergence de nouveaux concepts à introduire dans son réseau. Parmi ces concepts on distingue la gestion de la mobilité qui est due à l'émergence des réseaux sans fil et la gestion de la qualité de service dans le but d'opérer de nouvelles applications sur ce réseau. Les premières solutions de la gestion de la mobilité consistaient seulement à maintenir la connexion du mobile pendant que celui-ci change de point d'attachement au réseau. Cette solution ne tient pas compte de la gestion de la qualité de service offerte par le réseau au nouveau point d'attachement. Dans un réseau mobile offrant une qualité de service, la durée nécessaire à l'accomplissement de la nouvelle connexion est importante, cependant, si le mobile doit s'authentifier et si le réseau doit aussi installer la qualité de service à chaque changement de point d'attachement, la niveau de service se voit influencée par cette durée.

Afin de montrer la difficulté de la gestion de la qualité de service et de l'authentification tout en assurant la gestion de la mobilité dans les réseaux sans fil, une étude sur les trois types d'architectures avec leurs protocoles respectifs et l'interaction entre elles a été faite. Cette étude concerne initialement l'influence de la mobilité sur le niveau de service assuré au mobile. Nous avons identifié les principaux problèmes qui surgissent quand on exploite la qualité de service dans la mobilité IP comme la dégradation du niveau du service. Aussi, les réseaux sans fil mobiles rendent la sécurité difficile et délicate et l'authentification dans ce type de réseaux influence indirectement le niveau de service.

Ces incommodités sont dues essentiellement au fait que ces protocoles et ces architectures ont évolué séparément. Par conséquent, la combinaison de la gestion de la mobilité, de la sécurité et de la qualité de service est donc essentielle laquelle est une tâche difficile à résoudre. Le concept de la signalisation unifiée introduit dans ce mémoire vise à trouver une solution à ce pari. La solution proposée dans notre travail vise principalement à optimiser la durée nécessaire au mobile à se reconnecter tout en assurant l'authentification du mobile et l'installation de sa qualité de service dans le réseau en tenant compte de sa nouvelle position. Notre objectif tend à concevoir une infrastructure supportant un protocole unifiant la signalisation de la qualité de service, de la mobilité et de la sécurité.

Grâce à des simulations, nous avons analysé le comportement du nouveau protocole et l'avons comparé aux protocoles existants exécutés séparément. Il s'agit d'abord de connaître les limites de la nouvelle architecture et du nouveau protocole ainsi que les avantages et les inconvénients des deux types d'architecture, ensuite d'analyser le comportement des deux protocoles sur la même architecture pour savoir quel protocole pourrait le mieux satisfaire le but principal qui est l'amélioration de la qualité de service dans un réseau IP sans fil et mobile.

Nos conclusions mettent en évidence les avantages d'une signalisation unifiée en soulevant les problèmes qu'il faut éviter lors de la conception de ce genre d'architecture. Ainsi, il a été démontré que sur le plan de la qualité de service de ce type de signalisation par rapport à la signalisation indépendante, les profits sont considérables. Enfin, ce travail nous a permis de proposer des ouvertures sur d'autres champs d'étude pour l'architecture unifiée afin d'améliorer ce concept fort prometteur.

Mots clé : Gestion de la QoS, gestion de la mobilité, AAA, COPS, DiffServ, RSVP, EAP, IEEE802.1X, Diameter, interaction Mobilité et QoS, interaction la AAA et la Mobilité.

Remerciements

Je tiens à exprimer toute ma gratitude à toutes les personnes qui m'ont aidé à accomplir cette étude :

- Le Dr. Hakima Chaouchi et le Pr. Mezghiche Mohamed pour leur contribution scientifique ;

- Ma mère, mon père, mes frères et mes sœurs ainsi que mes amis pour leur soutien moral appréciable à plus d'un titre.

Table des matières

Chapitre I : Problématique et objectifs.

Le réseau Internet a évolué du contexte fixe vers le sans fil et mobile. Cela ne se fait pas sans poser plusieurs problèmes de recherche liés aussi bien à la gestion de la qualité de service (QoS), la mobilité et la sécurité. En effet, l'Internet initial était conçu juste pour assurer le routage de paquets dans un réseau fixe, tous les problèmes de QoS, mobilité et sécurité sont survenus après son premier déploiement qui n'était pas du tout commercial. Plusieurs travaux de recherche ont été menés pour rendre l'Internet mobile possible. Entre autre les travaux qui sont exposés dans [58] constituent la base de notre travail. Ces travaux concernent les problèmes de support des utilisateurs mobiles dans un réseau de type Internet en prenant en considération les aspects QoS et sécurité et en particulier l'authentification. L'un des problèmes posés par l'auteur dans [58] est la signalisation unifiée de la mobilité, QoS, et sécurité. En effet, l'auteur défend la thèse qu'une signalisation unifiée de la mobilité, QoS, et sécurité est meilleure qu'une signalisation faite de manière séparée ou indépendante dans le sens ou chacune s'exécute séparément. Par exemple, une signalisation de mobilité s'exécuterait sans considération de l'aspect QoS. Dans ce cas un utilisateur mobile pourrait se voir connecté à un nouveau point d'accès du réseau qui ne pourrait pas satisfaire ses besoins en QoS. Notre travail ici s'intéresse précisément dans cette problématique de signalisation unifiée proposée dans [58]. Avant de présenter notre solution qui consiste en une architecture unifiée et la simulation dans le quatrième et le cinquième chapitre, un état de l'art liés aux aspects QoS, mobilité et sécurité dans l'Internet est présenté dans le deuxième chapitre 2, puis l'interaction entre ces aspects est présentée dans le troisième chapitre.

Dans ce travail, notre défi est de concevoir une infrastructure supportant un protocole unifiant la signalisation de la QoS, de la mobilité et de la sécurité. Ce protocole que nous allons introduire permettra la signalisation simultanée ou quasi-simultanée des informations transportées par les différents protocoles pour exécuter un Handover rapide. L'architecture proposée doit être flexible pour supporter les besoins d'une nouvelle intégration d'un protocole, d'une nouvelle fonction ou d'une nouvelle extension du protocole de signalisation. Notons aussi que, dans cette architecture, nous nous intéressons à l'aspect d'authentification et d'autorisation de la signalisation AAA qui est étroitement lié au processus de la gestion de la mobilité. La signalisation de comptabilité n'est pas vraiment prise en considération parce qu'elle n'est pas exécutée au moment de l'interruption et par conséquent elle n'a pas d'impact direct sur le Handover. Le but fondamental de cette architecture est d'avoir une authentification et une installation de la QoS autorisée en même temps que la mise à jour de la localisation au moment de l'exécution du Handover par le mobile et ce afin de réduire l'interruption.

D'autre part, la liaison sans fil utilisée dans les réseaux mobiles a une basse bande passante et un taux d'erreur élevé par rapport aux liaisons filaires. De ce fait, l'architecture unifiée doit tenir compte du nombre de messages de signalisation envoyés sur la liaison sans fil. Ce nombre doit être minimisé et la taille des messages du protocole doit rester modestement petite. De plus, les nœuds mobiles fonctionnent vraisemblablement avec des batteries et minimiser la consommation de l'énergie est important ; cette économie doit être aussi prise en considération dans cette architecture.

Chapitre II

Etat de l'art

1. Introduction :

Le concept original et exceptionnel du protocole IP a engendré un succès planétaire du réseau Internet dans toutes les pratiques de notre société d'aujourd'hui que ce soit dans le domaine économique, administratif, scientifique, culturel ou dans bien d'autres activités. La communauté des chercheurs essaie constamment d'améliorer le réseau Internet et de l'adapter efficacement aux utilisations contemporaines. L'un des concepts introduit préalablement dans les réseaux fixes est celui des architectures de la Qualité de Service qui visent à améliorer le service offert à l'utilisateur. L'autre est celui des architectures AAA qui concernent la sécurité et la gestion des informations de configuration des réseaux et la facturation du service. Un autre concept introduit, dû principalement à l'émergence de la technologie sans fil, est celui de la gestion de la mobilité qui permet à l'utilisateur de rester connecté et d'utiliser sans contrainte le service offert tout en se déplaçant.

Ce chapitre se compose essentiellement de trois parties. Chaque partie analyse les architectures et les protocoles de chaque concept introduit ci-dessus. La première concerne les architectures de la Qualité de Service, suivie de l'architecture AAA et la dernière partie se rapporte aux architectures de la gestion de la mobilité.

2. Gestion de la Qualité de Service :

L'Internet est un réseau à commutation de paquets. Quand un paquet est envoyé dans le réseau, il pourrait atteindre sa destination avec le seul temps de propagation comme il pourrait soit l'atteindre avec un temps beaucoup plus long soit rejeté dans le cas d'une forte congestion. Le réseau ne fait aucune supposition sur le délai de délivrance du paquet, sur le débit disponible ou encore sur les pertes de paquets ; On dit que le réseau fournit un service au mieux (Best-Effort, BE). Il ne différencie pas les flots de données issus des différentes applications ou des différents utilisateurs.

Le concept de Qualité de Service (Quality of Services : QoS) apporte une nouvelle vision du réseau Internet. L'introduction de la QoS dans le réseau permet d'offrir, en plus du service Best-Effort, d'autres services qui permettraient aux applications telles que la téléphonie, la vidéo à la demande ou la conférence multimédia de s'exécuter correctement au travers de l'Internet. L'objectif de la QoS est donc de fournir aux clients (utilisateurs) des services de communications ayant des exigences qualitatives spécifiques. La QoS apporte également un intérêt vital au fournisseur de service Internet. Elle permet d'offrir des niveaux de services différents pour ainsi augmenter les bénéfices financiers des opérateurs selon les besoins des clients.

Dans cette section, après avoir défini la QoS et ses paramètres, nous présenterons les architectures fondamentales les plus connues de la QoS. Nous commencerons à étudier l'architecture IntServ qui est la première architecture proposée par Internet Engineering Task Force (IETF)[1] ensuite nous étudierons l'architecture DiffServ dont la conception est focalisée particulièrement pour pallier le problème de passage à l'échelle (Scalability) posé par IntServ. Enfin, un protocole de gestion de la Qualité de Service appelé COPS sera présenté.

2.1. Qualité de Service :

Un réseau fournit la QoS s'il est capable de spécifier des services adéquats pour des applications particulières selon certains paramètres inhérents aux caractéristiques des réseaux afin d'assurer le bon fonctionnement de ces applications. D'autre part, le réseau Internet est connu comme étant un ensemble de réseaux reliés pouvant être administrativement indépendants qui coopèrent pour assurer la communication entre les nœuds. Dans ce cas, la QoS n'est convenablement assurée que si les réseaux qui sont traversés par le flot d'une application assurent chacun la QoS ; on parle alors de la QoS de bout-en-bout.

Afin de concevoir et de réaliser une architecture de QoS, plusieurs paramètres non fonctionnels sont définis et pris en considération [13]. Ces paramètres sont soit orientés technologie, tels que le débit et le délai, soit orientés utilisateur, tels que la priorité et la confidentialité. Cependant, la QoS se décline principalement en quatre paramètres : le débit, la latence, la gigue et la perte de paquets. Ces paramètres sont définis dans ce qui suit :

Le débit (bande passante) : C'est le rapport entre le nombre de bits d'un flot transmis pendant un intervalle de temps et la durée de cet intervalle ; un flot étant un ensemble de paquets originaires d'une seule application exploitant une session donnée[2] qui requièrent la même QoS. Ce paramètre est évidemment limité à la bande passante du support de transmission mais il ne dépend pas que de ce support. Le traitement des paquets effectué dans les routeurs influence considérablement ce paramètre.

La latence (délai de délivrance) : C'est le délai de traversée du réseau d'un bout à l'autre par un paquet. La latence dépend du temps de propagation, du temps de traitement et de la taille des paquets.

La gigue (jitter) : Elle désigne les variations de latence des paquets du même flot. La présence de la gigue dans les flots peut provenir de changements d'intensité de trafic sur les liens de sorties des routeurs. Plus globalement, elle dépend du volume de trafic et du nombre d'équipements sur le réseau.

La perte de paquets : Elle correspond aux paquets éliminés (rejetés) ou perdus lors de la transmission d'un flot. Ce paramètre est souvent exprimé en taux de perte. La perte de paquets apparaît lorsque l'intensité du trafic sur les liens de sorties d'un nœud devient supérieure à sa capacité d'écoulement. Ce paramètre est donc une indication de la congestion.

On peut remarquer que ces quatre paramètres presque indépendants sont tous liés à la congestion. Dans le cas idéal où la congestion est absente, un flot utilise le débit adéquat qui n'est limité que par la bande passante du support de transmission ; il reçoit une latence minimale qui correspond au délai de propagation ; il réceptionne une gigue qui est nulle ; il n'aura aucune perte de paquets. Ce cas n'est pas possible puisque le réseau Internet est partagé par les applications et les paquets sont traités de la même façon quel que soit le niveau de service souhaité. La solution apportée par les architectures de la QoS qui seront présentées consiste à distinguer les flots pour leur fournir leurs traitements spécifiques.

[1] IETF : www.ietf.org
[2] Cette définition du flot reste la même pour l'étude des architectures de la QoS.

2.2. L'architecture IntServ et RSVP :

L'intégration de la QoS dans Internet en utilisant l'architecture IntServ « Integrated Services » vise à offrir une extension à l'architecture d'Internet pour ne pas perdre principalement les avantages incontestables de l'architecture actuelle d'Internet. L'idée est d'introduire, en plus du service Best-Effort traditionnellement offert par Internet, d'autres services qui vont permettre le bon fonctionnement de nouvelles classes d'applications. Cela veut dire transformer l'Internet en un réseau à intégrations de services.

2.2.1. Les classes d'applications :

Afin de guider la conception d'un modèle de service, une classification des types d'applications qui s'exécutent sur Internet est proposée dans [14][3]. Cette classification se base exclusivement sur le délai de délivrance de paquets (la latence). Son critère est le degré de la dépendance de l'application au délai de délivrance pour assurer son bon fonctionnement. On peut discerner deux classes :

La première est la classe des **applications temps-réel**. Ces applications ont besoin de réceptionner au bout d'un temps défini les données de chacun de leurs paquets sinon ces données seront probablement inutiles. Le délai maximum que le délai de délivrance de paquets ne doit pas dépasser est un paramètre important au bon fonctionnement de ces applications. Dans cette classe qui est très sensible à la latence deux types d'application peuvent être distingués.

Les applications intolérantes : Ces applications ne tolèrent pas la distorsion du signal et sont paramétrées avec un délai offset fixe. Ce délai doit être supérieur au délai maximum absolu de celui de la délivrance des paquets sinon le paquet n'est plus utilisable. Les exemples de ce type d'application sont la téléphonie où deux nœuds communiquent simultanément.

Les applications tolérantes : Ces applications peuvent tolérer un certain retard du délai de délivrance de paquet puisqu'ils s'adaptent en changeant des paramètres qui font varier leur délai offset. On les appelle aussi applications adaptatives. La lecture vidéo à distance est un bon exemple de ce type d'application.

La deuxième classe est la classe des **applications élastiques**. Ces applications n'utilisent pas immédiatement leurs données après leurs réceptions. En mettant en mémoire les données des paquets reçus, elles attendent la fin de la réception de toutes les données pour les utiliser. Cela ne signifie pas que ces applications sont insensibles au grand retard des paquets mais elles dépendent beaucoup plus de la latence moyenne et du débit. Les applications comme FTP, NFS, FAX et courrier électronique sont des exemples de cette classe.

2.2.2. Modèle de Service :

La précédente classification des applications a permis la construction d'un modèle de service dans IntServ à trois niveaux [14]:

Le service approprié aux applications élastiques est le service Best-Effort comme il l'a toujours été. Une autre appellation de ce service est « dès que c'est possible » (as-soon-as-possible), mais pour des raisons historiques le terme Best-Effort est gardé. Ce service ne propose aucune garantie sur le délai, la bande passante ou sur la perte de paquets.

[3] RFC (Request For Comment) : Ensemble de documents de référence décrivant les différents standards et normes en usage sur Internet. Les RFC, avant leur validation, sont des documents proposés et rédigés par des experts techniques. A ce stade, ils sont désignés par le terme draft (brouillon). Soumis à la communauté Internet, ils peuvent être commentés et amendés avant leur acceptation définitive. Toutes les RFC possèdent un numéro et un titre.

L'application intolérante exige qu'une limite supérieure du délai de chaque paquet soit assurée. Un service qui propose un délai maximum garanti de délivrance de paquets pour un flot convient parfaitement à cette classe d'application. Ce service est appelé Service Garanti (Garanteed Service).

L'application tolérante n'a pas besoin de fixer un délai offset supérieur au délai maximum de la délivrance de paquets. Le service proposé pour cette classe est appelé Service Prédictive (Predictive Service). Ce service ne garantit pas une limite supérieure fiable du délai mais il pourrait fournir une limite du délai maximum susceptible de changer selon le comportement d'autres flots.

2.2.3. L'architecture IntServ :

Le modèle IntServ tente de regrouper les avantages du réseau commutation de circuit et ceux du réseau à commutation de paquets sur un réseau à commutation de paquets. En spécifiant les caractéristiques d'un flot, un niveau de service est déterminé pour ce flot. Dans chaque routeur, un état par flot est maintenu et mis à jour périodiquement. Cela est réalisé en réservant explicitement des ressources sur chaque routeur du réseau tout au long du chemin du flot créant ainsi un circuit virtuel. Afin de mettre en place ces réservations, un protocole de signalisation est indispensable. Cela implique la distinction de deux éléments architecturaux : « Contrôle de trafic » et « Réservation de ressources »[14].

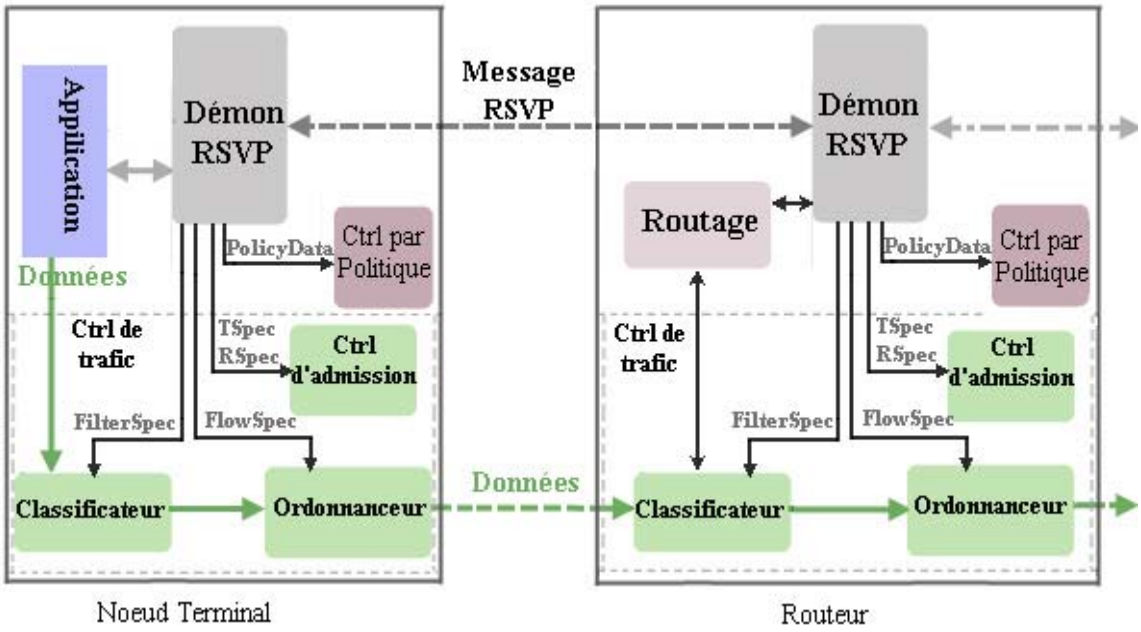


Figure II-1 : Architecture du routeur et du terminal IntServ.

2.2.3.1. Contrôle de trafic :

Le contrôle de trafic (Figure II-1) est l'ensemble de fonctions qui mettent en place le modèle de service décrit précédemment. Quand un paquet est reçu à l'entrée d'un routeur, le but de ce dernier est de le retransmettre sur l'une de ses sorties. Pour un paquet donné reçu, le routeur doit non seulement sélectionner sa route mais aussi décider de son départ, de suspendre son départ pendant un temps ou de le rejeter. Le routeur peut aussi décider de mettre en suspension l'émission d'un paquet même si le support de transmission est libre. Ces actions qui sont liées aux contraintes de l'état du routeur sont utilisées pour fournir les niveaux de service. Elles définissent le contrôle de trafic qui est implémenté par les trois composants suivants : le classificateur de paquet, le contrôleur d'admission et l'ordonnanceur de paquets (Figure II-1).

- Le rôle de l'élément classificateur de paquet (Packet Classifier) est de mettre en correspondance tous les paquets entrant avec les classes de QoS. Chaque paquet appartient à une classe et les paquets qui appartiennent à une même classe reçoivent le même traitement. Les flots sont tous traités par le classificateur afin que leurs paquets rejoignent leur classe.

- L'Ordonnanceur de paquets (Packet Scheduler) utilise un ensemble de files d'attente pour gérer les sorties des paquets vers le nœud suivant. Il sélectionne pour émission les paquets des différents flots en fonction des ressources qui leurs sont réservées. C'est l'élément qui accomplit la QoS promise pour un flot en prenant en considération l'ensemble du trafic.

- Le contrôle d'admission (Admission Control) qui est un module de décision est exécuté sur chaque nœud de la route du nouveau flot candidat pour vérifier si ce flot peut être supporté par le routeur. Il s'assure qu'un nouveau flot n'influence pas la QoS assurée pour les flots déjà en circulation dans le réseau. Il vérifie que suffisamment de ressources sont disponibles au moment de l'établissement de la réservation. Un nœud peut donc accepter ou rejeter un flot d'une application qui demande un niveau de QoS selon la disponibilité des ressources.

2.2.3.2. Classes de service :

Dans l'architecture IntServ, on peut définir plusieurs niveaux de service, mais trois niveaux ont été jugés suffisants pour couvrir les applications réparties qui s'exécutent sur Internet. Dans cette section nous présenterons les deux niveaux de service rajoutés au service Best-Effort qui sont le Service Garanti et le Service à Charge Contrôlée.

Le service **GS (Guaranteed Service)** [18] est destiné aux applications intolérantes qui exigent une garantie stricte du délai de délivrance des paquets à partir du moment de leur expédition de la source. Le niveau GS fourni par l'ensemble des éléments de service[4] tout au long d'une route, garantit un niveau de bande passante qui, quand elle est utilisée par un flot contrôlé, produit une borne du délai de délivrance sans perte de paquets tant que le flot est conforme. Le flot de données d'une application à qui une QoS est fournie est décrit par le modèle seau à jeton (Token Bucket) [18]. L'élément de service utilise les paramètres décrivant le flot donné pour savoir comment il va traiter les paquets de données de ce flot et calculer le délai maximum que va prendre une portion de données traversant le réseau. Le délai maximum et la bande passante restent stables tant que la route empruntée par le flot reste inchangée. Le GS ne contrôle pas le délai minimum ou la gigue ; il garantit seulement que les paquets ne dépassent pas un certain délai maximum établi au préalable.

Le service **CL (Controlled-Load)** [19] assure à une application un service très proche du service BE sous un réseau non ou légèrement chargé (avec l'hypothèse que le réseau fonctionne correctement). Cela se traduit par la garantie qu'un très grand pourcentage de paquets est délivré sans perte et avec un délai de délivrance ne dépassant pas trop le délai minimal pris par n'importe quel paquet convenablement délivré. Ce service est destiné aux applications tolérantes (adaptatives) qui sont très sensibles à la congestion dans le réseau mais qui peuvent contrôler leurs paramètres pour bien fonctionner. Les éléments de service considèrent que la bande passante est limitée par la capacité du support de transmission d'entrée du routeur. Ce service n'accepte pas des valeurs de paramètres de contrôle de QoS spécifiques tels que le délai de délivrance et la perte de paquets. Tout élément de service qui accepte une demande de CL s'assure que suffisamment de bande passante et de ressources sont disponibles pour prendre en considération le niveau de trafic d'un flot.

[4] L'élément de service peut être un routeur ou un sous-réseau qui fournit un niveau de QoS.

2.2.3.3. Protocole RSVP:

Le protocole RSVP (Resource ReSerVation Protocol) [15] [16] [17] est un protocole utilisé par les nœuds terminaux ou par les routeurs d'accès afin de demander une QoS pour un flot d'une application donnée (ou pour tous les flots du terminal) et par les routeurs intermédiaires afin d'installer et de maintenir les états pour fournir la QoS demandée. Il s'exécute au-dessus de IPv4 ou IPv6 comme un protocole de transport dans la pile des protocoles IP mais il ne transporte pas les données d'une application. Les données qu'il transporte lui sont « opaques » (il ne les interprète pas).

RSVP utilise sept messages (Annexe A :). Les plus importants pour l'établissement d'une session RSVP sont les messages *Path* et *Resv*. Les autres sont introduits soit pour signaler une erreur (*PathErr*, *ResvErr*) soit pour annoncer une confirmation de réservation (*ResvConf*) soit pour terminer explicitement la session RSVP (*PathTea*r, *ResvTear*).

Pour déclencher le processus de réservation de ressource dans le réseau, l'émetteur[5] envoie au(x) récepteur(s) un message *Path* pour chaque flot qu'il génère (**Figure II-2**). Le message *Path* décrit les caractéristiques du flot de données que l'émetteur souhaite générer. Il traverse les nœuds du réseau pour se créer un chemin jusqu'au(x) récepteur(s). Si une erreur s'est produite sur l'un des routeurs que le message *Path* a traversé, le routeur concerné arête la transmission du message *Path* et renvoie à l'émetteur le message *PathTear* pour effacer le chemin créé par le message *Path*. En réponse au message *Path*, chaque récepteur, fait une demande de réservation en envoyant le message *Resv* vers les émetteurs. Ce message va suivre exactement le (ou les) chemin inverse du message *Path*. Il décrit le type de réservation et les paramètres de la QoS demandée. Le récepteur peut demander la confirmation d'une demande de réservation. Il le fait en ajoutant son adresse IP dans le message *Resv*. Si la réservation est réussie sur tous les nœuds, l'émetteur recevra le message *Resv* et renvoie ensuite le message ResvConf contenant son adresse IP. Dans le cas où une erreur de réservation s'est produite, le nœud concerné renvoie au récepteur un message ResvTear. Le message *ResvTear* traverse le réseau de son point d'initiation vers tous les émetteurs en effaçant tous les états de réservation qui sont fait sur les nœuds.

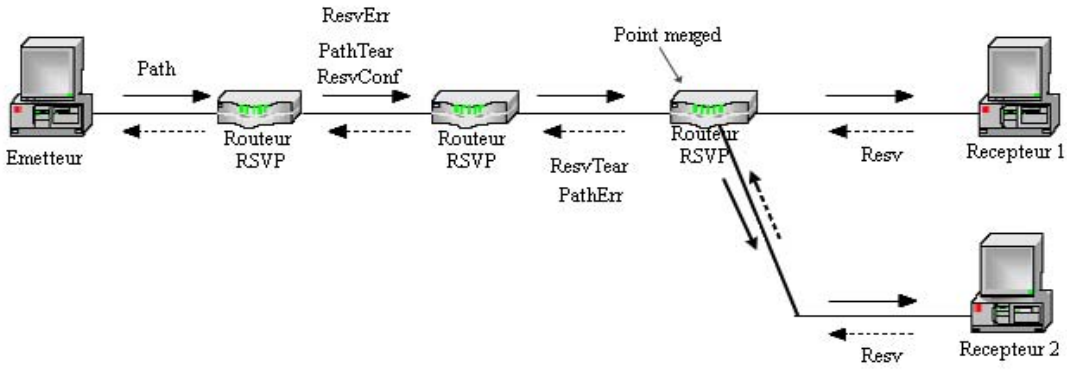


Figure II-2 : Signalisation RSVP

Bien que l'architecture IntServ regroupe des mécanismes robustes pour mettre au point la QoS, elle a cependant un inconvénient majeur qui retarde probablement voire empêche son déploiement [20]. La gestion des ressources « par flot » utilisée dans cette architecture implique le maintien dans chaque routeur de trois états pour chaque flot qui le traverse : Un

[5] émetteur c'est celui qui initie la session RSVP et le récepteur est celui qui initie la réservation en réponse à l'émetteur.

état pour le contrôle, un état pour la classification et un état pour l'ordonnancement ; ce qui réduit considérablement le nombre de flots gérés par le réseau. Ce problème est appelé problème de passage à l'échelle (Scalability). Cette architecture pourrait être avantageusement exploitée dans des réseaux à petite et moyenne taille. Cependant, IntServ est conçue afin d'être utilisée dans le réseau Internet qui peut être traversé par un nombre très important de flots. Nous présenterons par la suite une architecture qui évite le problème de passage à l'échelle mais en revanche ne fournit pas la même garantie de QoS que l'architecture IntServ.

2.3. L'architecture DiffServ :

Dans l'architecture DiffServ (Differentiated Service [22] [25]), l'élément de base qui fournit un niveau de service est le domaine DS (DiffServ Domain). Un domaine DS est un ensemble contigu de routeurs DiffServ. Ces routeurs sont de deux types (Figure II-3) : *les routeurs de bordure* qui délimitent le domaine DS et *les routeurs de cœur* qui se trouvent à l'intérieur du domaine DS.

Un flot de trafic d'une application ou d'un terminal qui entre dans le domaine DS est classifié puis conditionné par le routeur de bordure pour qu'il soit traité par les routeurs de cœur afin qu'il reçoive le service souscrit. Chaque routeur de cœur exécute un comportement particulier pour chaque niveau de service qui s'appelle Per-Hop Behavior (PHB) (section I.2.3.1). Les routeurs de bordure s'assurent que les paquets qui traversent le domaine DS soient adéquatement marqués pour recevoir le PHB approprié fourni par chacun des routeurs de cœur. Cette architecture assure le passage à l'échelle en repoussant toute la complexité du traitement vers les routeurs de bordure du DS qui maintiennent les états par flot. Les routeurs de cœur sauvegardent seulement les états par agrégation. Contrairement aux routeurs de bordure qui implémentent un classificateur qui se base sur une classification à plusieurs champs de l'en-tête IP, les routeurs de cœur, quant à eux, implémentent un classificateur simple qui se base seulement sur le contenu du DSCP (DS CodePoint) [21] qui est un seul champ de l'en-tête IP.

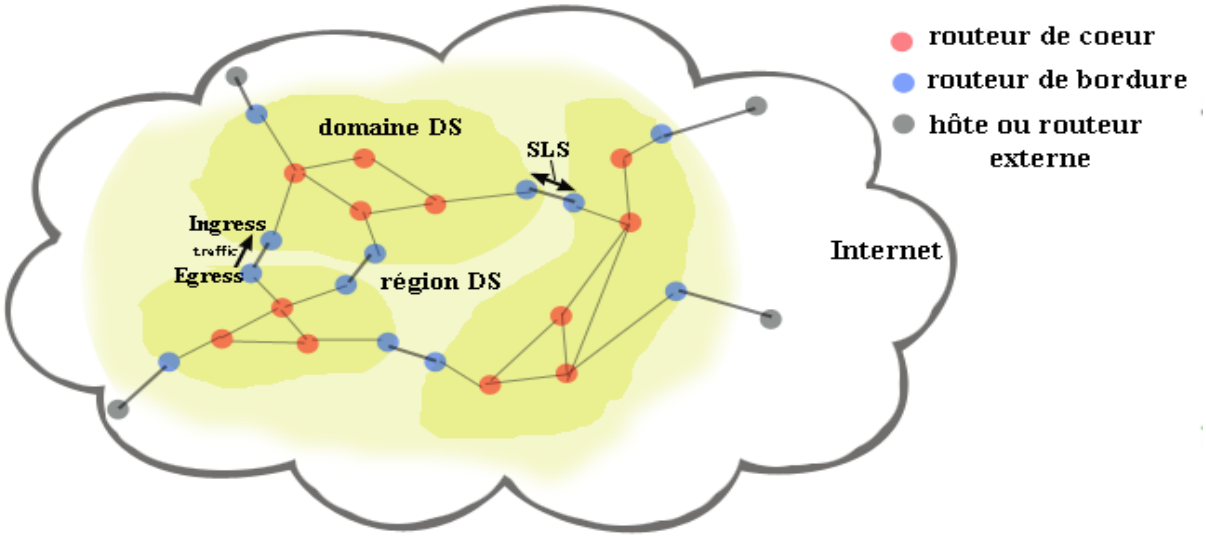


Figure II-3 : L'architecture DiffServ

Généralement un Domaine DS correspond à un ou à plusieurs réseaux sous la même administration (ISP :Internet Service Provider, FAI : Fournisseur d'Accès à Internet). Un ensemble contigu de domaines DS interconnectés par les routeurs de bordure forme une Région DS (Figure II-3). Les routeurs de bordure assurent aussi l'interconnexion des

domaines DS avec d'autres réseaux non-DiffServ et exécutent les fonctions du conditionnement de trafic comme spécifié dans le SLS. Le SLS (Spécification du Conditionnement du Trafic / Traffic Conditioning Specification) est un ensemble de paramètres et leurs valeurs qui, regroupés, spécifient l'ensemble de règles de classification et de conditionnement de trafic afin de définir le service offert au flot du trafic par le DS.

Un routeur DiffServ [27] peut être vu comme un ensemble de deux modules (Figure II-4) qui sont le Classificateur *(Classifier)* de paquets et le Conditionneur *(Traffic Conditioner)* . Ces modules peuvent être implémentés dans les routeurs de bordures tout comme dans les routeurs de cœurs mais avec des complexités différentes.

Le Classificateur est chargé de classifier les paquets dans des agrégations en se basant sur le contenu de l'en-tête IP. Tout le trafic qui rentre dans le routeur passe obligatoirement par ce module. A la sortie de ce module, le trafic sera réparti en plusieurs flots logiques. Les paquets appartenant à la même agrégation recevront le même traitement par le conditionneur de trafic et le mécanisme de PHB dans un routeur. Ce module est configuré de manière à ce qu'il satisfasse aux règles de classifications stipulées dans le TCS.

Le Conditionneur de Trafic traite différemment les agrégations de flot issues du classificateur. Quand les paquets sortent du conditionneur de trafic du routeur de bordure, le champ DS de chaque paquet est positionné à la valeur DSCP appropriée. Le conditionneur de trafic est constitué de plusieurs composants qui assurent chacun des fonctions particulières tout en respectant les règles spécifiées dans le SLS. Ces composants sont : Le *Mesureur* (Meter) qui mesure les propriétés temporelles de l'agrégation sélectionnée par le classificateur afin de déterminer si l'agrégation est conforme à son profil ou non ; Le *Marqueur* (Marker) qui regroupe les fonctions qui positionnent la valeur DSCP du champ DS des différents paquets qu'il reçoit ; Le *Reformateur* (Shaper) retarde les paquets d'un flot ou d'un ensemble de flots afin qu'ils soient conformes à leurs propriétés temporelles et enfin L'*Eliminateur* (Dropper) élimine les paquets appartenant à un flot de trafic parce que ces paquets rendent ce flot de trafic non conforme à son profil.

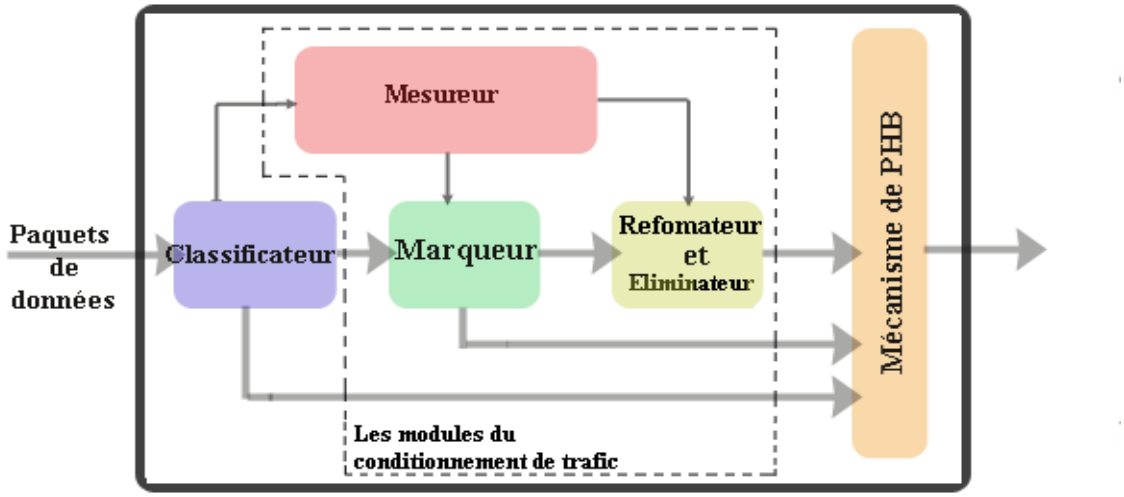


Figure II-4 : Routeur DiffServ.

2.3.1. Classes de services :

Dans l'architecture DiffServ une classe de service se traduit par une PHB (Per-Hop Behavior). Le PHB est le « comportement » observable appliqué par un routeur DS sur une agrégation de flots qui est définie par la valeur DSCP pour la transmission de paquets. Le

PHB dans un routeur est implémenté par le mécanisme de PHB en moyen de la gestion des files d'attente, de la gestion de buffer et du scheduler. En plus du PHB par défaut qui est le PHB Best-Effort, deux autres PHB ont été défini par l'IETF.

Le premier est le PHB Expedited Forwarding (PHB EF) [26] [24]. Ce PHB est spécifié afin de fournir une latence faible, une gigue faible et un taux faible de perte de paquets en assurant que l'agrégation de flot EF est servie par une certaine bande passante configurée. Le but de PHB EF est que les paquets marqués par le DSCP qui correspondent au PHB EF rencontrent des files d'attente très courtes ou vides avant leurs transmissions. De plus, si les files d'attente restent courtes par rapport au buffer disponible à l'agrégation EF la perte de paquets reste alors minime[24].

Le deuxième PHB est le PHB Assured Forwarding (PHB AF) qui est réellement un groupe de PHB. Le groupe PHB AF [23] définit quatre classes de AF dont chacune alloue une quantité de ressources (buffer et bande passante) dans le routeur DS. Il est un moyen pour le domaine DS de fournir différents niveaux d'assurance de transmission des paquets IP pour les clients du domaine DS. Ce niveau de service est destiné aux applications qui veulent que leurs paquets soient transmis avec une grande probabilité tant que leurs flots restent conformes à leurs profils. Les paquets qui excèdent le profil de trafic auquel ils appartiennent ne sont pas transmis avec le même degré de probabilité. Un routeur qui assure PHB AF ne réordonner pas les paquets d'un flot d'une application s'ils appartiennent à la même classe AF ; cela même s'ils ne sont pas conforme au profil. Une agrégation appartient à une classe AF et ne regroupe pas des paquets appartenant à deux ou à plus de deux classes AF.

Dans l'architecture DiffServ, en considérant le trafic global comme un ensemble réduit d'agrégations et en repoussant les traitements complexes aux extrémités du réseau qui est défini comme domaine DS, le but principal qui été le passage à l'échelle est bel et bien atteint. Aussi, le fait de considérer le réseau global qui fournirait une QoS de bout-en-bout comme un ensemble de domaines DS, permettra d'intégrer une architecture de QoS d'une manière incrémentale. L'autre avantage de DiffServ est qu'une API (Application Programming Interface) ou un changement logiciel des hôtes n'est pas indispensable pour que les applications qui s'exécutent sur Internet puissent avoir la QoS offerte par cette architecture. Par contre cette architecture, à moins de se contenter d'une sous utilisation du réseau, peut juste fournir une QoS approximative. De plus l'absence d'une signalisation empêche les mécanismes utilisés de fournir une QoS dynamique. Une autre faiblesse de DiffServ est que fournir la QoS aux trafics multicast reste un problème délicat surtout pour le conditionnement de ce genre de trafic.

2.4. IntServ sur DiffServ :

Afin de mettre à profit le passage à l'échelle de DiffServ et la flexibilité de IntServ pour garantir une QoS de bout-en-bout, le groupe de travail ISSLL de IETF propose une architecture qui combine DiffServ et IntServ [28]. Dans cette architecture, L'Internet est considéré comme une combinaison de routeurs de IntServ qui appliquent une classification MF et un contrôle de trafic par flot et des régions DiffServ qui appliquent une classification BA et un contrôle de trafic par agrégat. Les réseaux IntServ sont utilisés comme réseaux d'accès tandis que les régions DiffServ sont utilisées dans les réseaux de cœur (backbone) (Figure II-5). Du coté IntServ, les régions DiffServ sont vues comme des liaisons virtuelles ; pour DiffServ, les nœuds IntServ sont considérés comme des clients à qui le service est fourni en spécifiant un SLS. Les deux types de réseaux sont attachés avec les routeurs de bordure IntServ qui sont adjacents aux régions DiffServ et les routeurs de bordure DiffServ qui sont adjacents au réseau IntServ.

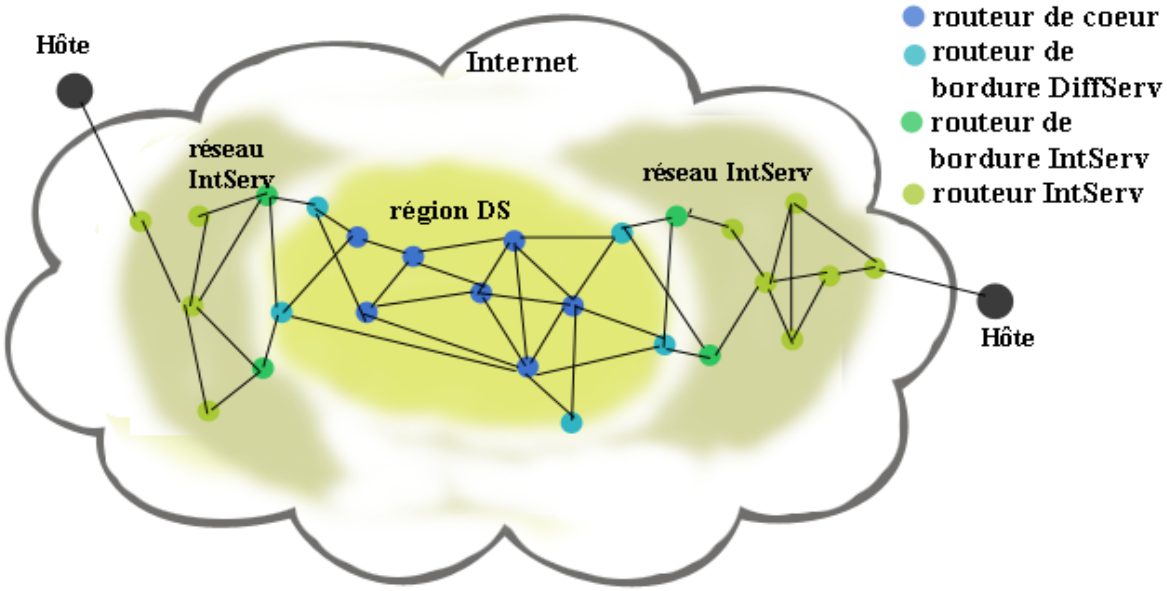


Figure II-5 : Architecture de IntServ sur DiffServ

Plusieurs modifications peuvent être apportées au routeur IntServ et au routeur DiffServ pour concevoir respectivement les routeurs de bordure de IntServ et de DiffServ afin de permettre l'interaction entre les deux type de réseaux. Les fonctions des routeurs de bordure IntServ et celles des routeurs de bordure DiffServ dépendent de la réalisation de l'architecture. Dans le cas où la région DiffServ ne prend pas en considération la signalisation RSVP, les routeurs de bordure IntServ peuvent agir comme un agent de contrôle d'admission pour la région DiffServ. Ils appliquent alors un contrôle d'admission basé sur les ressources disponible dans la région DiffServ et les routeurs de bordure DiffServ agissent comme de simples routeurs DiffServ et leur seule responsabilité est de conditionner le trafic en se basant sur le SLS négocié avec le client (IntServ). Dans le cas où la région DiffServ prend en considération RSVP, les routeurs de bordure IntServ appliqueront un contrôle d'admission selon la disponibilité de ressources locales. Dans ce cas, les routeurs de bordure DiffServ implémenteront un agent de contrôle d'admission pour la région DiffServ et participeront à la signalisation RSVP. Le terminal ou le premier routeur IntServ d'accès peut toujours marquer ces paquets IP avec le DSCP adéquat au niveau service DiffServ qu'il souhaite avant de former son flot pour le service IntServ selon la réservation de ressources faite par la signalisation RSVP. Dans l'autre cas, le marquage se fait par les routeurs de bordure IntServ ou par les routeurs de bordure DiffServ en suivant une correspondance entre DSCP qui implique un PHB dans la région DiffServ et un niveau de QoS de la région IntServ.

2.5. Le protocole COPS :

La gestion par politique est née du besoin des opérateurs d'automatiser les processus de configuration, qui autre fois se faisait "manuellement" par un administrateur, des nœuds et des équipements réseau. Cette automatisation permet de contrôler les flux d'informations qui circulent dans ces nœuds ainsi que de gérer plus facilement les équipements réseau. Les politiques (règles) fonctionnent sous la forme *« si condition alors action »*. La politique étant définie par une discussion entre l'utilisateur et l'opérateur. On utilise pour cela soit le langage naturel, soit des règles déjà établies par l'opérateur du réseau. Dans ce cas, l'utilisateur est restreint dans son choix aux règles définissant la politique souhaitée. Une fois traduite dans un langage compréhensible par le réseau, elle permet de déterminer le protocole que l'on souhaite utiliser pour gérer la QoS et son paramétrage.

Le protocole COPS (Common Open Policy Service [38]) est un protocole de gestion de réseau qui a été conçu par l'IETF. Prévu au début pour gérer les règles de QoS associées à RSVP, ce protocole a été repris pour s'appliquer à l'ensemble des technologies de QoS. Par la suite, ce modèle devient générique et applicable à d'autres domaines que celui de la QoS. L'un de ces domaines est, entre autres, le domaine de la sécurité. COPS est un protocole de gestion de réseau IP dans lequel un serveur transmet à des appareils clients (généralement des routeurs) les règles dont ils sont chargés de mettre en application afin d'assurer la politique définie. Dans cette partie nous allons présenter ce protocole et son architecture sur laquelle il s'applique.

2.5.1. Architecture générale de la gestion par politique :

Le protocole COPS est utilisé pour faire communiquer les informations politiques entre le point d'application de politique (Policy Enforcement Point : PEP) et le serveur de politique (Policy Decision Point : PDP) dans le contexte d'un type de client (application) particulier (Figure II-6). Une politique est définie comme un ensemble de règles capables de gérer et de contrôler l'accès aux ressources du réseau. Elle peut être considérée comme un ensemble de règles d'autorisation.

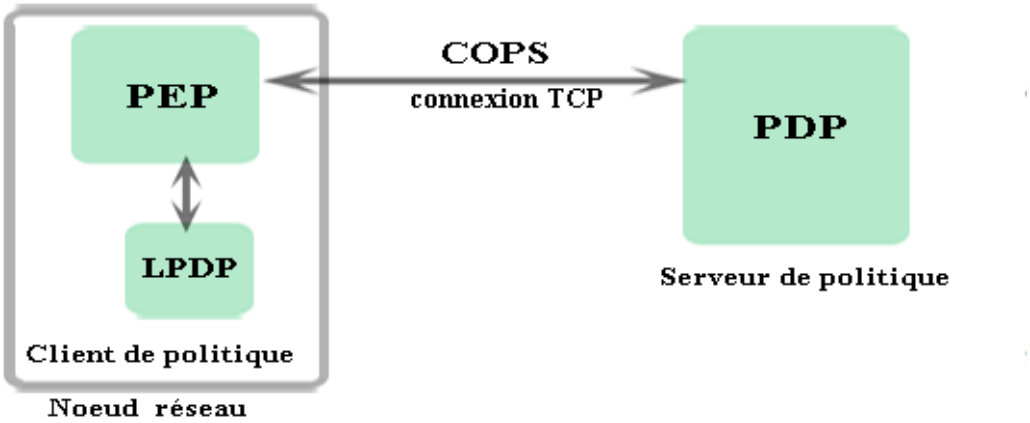


Figure II-6 : L'architecture par politique

Le PEP est une entité logique qui va appliquer les décisions politiques ; il est configurable à distance et situé dans les nœuds réseaux. Il joue le rôle de policier pour les demandes des utilisateurs. Le PDP est aussi une entité logique prenant des décisions politiques pour elle-même et pour d'autres éléments réseaux lui faisant la requête (Figure II-6). Par conséquent, il décide d'accepter ou non les demandes. Cette entité détermine les décisions et les configurations à appliquer aux ressources pour satisfaire les politiques réseaux. Elle joue le rôle de juge. Pour ce faire, le PDP consulte généralement une base de politiques. Il peut aussi consulter un serveur d'authentification afin d'authentifier les utilisateurs qui utiliseront les ressources. Le serveur de politique local (Local Policy Decision Point : LPDP) pourrait être utilisé par le PEP pour avoir des décisions locales dans l'absence temporaire du PDP.

Pour sa fiable communication, le protocole COPS utilise une connexion TCP qui est toujours initialisée par le PEP. Les résultats des échanges (requêtes/décisions) sont sauvegardés sous forme d'état qu'on appelle états de la requête. Le protocole permet de garder le lien entre les états de requêtes qui sont dans le PEP et ceux qui sont dans le PDP. Le PDP peut envoyer au PEP des décisions non sollicitées pour changer les états déjà approuvés et le PEP transmet au PDP le rapport d'exécution des décisions pour que le PDP puisse assurer une bonne gestion et une facturation cohérente. Le serveur PDP peut répondre à de

nouvelles requêtes différemment en vertu des informations d'états relatives à des requêtes ou à des décisions antérieures. Le PEP informe toujours le PDP quand ses états de requêtes sont changés. Il est également chargé d'effacer tout état qui n'est plus valide à cause d'un événement issu d'un utilisateur ou d'une décision du serveur.

Le protocole COPS est extensible. Il est conçu de manière à ce qu'il supporte des informations et des objets de clients différents sans avoir recours à un changement dans le protocole lui-même. Il peut transporter des objets privés qui contiennent les données nécessaires pour identifier les états des requêtes, établir le contexte pour une requête, identifier le type de requêtes, référencer les requêtes déjà installées, faire correspondre les décisions politiques, établir les rapports d'erreur, fournir l'intégrité des messages et transporter les informations des clients spécifiques. Le contexte de chaque requête correspond au type de l'événement qui l'a déclenchée. Pour la distinction entre les différents genres de client, le type de client est identifié dans chaque message. Les différents types de clients peuvent avoir différents types de données et peuvent exiger différents types de décision politique.

2.5.2. La communication COPS :

Le protocole COPS définit dix messages échangés entre le PEP et le PDP (Annexe B :) et qui sont : *Request* (*REQ*), *Decision* (*DEC), Report State* (*RPT*), *Delete Request State* (*DRQ*), *Synchronize State Req* (*SSQ*), *Client-Open* (*OPN*), *Client-Accept* (*CAT*), *Client-Close* (*CC*), *Keep-Alive* (*KA*), *Synchronize Complete* (*SSC*).

Après avoir établi une connexion TCP entre le PEP et le PDP et après les échanges de négociation de sécurité, le PEP envoie un ou plusieurs messages *OPN* au PDP ; un pour chaque type client supporté par le PEP. Ensuite, si le type de client n'est pas pris en charge par le PDP, celui-ci répond avec un message *CC* spécifiant que le type de client n'est pas supporté et suggère éventuellement l'adresse et le port d'un autre PDP ; sinon, le PDP répond par un message *CAT* pour chaque type de client demandé par le PEP et supporté aussi par le PDP. Le PDP spécifie dans le message *CAT* l'intervalle des messages *KA*. Après cela, le PEP et le PDP peuvent commencer à s'échanger tous les deux des messages COPS selon le modèle de COPS utilisé. Deux scénarios de communications sont possibles : l'un est le scénario d'externalisation et l'autre est le scénario d'approvisionnement.

Dans le modèle d'externalisation (outsourcing model [39]), le PEP doit prendre une décision qu'il « externalise » (qu'il fait soumettre) au PDP. Ce modèle est parfaitement adapté à des environnements réseaux utilisant des protocoles de signalisation comme RSVP dans lesquels les demandes de ressources RSVP seront traitées de manière centralisée. Cette implémentation particulière de COPS est appelée COPS-RSVP. D'autres protocoles de signalisation peuvent tirer partie de ce mode de gestion par politiques.

Dans le modèle d'approvisionnement (provisioning model [40]) le paramétrage de certains éléments réseau peut dépendre de la politique du réseau. Un PDP peut faire charger dans les équipements (les PEP) de nouvelles informations de configuration ou des mises à jour partielles en réponse à des événements administratifs. Il peut s'agir par exemple de changements de politique tarifaire d'un opérateur sur les équipements, d'un changement de SLS d'un utilisateur, d'une autorisation manuelle etc. Ces règles sont définies à chaque fois que cela est nécessaire. Ce modèle met en œuvre une structure de données dénommée PIB adaptée aux données politiques. Le modèle d'approvisionnement convient parfaitement au contrôle par politique des réseaux non fondés sur des protocoles de signalisation tel les réseaux de type DiffServ. Toutefois, il peut être utilisé pour d'autres types d'architecture.

Les deux scénarios utilisent les messages de requête et de décision pour faire installer les états de requête sur le PEP et sur le PDP. Ces états sont mis en synchronisation entre le PEP et le PDP en utilisant les messages *SSQ, SSC, REQ, DEC* et *DRQ* (Diagramme de séquence II-1). Dans tous les cas, le PEP peut informer le PDP sur le statut local d'un état installé en utilisant les messages de rapport *RPT* quand cela est approprié. Le message *RPT* peut être utilisé aussi pour déterminer des actions à entreprendre ou pour produire des mises à jour périodiques pour des fins de gestion ou de comptabilité.

Le message *KA* est utilisé pour vérifier la validation de la connexion entre le client et le serveur quand il n'y a aucun message COPS à envoyer entre le PEP et le PDP. En recevant le message *KA* de la part du PEP, le PDP répond avec le même type de message. Si l'un des cotés ne reçoit pas de message *KA* ou un autre message COPS de la part de l'autre coté, selon un intervalle prédéfini, la connexion devrait être considérée comme perdue.

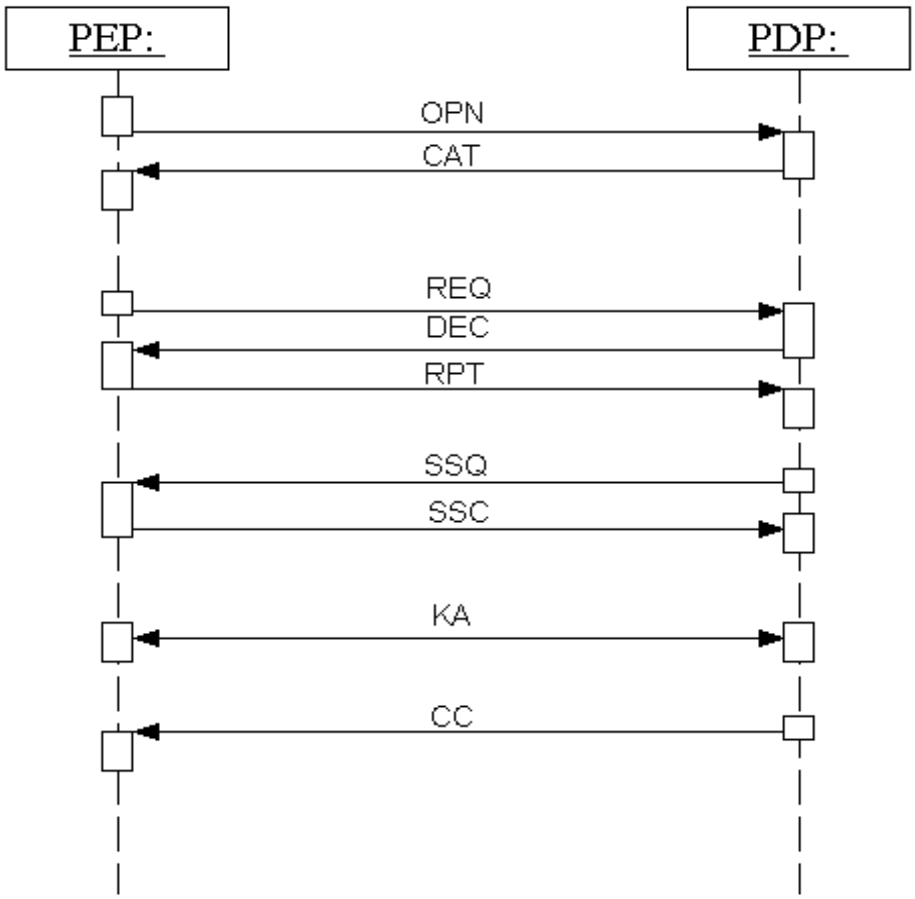


Diagramme de séquence II-1 : communication COPS.

Enfin, le message *CC* est utilisé pour une réponse négative au message *OPN* correspondant en informant l'autre coté que le type client spécifié n'est pas ou n'est plus supporté ou activé. Quand le PEP détecte la perte d'une connexion à cause de l'écoulement du temps **KA Timer** sans la réception d'aucun message, il doit envoyer explicitement un message *CC* pour chacun des types clients ouverts en spécifiant le code d'erreur de l'échec de la communication. Le PEP peut ensuite procéder à la terminaison de la connexion avec le PDP et peut tenter de se reconnecter avec le même PDP ou avec un PDP de secours. Quand le PDP se prépare à la fermeture, il doit aussi envoyer explicitement un message *CC* pour chaque type client de chaque PEP et peut spécifier un PDP alternatif.

3. Architectures et protocoles AAA :

Le terme AAA est dérivé des initiales des trois mots anglais « Authentication, Authorization et Accounting » qui ont les significations suivantes :

Authentication (Authentification): cela consiste à reconnaître qu'une entité (personne ou équipement) est bien celle qu'elle prétend être afin de l'autoriser ou non à accéder à des ressources ou à des services. Ceci est généralement réalisé en utilisant un secret partagé entre l'utilisateur et le serveur tel une combinaison Nom d'hôte / Mot de passe, une clé, reconnaissance biométrique etc.

Authorization (*Autorisation)* : cela a pour but de déterminer si l'entité authentifiée est autorisée ou non à accéder et/ou à utiliser les ressources ou les services demandés. En d'autres termes, l'opération d'autorisation va définir quels seront les droits et services que le client va recevoir une fois son authentification effectuée par le serveur AAA. La fonction d'authentification et d'autorisation sont effectuées par le même environnement AAA.

Accounting (Comptabilité) : cela a pour objet la collecte d'informations relatives à l'utilisation des services et des ressources réseaux. Ces informations sont utiles pour facturer convenablement les services et pour mieux gérer les ressources et estimer leur coût d'allocation. Dans cette opération, le serveur AAA va fournir des méthodes de collecte et d'envoi de données concernant l'utilisateur.

Une architecture AAA [29],[30],[31]et[32] est généralement une architecture Client-Serveur qui rend réalisable l'ensemble de ces fonctionnalités (Authentification, Autorisation, Comptabilité) à travers de multiples technologies de réseaux et de plates-formes. Les serveurs AAA permettent de gérer les utilisateurs et les clients AAA dans des domaines. Les clients AAA sont installés sur des routeurs ou sur des Serveurs d'Accès au Réseau (NAS : Network Access Server). Dans une architecture AAA (Figure II-7), on trouve les utilisateurs sur lesquels les fonctionnalités sont appliquées, les clients AAA qui déclenchent les demandes de services et le serveur AAA qui se charge de gérer les utilisateurs.

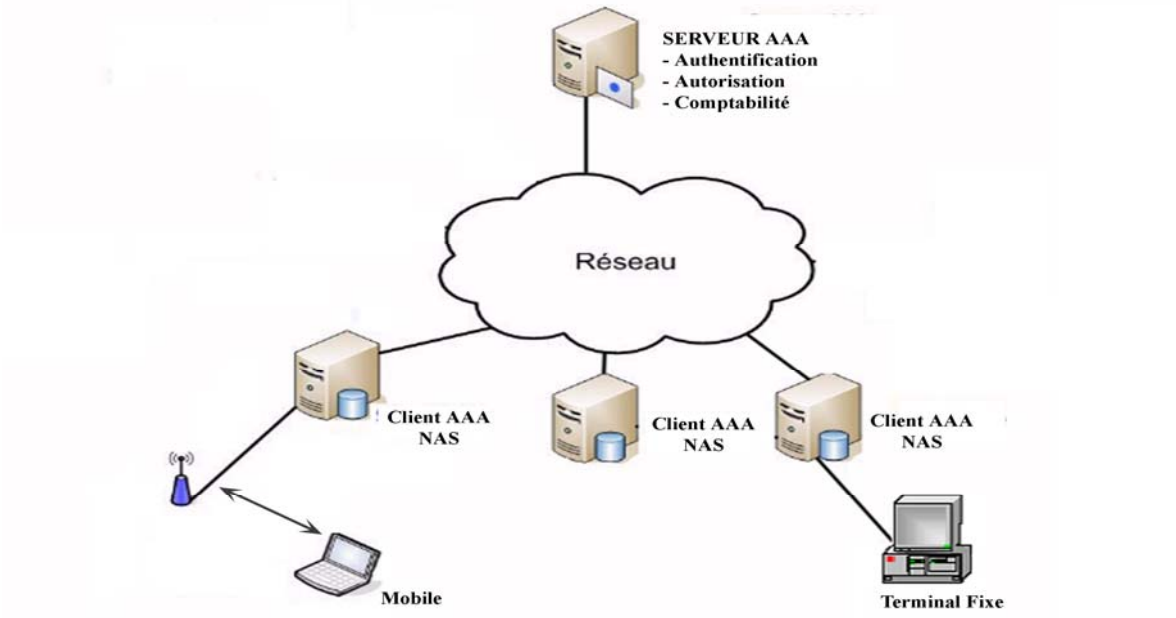


Figure II-7 : Architecture générale d'un réseau AAA

Le déroulement global de la procédure d'authentification d'un utilisateur est le suivant : l'utilisateur souhaite se connecter au réseau ; le NAS n'ayant pas les capacités d'authentifier l'utilisateur, collecte et transmet au serveur AAA les données qu'il a demandées auprès de l'utilisateur. Le serveur AAA va ensuite effectuer l'identification de l'utilisateur et renvoyer

le résultat (autorisation ou non) au client AAA avec, si nécessaire, les droits de l'utilisateur. Durant la transaction entre le client AAA et l'utilisateur, le client AAA peut demander des données au serveur concernant l'utilisateur.

Cette Architecture permet de contrôler l'accès aux réseaux et de connaître l'utilisation de leurs ressources. L'opérateur peut ainsi facturer le temps de connexion, la quantité d'informations téléchargées ou transmises, les services offerts etc. L'avantage principal de cette architecture est la possibilité de centraliser sur une base de données communes toutes les informations utiles pour l'accès et l'utilisation de ressources par les utilisateurs de nombreux clients AAA en évitant la synchronisation de plusieurs bases de données qui peuvent être logées dans ces routeurs d'accès.

Pour l'échange d'informations entre les nœuds AAA (utilisateur, client AAA et Serveur AAA), l'architecture AAA utilise des protocoles particuliers nommés protocoles AAA. Dans ce qui suit de cette section, nous présenterons les protocoles les plus usités et les plus connus sur le terrain de l'architecture AAA.

3.1. RADIUS:

Le protocole RADIUS (Remote Authentication Dial In User Service [33]) a été conçu pour permettre l'échange des informations d'authentification et d'autorisation entre les NAS et un serveur AAA centralisé. Il utilise le protocole UDP comme protocole de transport et permet la communication entre un client RADIUS (client AAA) et un serveur RADIUS (serveur AAA) qui traite les demandes du client. Le serveur RADIUS est responsable de la réception des demandes de connexions des utilisateurs se connectant au NAS, de l'authentification des utilisateurs et de la réponse avec toutes informations nécessaires à la configuration du client RADIUS pour fournir les services souscrits. Un serveur RADIUS peut agir comment un client Proxy (mandataire) qui reçoit des demandes de la part d'un client RADIUS pour les transmettre ensuite à un autre serveur RADIUS.

3.1.1. La communication RADIUS :

RADIUS comporte six types de message qui sont les suivants : *Access-Request* (requête d'accès), *Access-Accept* (accès accepté), *Access-Reject* (accès rejeté), *Access-Challenge* (envoie de challenge), *Accounting-Request* (demande de comptabilité) et *Accounting-Response* (réponse de la comptabilité). Chaque message transporte un ou plusieurs des 63 attributs définis dans la spécification de RADIUS. Les attributs sont utilisés pour transporter des informations d'authentification, d'autorisation, de comptabilité et de configuration dans certains cas.

Quand un client AAA est configuré pour utiliser RADIUS, tout utilisateur qui veut se connecter à ce client doit lui présenter des informations d'authentifications. Cela pourrait être une simple saisie par l'utilisateur de son nom d'utilisateur et de son mot de passe après avoir le prompt du login. L'utilisateur pourrait être connecté au client RADIUS avec un protocole tel le PPP (Point-to-Point Protocol) ou avec le protocole IEEE 802.1X (Sec. I.3.3). Une fois ces informations obtenues, le client peut choisir d'entamer la procédure AAA en utilisant RADIUS. Pour ce faire, le client génère un message *Access-Request* contenant des attributs qui comprennent le nom de l'utilisateur. Il transmet ensuite ce message au serveur RADIUS à travers le réseau (Diagramme de séquence II-2). Quand le serveur reçoit la requête, il consulte sa base de données d'utilisateurs afin de trouver l'utilisateur qui correspond à la requête. L'entrée de l'utilisateur dans la base de données contient une liste des exigences qui doivent être satisfaites pour que l'utilisateur soit permis à se connecter. Cela inclut toujours la vérification du mot de passe, mais aussi parfois la spécification du client ou le port auquel l'utilisateur est autorisé à se connecter. Si au moins une condition n'est pas satisfaite, le

serveur envoie un message *Access-Reject* indiquant que l'utilisateur n'est pas valide et un texte à afficher à l'utilisateur pourrait être inclus dans ce message.

Si toutes les conditions sont réunies et que le serveur RADIUS désire plus de vérification auxquelles l'utilisateur devrait répondre, le serveur transmet au client un message *Access-Challenge*. L'utilisateur recevra dans ce message un numéro généré au hasard (le challenge) qui ne sera probablement jamais répété. L'utilisateur cryptera correctement avec une fonction de hachage le mot de passe en utilisant son mot de passe et le challenge et retransmet ensuite le résultat au client RADIUS. Ce dernier construit un nouveau message *Access-Request* qui retransmet les nouvelles informations (résultat). Le serveur RADIUS répond à ce nouveau message *Access-Request* soit avec un *Access-Accept* soit avec un *Access-Reject* soit avec un autre *Access-Challenge*. Si toutes les conditions sont réunies, que cela soit après un *Access-Challenge* ou après une acceptation sans *Access-Challenge*, le serveur retransmet au client une liste de valeur de configuration pour l'utilisateur dans un *Access-Accept*. Ces valeurs incluent toutes les valeurs nécessaires pour délivrer les services désirés.

Une fois l'opération d'authentification et d'autorisation accomplie, le client qui commencera à délivrer le service va générer un message *Accounting-Request* [34] décrivant le type de service à délivrer et ce message sera ensuite envoyé au serveur pour signaler le début de la délivrance du type de service décrit. A la réception de ce message, le serveur RADIUS renvoie *Accounting-Response* attestant la réception du message *Accounting-Request (Start)*. A la fin de la délivrance du service, le client génère un message *Accounting-Request* décrivant le service délivré et ce message sera envoyé au serveur pour signaler la fin de la délivrance du service. Ce message peut optionnellement contenir des attributs qui transportent des statistiques tels le temps écoulé, les octets entrés et sortis ou les paquets entrés et sortis. A la réception de ce message, le serveur renvoie un message *Accounting-Response* attestant la réception du message *Accounting-Request (Stop)* (Diagramme de séquence II-2).

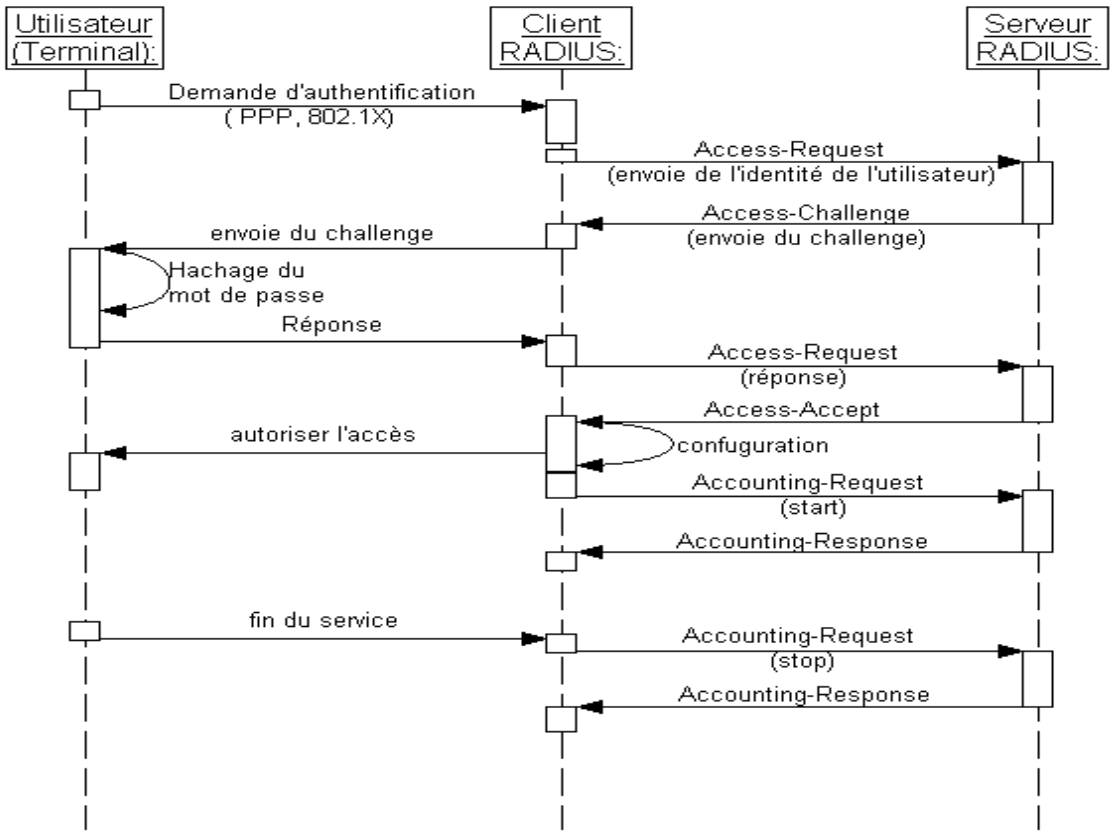


Diagramme de séquence II-2 : authentification RADIUS.

3.2. Diameter :

Le protocole Diameter de base [35] est conçu pour servir de support à une architecture AAA. Il fournit des mécanismes de base pour établir un transport fiable, délivrer les messages AAA et gérer les erreurs. Diameter peut être le successeur du protocole RADIUS auquel il a repris les principales fonctions et en a rajouté de nouvelles pour s'adapter aux nouvelles technologies et pour particulièrement prendre en considération les applications de la mobilité et inter-domaine. En effet, RADIUS était conçu à l'origine uniquement pour des protocoles tels que SLIP ou PPP. Il n'était pas vraiment extensible et ne pouvait donc offrir des services d'authentification pour les nouvelles technologies sans fils, cellulaire ou autres. Diameter se voulant évolutif, il est possible de redéfinir les messages du protocole, ses champs, ses applications, les procédures d'authentification pour s'adapter à de futures utilisations. Diameter peut fournir à une application deux types de service. Le premier type est l'authentification, l'autorisation et optionnellement la comptabilité ; le deuxième utilise seulement la comptabilité.

Afin de fournir un transport de données garanti et pour ne pas perdre surtout des informations de comptabilité, Diameter s'exécute sur un protocole de transport fiable. Pendant que RADIUS utilise le protocole de transport UDP pour communiquer, Diameter utilise des protocoles fiables qui sont : TCP et SCTP[6].

La communication du protocole Diameter se fait en utilisant deux types de message (Annexe C :): les messages de requête (Request) et les messages de réponse (Answer). Une requête peut être initiée soit par le client Diameter soit par le serveur Diameter. Dans ce sens, Diameter est un protocole Peer-to-Peer contrairement au protocole RADIUS qui est un protocole Client-Serveur où seul le client peut initier une requête. Cela facilite l'implémentation du traitement des déconnexions imprévues et des demandes de ré-authentification et de ré-autorisation par le serveur.

Le protocole Diameter est constitué d'un protocole Diameter de base et d'applications Diameter. Le protocole Diameter de base peut être utilisé par lui-même pour les applications de comptabilité. Il concerne lui-même avec la capacité de négocier comment les messages sont envoyés et les règles à appliquer pour tous les échanges de messages entre les entités Diameter. Les messages Diameter servent aussi bien à la signalisation du protocole qu'à délivrance de données propres aux applications clientes. Pour l'utilisation d'authentification et d'autorisation, il est étendu à des applications Diameter particulières. L'une de ces applications déjà spécifiées dans les RFC est l'application Mobile IPv4 Diameter [37] (sect. II.3.1) qui permet la gestion de la mobilité avec l'infrastructure AAA en utilisant Mobile IPv4, l'autre est l'application NAS Diameter [36] qui permet l'accès au réseau via PPP/SLIP et une autre est EAP Diameter [56] (sect. I.3.3.3) pour l'authentification EAP. Le client Diameter et le serveur Diameter sont compatibles avec les applications NAS et Mobile IPv4. Dans le cas où le Serveur (resp. Client) Diameter ne supporterait pas les applications NAS et Mobile IPv4, ce serveur ne porte plus le nom Serveur (resp. client) Diameter mais serveur (resp. Client) X Diameter où X est le nom de l'application supportée.

En plus du client et du serveur, le protocole Diameter introduit des agents qui sont les agents de relais, de Proxy, de re-direction et de translation. Un élément de l'architecture est appelé entité ou nœud Diameter et un ensemble de ces nœuds connectés entre eux constitue un réseau Diameter. L'introduction des agents a plusieurs intérêts. Ces agents peuvent être

[6] **SCTP** (Stream Control Transmission Protocol) : Il fournit des services similaires à TCP, assurant la fiabilité, la remise en ordre des séquences, et le contrôle de congestion et Il fonctionne sur le protocole IP. Une avancée majeure de SCTP est la possibilité de communications multi-cibles, où une des extrémités (ou les) de la connexion est constituée de plusieurs adresses IP. À l'origine, SCTP était destiné au transport de protocoles téléphoniques sur le réseau IP.

utilisés pour multiplexer plusieurs requêtes issues des différents équipements du client Diameter distribués et considérés comme un seul groupe, pour ajouter aux messages de requête et de réponse des données utiles, pour servir à la décentralisation du système afin de faciliter la configuration et la maintenance et pour répartir la charge.

3.2.1. Fonctionnement du protocole Diameter :

Le protocole Diameter de base définit comment les entités Diameter peuvent négocier leurs capacités, comment les massages sont envoyés et comment les sessions Diameter peuvent être annulées ou terminées. Pour transporter les données, les messages Diameter utilisent les AVP *Attribute Value Pairs)* qui peuvent être des fonctions d'authentification, d'autorisation, de comptabilité, de sécurité ou de routage (Annexe C :).

Une communication entre deux peer Diameter commence avec l'envoi d'un message *CER (Capabilities-Exchange-Request)* d'un peer vers l'autre. Quand deux peer établissent une connexion de transport, ils commencent à s'échanger les messages *CEA (Capabilities-Exchange-Answer)* et *CER* (Diagramme de séquence II-3). Pendant cette phase qu'on peut appeler phase des échanges de capacité, ces messages permettent aux entités Diameter de découvrir (informer) l'identité du peer et ses capacités locales telles que la version du protocole supporté, les applications Diameter supportées, le mécanisme de sécurité utilisé etc. Le récepteur du message *CER* qui n'a pas d'application Diameter en commun avec l'émetteur de ce message, doit renvoyer un message *CEA* avec un AVP qui spécifie l'erreur et déconnecte ensuite la connexion transport.

Etant donné la nature du protocole Diameter, il est recommandé de détecter l'échec du transport dès que possible. Les messages *DWR (Device-Watchdog-Request)* et *DWA (Device-Watchdog-Answer)* sont utilisés pour arriver à cette fin. Quand un peer est devenu injoignable à cause de la fermeture de la connexion transport ou parce qu'il n'a pas répondu après trois messages *DWR* avec un message *DWA*, il doit être remplacé par un autre peer.

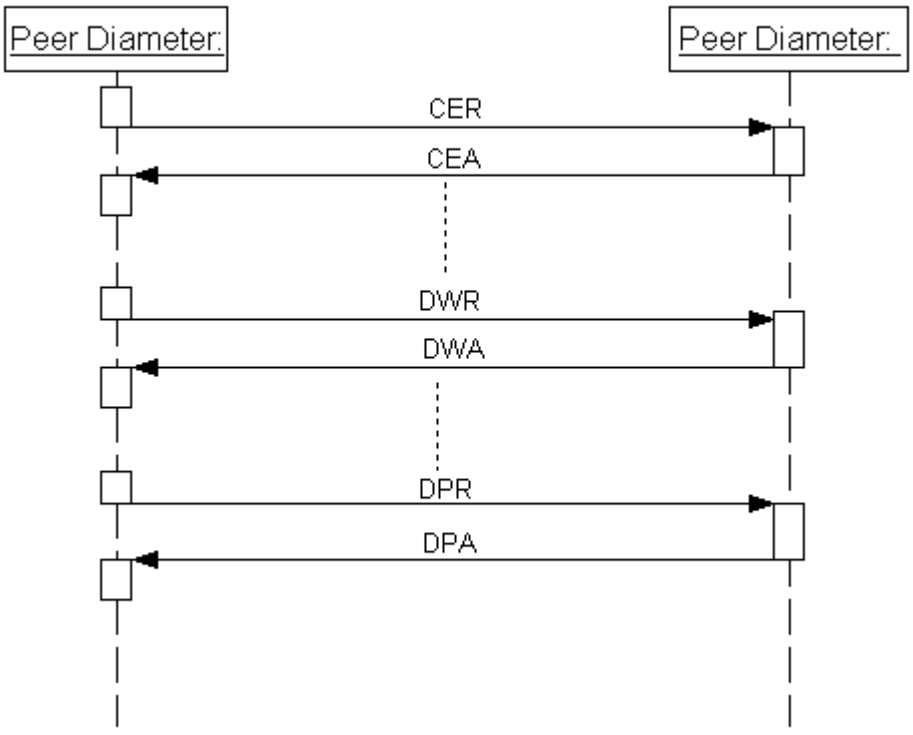


Diagramme de séquence II-3 : communication de deux Peer Diameter

Un serveur Diameter peut initialiser une demande de re-authentification et/ou de re-autorisation pour une session particulière en utilisant le message *RAR (Re-Auth-Request)* comme dans le cas où le serveur voudrait avoir une confirmation que l'utilisateur est toujours entrain d'utiliser le service. Un client qui reçoit un message *RAR* correspondant à une session active initialise une re-authentification et/ou re-autorisation si l'application supporte ce mécanisme et renvoie un message *RAA* (*Re-Auth-Answer)* au serveur.

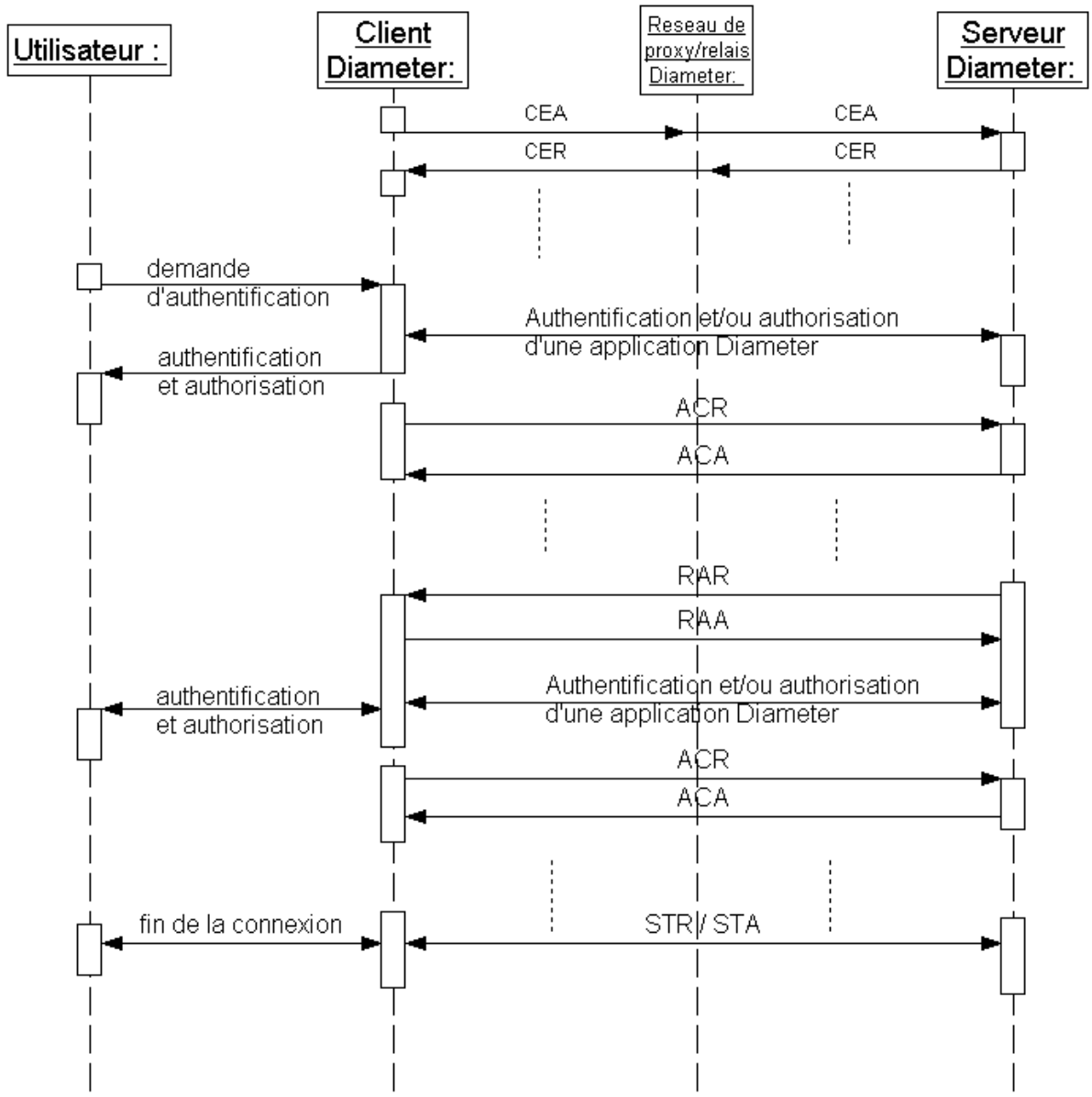


Diagramme de séquence II-4 : communication client-serveur Diameter.

L'état de session maintenu dans une entité Diameter doit être effacé à la réception d'un message *STR* (*Session-Termination-Request*) ou d'un message *STA (Session-Termination-Answer)*. Quand une session utilisateur qui est sous l'autorisation Diameter se termine, le client Diameter qui fournit le service va envoyer un message *STR* vers le serveur Diameter qui a fourni l'autorisation pour l'informer que la session n'est plus active. L'origine de fin de la session pourrait être la déconnexion volontaire de l'utilisateur, l'expiration de la session, une action administrative, une action prise après le message *ASR* ou l'arrêt de l'équipement d'accès etc (Diagramme de séquence II-4). Le client Diameter pourrait aussi émettre un message *STR* pour une session qui était autorisée mais qui ne serait jamais démarrée comme

dans le cas où les ressources seraient soudainement épuisées. Le client ou le serveur Diameter répondra toujours avec un message *STA* à la réception du message *STR*. Le serveur Diameter peut demander au client Diameter d'annuler (stopper) le service fournit à l'utilisateur en lui envoyant le message *ASR (Abort-Session-Request)*. Cela pourrait être le cas quand le crédit est terminé. A ce moment, le client répond au serveur avec un message *ASA (Abort-Session-Request)*. Quand un client arrête une session suite au traitement de la requête *ASR*, il va renvoyer au serveur qui a fait l'autorisation un message *STR*.

Comme il a été déjà précisé, Diameter assure aussi la comptabilité. Le protocole comptabilité (Accounting) Diameter [35] est basé sur un modèle orienté serveur avec possibilité de délivrance d'informations de comptabilité en temps-réel. Ce modèle signifie que l'équipement qui génère les données de comptabilité obtient du serveur qui a autorisé la session ou du serveur de comptabilité les directives de la manière et du moment de l'envoi des données de comptabilités. Le client Diameter qui reçoit un message de succès d'authentification et/ou d'autorisation de la part du serveur Diameter collecte des informations de comptabilité pour la session. Le message *ACR (Accounting-Request)* est utilisé pour transmettre ces informations au serveur Diameter qui, à son tour, répond avec un message *ACA (Accounting-Answer)* pour confirmer la réception (Diagramme de séquence II-4).

3.3. Le protocole EAP :

Le protocole EAP (Extensible Authentication Protocol) est une norme IETF décrite dans [42] et qui définit une infrastructure permettant aux clients d'accès réseau et aux serveurs d'authentification d'héberger des modules pour les méthodes et technologies d'authentification actuelles et futures. Les systèmes utilisent EAP pour authentifier l'accès réseau pour les connexions PPP (Point-to-Point Protocol,[44]) et pour l'accès réseau basé sur IEEE 802.1X [48] aux commutateurs Ethernet (IEEE 802, [50]) et aux points d'accès sans fil (IEEE 802.11, [49]). Nous décrirons dans cette partie l'EAP et son utilisation dans Diameter. Par la suite, le protocole IEEE 802.1X qui se base sur l'EAP sera décrit.

EAP a été créé à l'origine comme extension du protocole PPP afin de permettre le développement de méthodes arbitraires d'authentification de l'accès réseau. Avec les protocoles d'authentification PPP tels que CHAP (Challenge Handshake Authentication Protocol, [45]), MS-CHAP (Microsoft Challenge Handshake Authentication Protocol, [46]) et MS-CHAP version 2 (MS-CHAP-v2, [47]), un mécanisme d'authentification spécifique est choisi au cours de la phase d'établissement de liaison. Ensuite, au cours de la phase d'authentification, le protocole d'authentification déjà négocié permet l'échange d'informations sur les informations d'identification du client qui se connecte (Annexe D :).

Avec EAP, le mécanisme d'authentification spécifique n'est pas choisi au cours de la phase d'établissement de liaison de la connexion PPP ; les communicants PPP négocient afin d'effectuer EAP au cours de la phase d'authentification de la connexion. Lorsque la phase d'authentification est atteinte, les communicants négocient l'utilisation d'un mécanisme d'authentification EAP spécifique, appelé méthode d'authentification EAP. Une fois l'accord négocié sur la méthode EAP, le protocole EAP permet l'échange ouvert de messages entre le système à authentifier et le serveur d'authentification. La conversation est constituée de demandes et de réponses concernant les informations d'authentification.

L'avantage qui a permis à l'EAP d'avoir le succès est sa souplesse à supporter de multiples méthodes d'authentifications : mot de passe, carte à puce, certificat électronique etc. Cela l'a promu dans les applications de tous les contextes des réseaux y compris les réseaux sans fil (WiFi : IEEE802.11) où il a prouvé son efficacité grâce au protocole IEEE802.1x qui sera présenté par la suite.

3.3.1. Une architecture pour le EAP :

D'un point de vue architecturale (Figure II-8), une infrastructure EAP est constituée des trois éléments suivants :

Le Système à authentifier, suppliant ou le peer [7] est le système qui est au bout de la liaison physique ou logique et qui tente d'accéder au réseau (le terminal). Pour cela il répond à l'authentification afin d'avoir un service réseau ou une connexion ;

L'authentificateur est le système qui initie une authentification EAP. Généralement, il est un point d'accès (AP) ou serveur d'accès réseau (NAS) qui nécessite une authentification EAP avant d'accorder l'accès à un réseau ;

Le serveur d'authentification ou Serveur AAA [8] est le système serveur qui négocie l'utilisation d'une méthode EAP spécifique avec le suppliant. Il valide les informations d'identification afin de vérifier l'identité de l'utilisateur. En général, le serveur d'authentification est un serveur RADIUS ou un serveur Diameter.

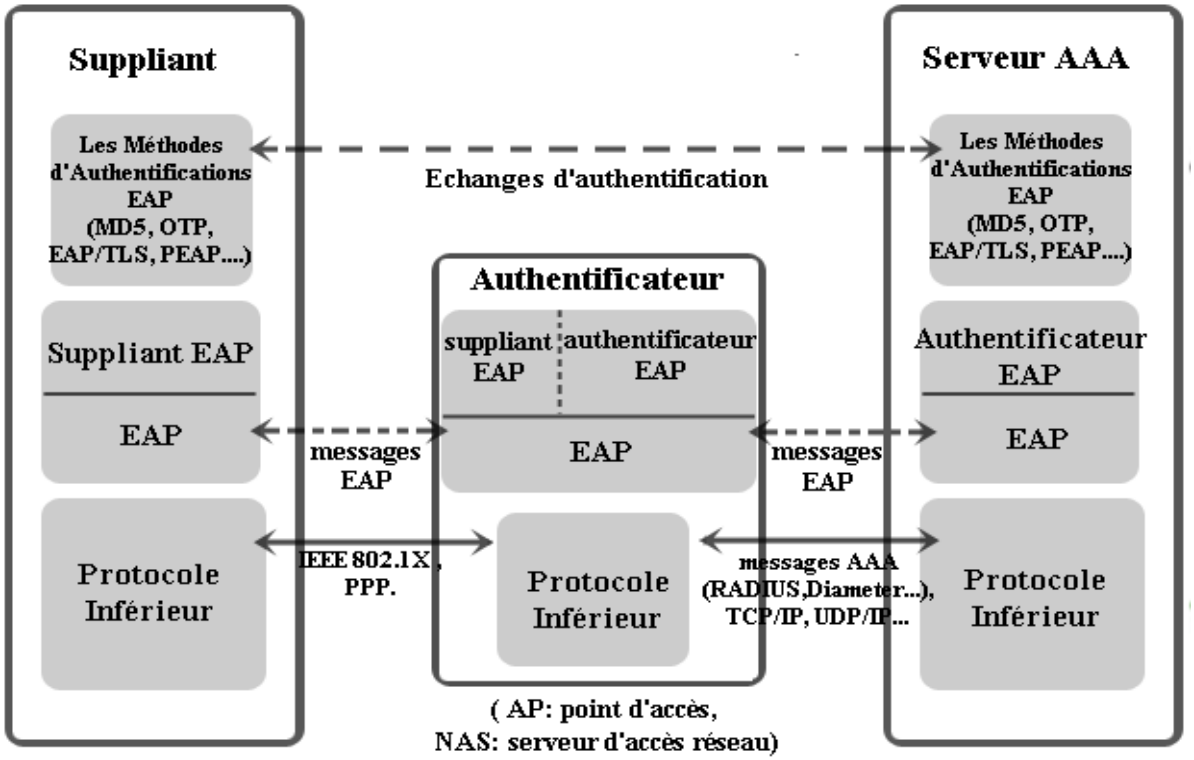


Figure II-8 : Une infrastructure générale pour l'EAP.

Pour son bon fonctionnement, l'organisation du protocole est sous la forme de couches communicantes. On peut différencier les composants suivants :

Le protocole inférieur à EAP : ce protocole est responsable de l'encapsulation des trames EAP afin de les transporter entre le suppliant et l'authentificateur ou entre l'authentificateur et le serveur AAA. Il peut être PPP, IEEE 802 de LAN, IEEE 802.11 (WiFi) de WLAN, UDP, TCP, protocole RADIUS etc. ;

Le EAP: Il reçoit et transmet les paquets EAP par l'intermédiaire du protocole inférieur. Cette couche implémente la détection de la duplication et la retransmission des messages. Le protocole délivre (ou reçoit) au suppliant EAP et à l'authentificateur EAP les messages EAP.

[7] Nous utiliserons souvent le terme suppliant.

[8] Bien que le serveur d'authentification et le serveur AAA ne soient pas vraiment les mêmes, dans cette section ces deux noms sont utilisés d'une manière interchangeable.

La couche EAP fait le démultiplexage de paquets EAP rentrant selon le champ Code de message.

Le suppliant EAP et authentificateur EAP : une implémentation EAP sur une machine pourrait supporter soit le suppliant EAP soit l'authentificateur EAP mais il est possible qu'une implémentation implémente les deux composants à la fois pour une authentification mutuelle. Le suppliant EAP et l'authentificateur EAP font le démultiplexage des paquets EAP et ils les délivrent seulement à la méthode correspondante ;

Les méthodes d'authentification EAP : elles implémentent les algorithmiques d'authentifications et transmettent au (ou reçoivent du) suppliant EAP et à l'authentificateur EAP les messages EAP. L'implémentation d'une méthode sur une machine pourrait être conçue de telle sorte qu'elle puisse réceptionner des paquets de la part du suppliant, de l'authentificateur ou des deux.

Dans l'architecture présentée dans la Figure II-8, l'authentificateur réagit comme un intermédiaire (pass-through) entre le serveur AAA et le suppliant. L'authentificateur n'a pas besoin de comprendre la communication entre le suppliant et le serveur d'authentification. Toutefois il doit interpréter les messages de résultat du succès ou de l'échec de l'authentification ; ce qui l'aidera à décider de laisser ou d'empêcher l'utilisateur d'accéder au réseau. S'il laisse l'utilisateur accéder au réseau, l'ensemble du trafic de l'utilisateur vers le réseau passera par cet authentificateur. Un succès d'authentification est donc un échange de messages d'EAP ayant comme résultat la permission au suppliant pour l'accès au réseau par l'authentificateur et l'acceptation de cet accès par le suppliant. Une décision typique de l'authentificateur implique les aspects de l'authentification et de l'autorisation assurés par le serveur AAA.

3.3.2. Fonctionnement de EAP :

Dès qu'un système se présente pour accéder au réseau, l'authentificateur envoie une *Requête-Identité* au suppliant. Cette requête n'est pas indispensable et pourrait être une phase ignorée comme dans le cas où l'identité ne serait pas exigée parce qu'elle est déterminée, par exemple, grâce au port auquel le suppliant est connecté, à son adresse MAC ou à son numéro de téléphone. A la réception de la *Requête-Identité*, le suppliant répond à l'authentificateur avec un message *Réponse-Identité* pour lui envoyer son identité. L'authentificateur retransmet la *Réponse* au serveur AAA. Ce dernier envoie une *Requête* d'un type de méthode EAP (EAP/MD5-Challeng, EAP/OTP…) qu'il voudrait utiliser pour inviter cet utilisateur à s'authentifier. Cette *Requête* est reçue par l'authentificateur qui la retransmet au suppliant (Diagramme de séquence II-5). Généralement, elle contient un défi ou un certificat du serveur. Avec ses informations secrètes et les informations qu'il a reçues, le suppliant calcule les données nécessaires pour s'authentifier. Il met ces données dans une *Réponse* du même type de méthode qui est demandé dans la *Requête* et qui est supporté par ce suppliant. Il envoie cette *Réponse* à l'authentificateur qui la retransmet au serveur d'authentification. Ce dernier traite les données reçues dans cette *Réponse* toujours selon la même méthode. Tant que c'est indispensable, les *Requête*s et les *Réponses* continuent à s'échanger. Les séquences *Requête* suivie de *Réponse* continueront jusqu'à que le serveur envoie un *Succès* ou un *Echec*. Toutefois, dès que le suppliant a commencé à répondre à une méthode d'authentification donnée, il ne peut plus revenir en arrière et choisir une autre méthode ; il doit aller jusqu'au succès ou à l'échec de l'authentification. Ainsi, une seule méthode d'authentification peut être utilisée au sein d'une même session d'authentification identifiée par la valeur du champ identifiant. De plus, hormis la *Requête* initiale, une *Requête* n'est jamais émise avant la réception de la *Réponse* valide de la *Requête* précédente.

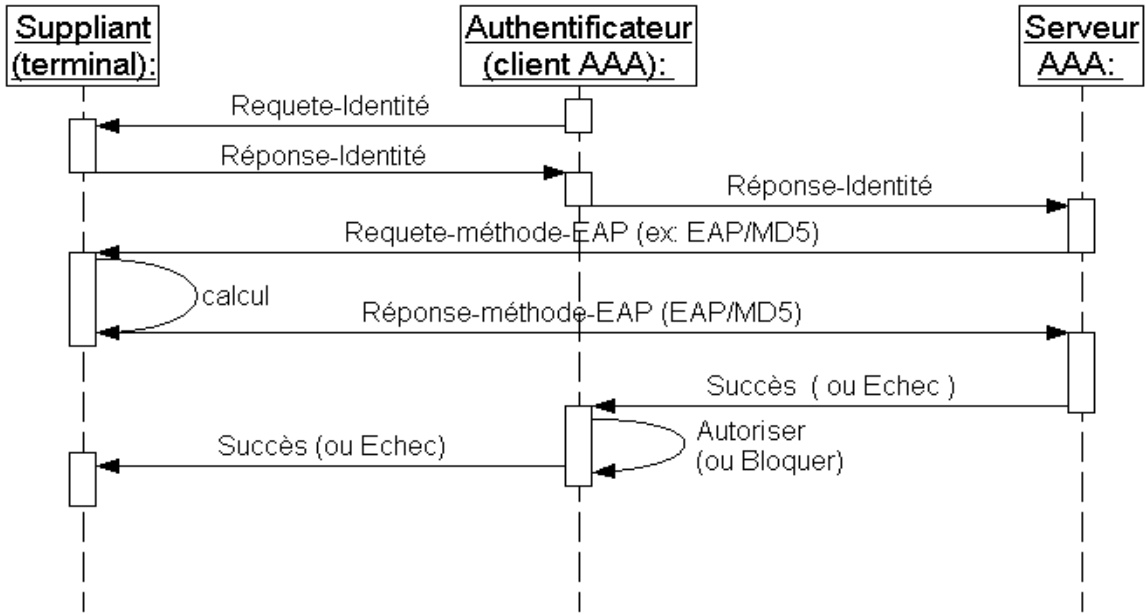


Diagramme de séquence II-5 : communication EAP.

Après s'être assuré que l'utilisateur s'est bien authentifié, le serveur d'authentification envoie le message *Succès* à l'authentificateur qui, avant de le retransmettre à son tour au suppliant, autorise l'accès à cet utilisateur. S'il ne peut pas authentifier le suppliant à cause d'une *Réponse* inacceptable pour une ou plusieurs *Requêtes*, le serveur transmet dans ce cas un message *Echec* qui transite de la même façon que le message *Succès* mais, cette fois, l'authentificateur bloque l'accès au système à authentifier.

S'il ne supporte pas la méthode proposée par le serveur, le suppliant va négocier une autre méthode. Pour cela, il prépare une *Réponse-Nak* pour refuser la méthode demandée et en proposer d'autres. Il l'envoie ensuite au serveur par l'intermédiaire de l'authentificateur. Le serveur peut alors choisir l'une des méthodes que le suppliant supporte et continuer le processus d'authentification normalement mais si le serveur ne supporte pas ces méthodes, il envoie un message *Echec* à l'authentificateur qui, après lui avoir retransmis ce message, met fin à la communication avec ce suppliant.

3.3.3. EAP dans Diameter :

Le protocole Diameter est conçu de manière à ce que des extensions du protocole de base soient possibles afin d'exploiter le fonctionnement de base de Diameter. Ces extensions sont appelées applications Diameter. C'est dans ce contexte que EAP est utilisé avec Diameter et est défini comme une application Diameter dite application EAP Diameter. Dans l'application EAP Diameter [56], l'authentification s'effectue entre le suppliant (l'utilisateur) et son serveur d'origine[9] Diameter par l'intermédiaire de l'authentificateur qui est aussi un client Diameter.

L'application EAP Diameter définit deux nouveaux messages qui sont : *Diameter-EAP-Request* (*DER*) et *Diameter-EAP-Answer* (*DEA*). Ces messages utilisent de nouveaux AVP pour transporter les messages requête/réponse de EAP (Annexe C :). Le message *DER* est envoyé par le client Diameter au serveur Diameter et transporte une *Réponse-EAP* envoyée par le suppliant. Le message *DEA* est envoyé par le serveur Diameter à l'authentificateur soit

[9] Serveur d'origine qu'on peut appeler Home serveur ou serveur mère. C'est le serveur qui est situé dans le domaine ou le réseau où l'utilisateur est inscrit.

parce que le message participe à une authentification à plusieurs phases, soit parce que le suppliant s'est authentifié avec succès et dans ce cas là le *DEA* contient le message *Succès-EAP*, soit parce que le suppliant ne s'est pas authentifié avec succès et dans ce cas là, il inclut un AVP indiquant un échec d'authentification.

Comme toute application Diameter, l'application EAP Diameter est signalée en utilisant les messages *Capabilities-Exchange-Request* (*CER*) et dans les messages *Capabilities-Exchange-Answer* (*CEA*) durant la phase d'échanges de capacités pour que le serveur et le client Diameter s'informent mutuellement que l'application EAP Diameter est supportée (Diagramme de séquence II-6).

La conversation entre le suppliant et l'authentificateur commence par la négociation de EAP par la couche liaison (PPP, IEEE 802.11i). Une fois l'EAP initié, dans le cas général, l'authentificateur envoie un message *DER* au serveur pour indiquer EAP-Start (Diagramme de séquence II-6). Si le serveur Diameter souhaite continuer l'authentification EAP, il répond avec un message *DEA* la *Requête-Identité-EAP*. A la réception du message *DEA* par l'authentificateur, la *Requête-Identité-EAP* est décapsulée et envoyée ensuite au suppliant pour que celui-ci s'identifie. Quand l'authentificateur reçoit la *Réponse-Identité-EAP* envoyée par le suppliant, il l'encapsule dans un message *DER* et il l'envoie ensuite au serveur.

Afin d'accomplir l'authentification, le client et le serveur s'échangent les messages *DER* qui transportent les *Réponse-EAP* du suppliant et les messages *DEA* qui transportent les *Requête-EAP* du serveur Diameter selon la méthode d'authentification EAP choisie. La conversation continue jusqu'à ce que le serveur Diameter envoie un message *DEA* indiquant le succès ou l'échec de l'authentification.

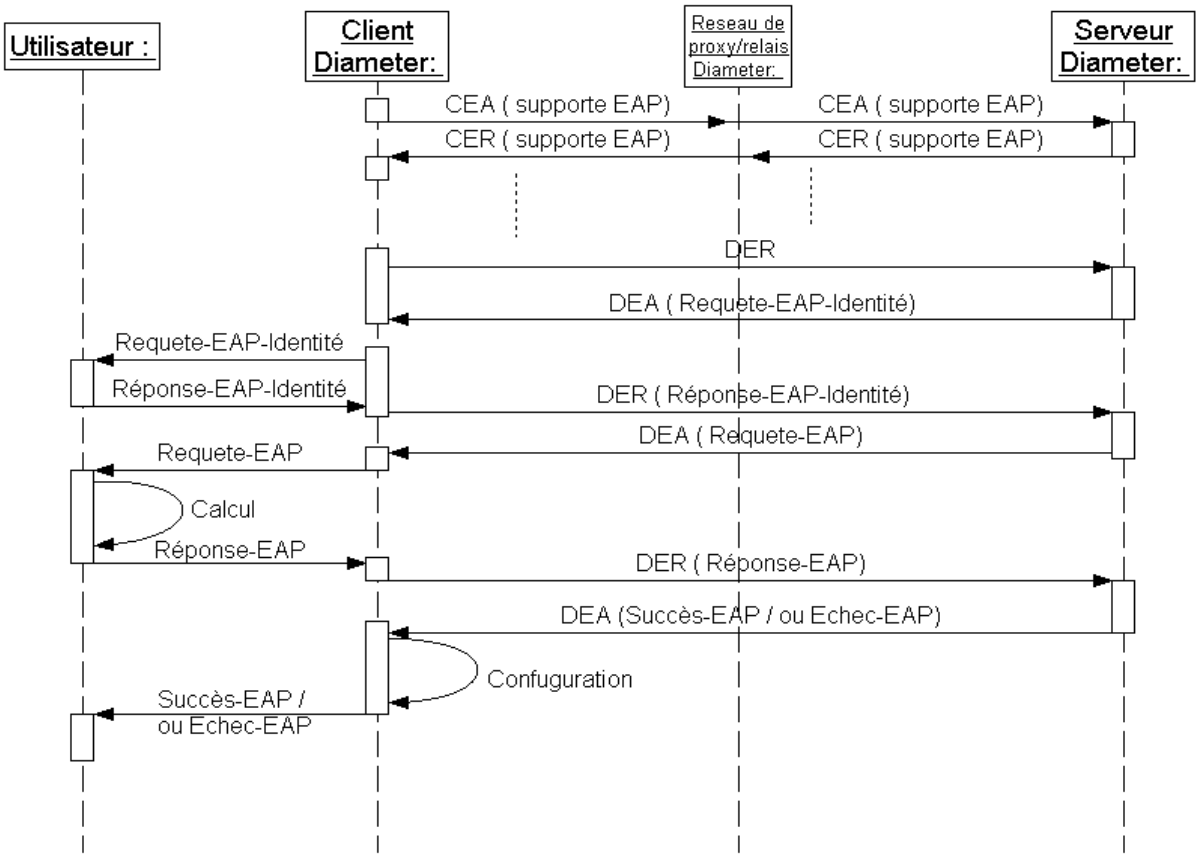


Diagramme de séquence II-6 : communication EAP/Diameter.

3.3.4. Le protocole IEEE 802.1X :

En plus de son utilisation dans PPP, EAP est aussi utilisé dans la couche liaison IEEE 802. La norme IEEE 802.1X [48] mise au point par l'IEEE définit la façon dont EAP est utilisé pour l'authentification par les périphériques IEEE 802, notamment les points d'accès sans fil IEEE 802.11 (borne WiFi) et les commutateurs Ethernet d'authentification. IEEE 802.1X diffère de PPP en ce sens que seules les méthodes d'authentification EAP sont prises en charge. Cette authentification par IEEE 802.1X intervient avant tout mécanisme d'auto-configuration (ex : DHCP, Dynamic Host Configuration Protocol). L'objectif de ce standard est donc de valider uniquement un droit d'accès physique au réseau depuis une connexion indépendamment du support de transmission utilisé et en s'appuyant sur des mécanismes d'authentification existants. Ce protocole s'appuie sur l'encapsulation EAP pour mettre en relation le serveur d'authentification et le système à authentifier. Le domaine d'application de ce protocole correspond donc à tous les modes de connexion pouvant être considérés comme des connexions dites point à point telles que la connexion réseau sans fil entre un poste utilisateur et une borne d'accès ou la connexion filaire entre un poste utilisateur et un commutateur.

Dans le fonctionnement du protocole, les trois entités qui interagissent (Figure II-9) sont le suppliant (utilisateur client), l'authentificateur (commutateur, borne WiFi) et un serveur d'authentification. L'authentificateur contrôle une ressource disponible via le point d'accès physique au réseau, nommé PAE (Port Access Entity). Le suppliant souhaitant accéder à cette ressource doit s'authentifier au près du serveur d'authentification. Dans cette phase d'authentification, l'authentificateur se comporte comme un mandataire (Proxy) entre le suppliant et le serveur d'authentificateur. La phase d'authentification est assurée grâce au protocole EAP et le IEEE 802.1X ne fournit qu'un cadre fonctionnel à l'interaction entre les équipements.

C'est au niveau du PAE que porte l'essentiel des modifications introduites par le protocole IEEE 802.1X. La principale innovation apporter par ce standard consiste à scinder le port d'accès physique au réseau en deux ports logiques qui sont connectés en parallèle sur le port physique. Le premier port logique est dit « contrôlé » et peut prendre un état sur les deux états « ouvert » et « fermé ». Le deuxième port logique est, lui, toujours accessible mais il ne gère que les trames spécifiques à IEEE 802.1X pour la communication EAP avec le serveur AAA (Figure II-5). La figure suivante montre le schéma général explicatif du protocole 802.1X et les entités participantes ; l'exemple pris est une connexion sans fil (WiFi).

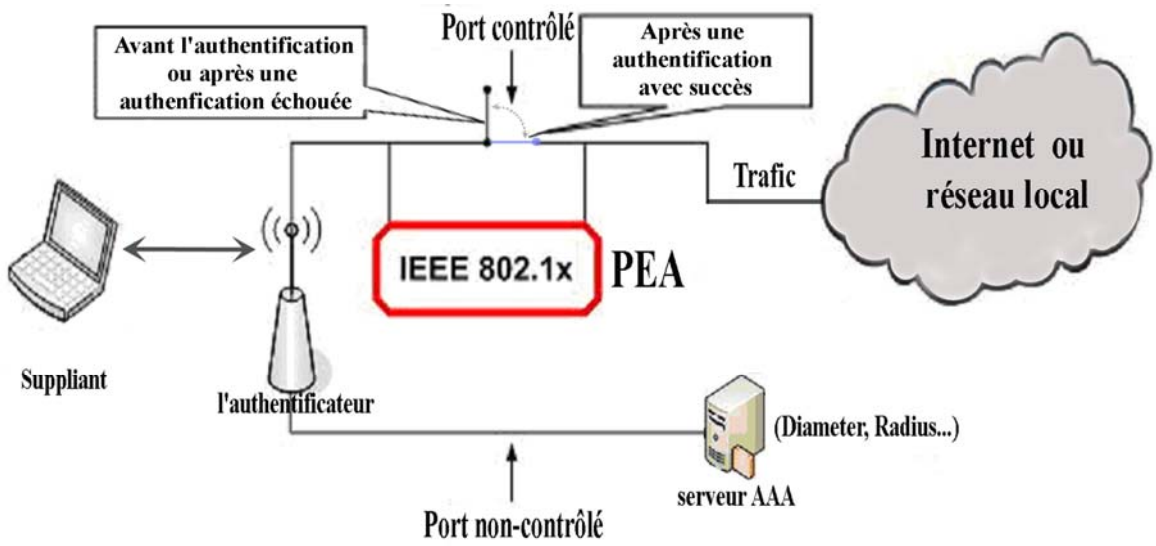


Figure II-9 : L'authentification avec IEEE 802.1X

Le dialogue entre l'authentificateur et le suppliant se fait en utilisant le protocole EAP. Les paquets EAP sont transportés dans des trames spécifiques avec le protocole EAPOL (EAP Over LAN) marquées avec un numéro de type spécifique. Le protocole EAPOL permet donc une encapsulation directe de EAP dans le protocole de la couche liaison. L'encapsulation EAPOL est définie pour les trois types de réseaux suivants : 802.3/Ethernet MAC, 802.5/Token Ring et FDDI/MAC. Ces trames EAPOL peuvent être de quatre types :

- *EAPoL-Start* permet au suppliant de prévenir le contrôleur d'accès qu'il souhaite se connecter au réseau,
- *EAPoL-Packet* est le paquet qui encapsule les paquets EAP,
- *EAPoL-Key* permet l'échange de clés de cryptage si le chiffrement est disponible,
- *EAPoL-Logoff* permet au suppliant de demander la fermeture de sa session,
- *EAPoL-Encapsulated-ASF-Alert* permet aux clients dont l'authentification a échoué de pouvoir tout de même être supervisés à distance (par exemple, par SNMP : Simple Network Management Protocol).

La communication entre l'authentificateur et le serveur d'authentification se fait par une simple ré-encapsulation des paquets EAP dans un format spécifique au serveur d'authentification (RADIUS, Diameter) sans modification du contenu du paquet par le système authentificateur. L'authentificateur effectue cependant une lecture des informations contenues dans les paquets EAPOL afin d'effectuer les actions nécessaires sur le port contrôlé en cas d'authentification réussie ou de bloquer ce port s'il y a une demande explicite du suppliant.

Avant la connexion du suppliant au port physique du PAE de l'authentificateur, le port contrôlé de celui-ci est bloqué (interrupteur ouvert : Figure II-5) et seul le port non contrôlé est accessible. Lorsque le suppliant se connecte au port physique de l'authentificateur, il reçoit un paquet EAP l'invitant à s'authentifier. La réponse du suppliant est reçue sur le port non contrôlé de l'authentificateur et est retransmise ensuite par ce dernier au serveur d'authentification. Par la suite, afin d'accomplir l'authentification EAP tel qu'elle a été décrite précédemment, un dialogue s'établit entre le serveur d'authentification et le suppliant par le biais du relais offert par le port non contrôlé du PAE de l'authentificateur.

Quand l'authentificateur voit passer une réponse de succès d'authentification en provenance du serveur d'authentification, il débloque son port contrôlé (interrupteur fermé : Figure II-9) en donnant ainsi au client authentifié l'accès au service bloqué derrière le port contrôlé. A partir de cet instant, le trafic est assuré normalement. Cependant, les automates implémentant le protocole restent actifs et peuvent à nouveau réactiver un processus d'authentification en cas, par exemple, d'une demande explicite du client ou du serveur ou de la déconnexion physique de l'utilisateur au réseau.

4. Gestion de la Mobilité IP :

La technologie actuelle permet de se connecter aux réseaux en utilisant une interface sans fil. A l'aide de terminaux portables, l'utilisateur peut se déplacer sur toute la zone de couverture d'un réseau sans fil ; de ce fait, il devrait changer constamment son point d'attachement au réseau. Un protocole qui permet de rendre transparent le mouvement de l'utilisateur au réseau sans perdre sa session est par conséquent indispensable. Mobile IP [1] est le protocole développé par l'IETF afin de rendre possible la Mobilité dans les réseaux IP notamment dans Internet. En plus de la gestion connue dans les réseaux sans fil et/ou fixe, les réseaux sans fil mobiles doivent, quant à eux, prendre en considération la gestion de la mobilité. La gestion de la mobilité permet de localiser le terminal mobile et de maintenir sa session pendant que celui-là se déplace.

Selon le déplacement du nœud mobile, on distingue deux types de mobilité. La macro-mobilité est considérée quand le MN se déplace d'un domaine à un autre pendant une session active. Elle concerne aussi l'activation d'une nouvelle session dans un réseau visité appelé aussi itinérance ou roaming. La micro-mobilité est considérée quand le nœud mobile se déplace à l'intérieur d'un même domaine. Dans cette section, nous présenterons les architectures et les protocoles associés à chacun de ces deux types de mobilité pour assurer la gestion de la localisation du mobile et la gestion du Handover ; nous commencerons par l'introduction des deux derniers aspects.

4.1. Gestion de la localisation :

Le protocole IPv4 identifie de manière unique toute machine connectée à Internet en utilisant l'adresse IP qui est constituée d'un préfixe de sous-réseau sur lequel la machine est connectée pour accéder à Internet (les bits du poids fort), et d'un identifiant qui identifie la machine au sein de son sous-réseau (les bits du poids faible). Les routeurs gardent en mémoire seulement les entrées correspondantes aux sous-réseaux et acheminent d'une manière identique les datagrammes qui ont le même préfixe. On conclue que tout terminal connecté au réseau dépend de son emplacement géographique (réseau auquel il appartient). Cela fonctionne correctement dans les réseaux fixes mais, dans les réseaux mobiles, la Mobilité introduit un nouveau problème de routage. Le nœud mobile peut changer de point d'accès et, de ce fait, le préfixe de son adresse IP ne détermine plus le sous-réseau sur lequel il se connecte.

A première vue, pour qu'un nœud mobile garde la possibilité de communiquer quand il change le point d'accès, on peut concevoir deux solutions :

- le nœud mobile change d'adresse IP à chaque fois qu'il change de point d'accès,
- des chemins spécifiques au MN doivent être propagés dans presque toute la structure de routage de l'Internet.

Ces deux méthodes ne sont malheureusement pas envisageables. La première n'est pas transparente et ne conserve pas la connexion au niveau de la couche transport (TCP, UDP) ou des couches supérieures lorsque le nœud mobile change de point d'accès ; la seconde pose des problèmes de passage à l'échelle « scalability ».

La gestion de la localisation dans la Mobilité consiste à résoudre le problème décrit ci-dessus afin de garder le nœud mobile joignable pour qu'il puisse recevoir et émettre les données quel que soit son emplacement dans le réseau Internet. Le principe de la solution utilisée dans les protocoles et les architectures standardisés est que le nœud mobile, à l'aide du sous-réseau qu'il visite, informe les entités fonctionnelles concernées pour que les paquets qui lui sont envoyés soient routés vers son nouvel emplacement.

4.2. Gestion du Handover :

Le Handover (ou Handoff, relais) est la procédure exécutée à chaque fois que le MN fait un passage d'un point d'attachement (ou point d'accès) à un autre. La gestion du Handover permet au réseau de maintenir la connexion de l'utilisateur du mobile pendant que celui-ci se déplace et que le MN continue à changer de point d'attachement. Plus précisément, le réseau d'accès offre des aptitudes particulières à minimiser l'interruption de la session de l'utilisateur en cours en terme de temps et de données. Donc, le but de l'étude du Handover est de réduire le temps de son exécution afin d'optimiser l'interruption de la connexion.

4.2.1. Profondeur du Handover :

Les réseaux d'accès sont un ensemble de domaines. Chaque domaine est relié par une ou plusieurs passerelles (ANG : Access Network Gateway) qui le connectent à Internet. Il se

constitue d'un ensemble de routeurs d'accès (AR : Access Router) dont chacun peut gérer un ou plusieurs points d'accès (AP : Access Point) (Figure II-10). Deux domaines sont administrativement indépendants quand ils n'appartiennent pas au même fournisseur. Afin de résumer la profondeur des différents types de Handover, la Figure II-10 montre un exemple illustratif d'un réseau d'accès regroupant deux domaines administratifs.

Au fur et à mesure que le nœud mobile se déplace, il s'attache aux points d'accès ; le Handover implique toujours le changement du point d'accès. On distingue quatre niveaux logique du Handover [11] selon que la route du flot de données d'un nœud mobile en mouvement implique comme éléments du réseau d'accès.

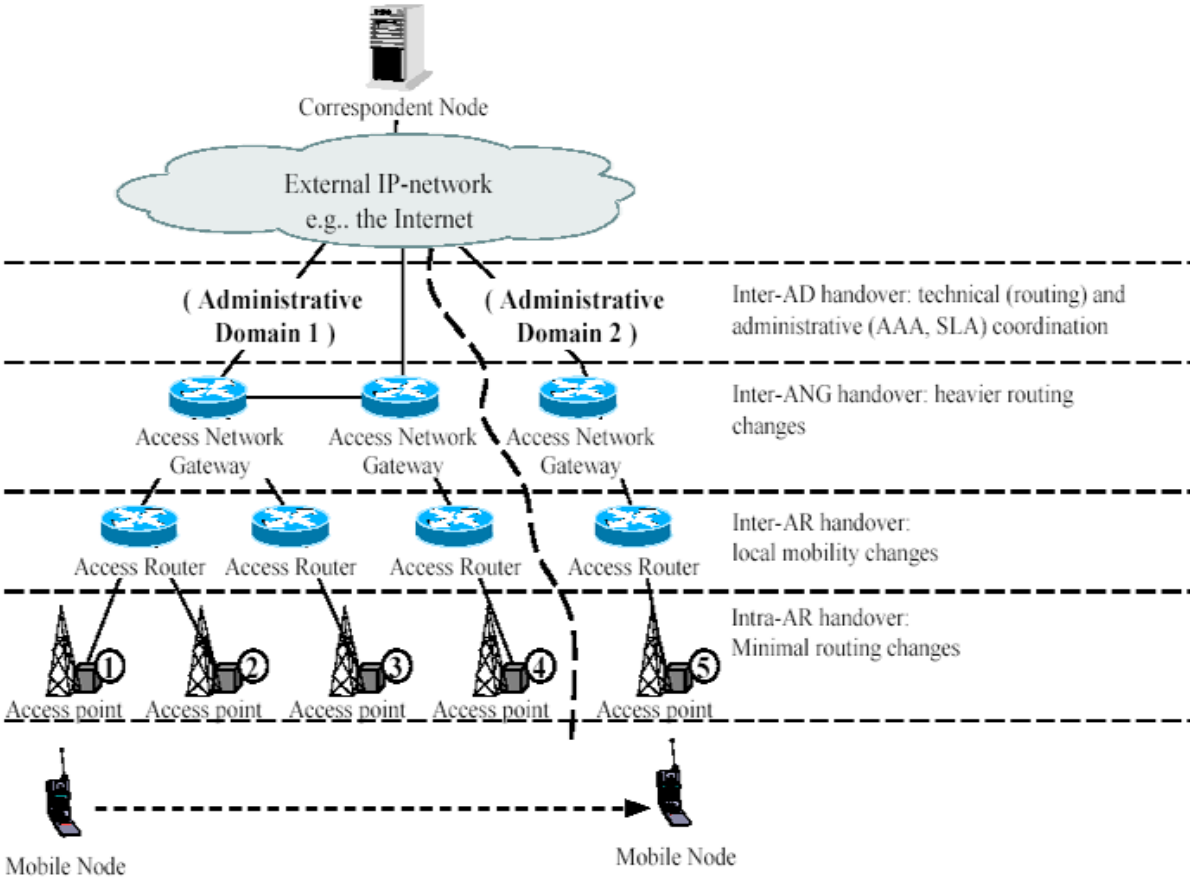


Figure II-10 : Illustration du Handover.

Handover Intra-AR est un Handover entre le point d'accès 1 et le point d'accès 2. Le nœud mobile change de point d'accès mais garde le même routeur d'accès. A ce niveau du Handover, le flot du nœud mobile déclenchant le Handover ne change pas de route entre le routeur d'accès et la passerelle. Ce Handover gère seulement la ressource radio et le contrôle d'admission peut être accompli aisément et rapidement par le routeur d'accès lors de l'initialisation du transfert. Ce Handover est souvent appelé Handover du niveau 2 car il peut être transparent à la couche IP si l'interface (technologie) au mobile n'a pas été changée.

Inter-AR (**Intra-ANG**) est un Handover entre le point d'accès 2 et le point d'accès 3. Le nœud mobile change de retour d'accès en gardant toujours la même passerelle. Dans ce Handover, même si la passerelle reste la même pour la route du flot du nœud mobile concerné, le Handover affecte, en plus la ressource radio, les ressources du réseau d'accès entre le routeur d'accès et la passerelle. Le nouveau routeur d'accès exécutera le contrôle d'admission pour ce nouveau flot.

Inter-ANG (**Intra-D**) est un Handover entre le point d'accès 3 et le point d'accès 4. Ce Handover implique le changement du routeur d'accès et de la passerelle mais il reste toujours dans le même domaine. La passerelle peut changer quand le nœud mobile se déplace dans un grand réseau d'accès qui contient plus d'une passerelle. Dans ce Handover, les ressources du réseau d'accès sont plus affectées. En plus de la gestion du Handover du niveau inférieur (Intra-AR et Inter-AR), une nouvelle adresse IP doit être attribuée au mobile.

Inter-D est un Handover entre le point d'accès 4 et le point d'accès 5. Dans ce Handover, le nœud mobile change complètement de réseaux d'accès. Par conséquent, il change de routeur d'accès, de passerelle et de domaine. On peut évidemment avoir un Handover Inter-D entre deux domaines du même fournisseur. Le Handover le plus complexe est quand le nœud mobile change du domaine administratif. En plus de la gestion de l'allocation de ressources réseau, de l'établissement de la nouvelle route et de l'attribution d'une nouvelle adresse IP au nœud mobile, le nœud mobile doit se ré-authentifier et être autorisé à l'utilisation des ressources et/ou des services du nouveau réseau d'accès ; les enregistrements de comptabilité doivent être aussi initialisés.

Le même Handover physique peut déclencher différents Handover logiques ; par exemple dans le cas d'un Handover Inter-ANG, en plus de la gestion de celui-ci, il y aura la gestion du Handover Inter-AR et du Intra-AR. De ce fait, les différents niveaux du Handover créent des charges de signalisation différentes ; plus le niveau du Handover est haut, plus le Handover se propage dans le réseau d'accès et demande plus de temps à installer la route vers le nœud mobile et à réserver les ressources.

4.2.2. Déroulement du Handover :

L'exécution de la procédure du Handover n'est pas considérée comme un bloc d'instructions indivisible. On peut distinguer des traitements qui doivent se faire au début de la procédure, avant les autres traitements ou à la fin de la procédure ; cela permet le découpage de la procédure du Handover en trois phases :

Phase d'initiation dont objectif est de reconnaître (confirmer) le besoin d'exécuter le Handover et de l'initier par la suite. Le Handover pourrait être exigé soit par le nœud mobile soit par le réseau. Généralement, il est initialisé quand le signal radio entre le nœud mobile et le routeur d'accès commence à se dégrader. Il pourrait toutefois être initialisé pour des raisons de gestion, de maintenance ou de changement des conditions du réseau comme dans le cas où il faut déplacer certain nœud mobile d'un point d'attachement surchargé vers un autre point d'attachement afin d'assouplir la charge sur le premier et offrir une meilleure communication.

Phase de décision dans laquelle, des mesures de la transmission radio entre le nœud mobile et les routeurs d'accès et des informations sur les règles à appliquer sont collectées. En fonction de ces informations, le meilleur routeur d'accès est sélectionné pour exécuter le Handover afin d'entamer la phase d'exécution. Cette phase peut être vue comme une phase de préparation à la phase d'exécution.

Phase d'exécution dans laquelle, le nœud mobile se détache de l'ancien routeur d'accès et s'attache au nouveau routeur d'accès. L'ordre des deux opérations n'est pas déterminé. De plus, certaines technologies permettent au nœud mobile de communiquer simultanément avec deux routeurs d'accès. Cette phase marque l'achèvement du Handover et le nœud mobile peut par la suite continuer sa communication.

4.2.3. Les types du Handover :

La procédure du Handover pourrait impliquer en plus du rétablissement de la route à travers le réseau fixe, de la réservation des ressources, de la négociation d'une nouvelle adresse IP et de la procédure d'enregistrement, d'autres traitements comme la récupération de

données en cours d'émission, l'anticipation de certaines procédures et l'optimisation d'exécution. Selon les priorités des ces traitements dans le Handover, la gestion du Handover et les possibilités offertes par la technologie, on peut classifier le Handover en plusieurs types.

La durée de la procédure de mise à jour de la localisation (d'enregistrement du mobile sect.I.4.3.1.1) et du rétablissement de la nouvelle route vers le mobile est assez conséquente. Pendant toute cette phase, les paquets de données qui sont destinés au nœud mobile et qui circulent encore dans le réseau sont, de toute évidence, routés vers l'ancien point d'attachement du mobile. Ces paquets sont susceptibles d'être perdus. Afin de pallier ces désavantages, quelques actions spécifiques peuvent être entreprises selon la phase du Handover. Par exemple, la phase de décision du Handover peut déclencher l'enregistrement anticipé ou la temporisation par l'ancien routeur d'accès des données émises vers le nœud mobile afin de les envoyer vers le nouveau routeur d'accès. Ces mécanismes ont pour but de réduire la durée totale du Handover et le taux des paquets perdus durant le Handover. Ces deux derniers critères définissent les trois types de Handover suivants : le **Handover sans couture (Smooth Handover)** est un Handover avec un minimum de pertes de paquets. Les efforts de la gestion du Handover tentent de récupérer tous les paquets (le maximum) pendant la procédure du Handover. Le **Handover rapide (Fast Handover)** est le type du Handover dont le paramètre déterminant est le temps. La gestion de Handover s'attache à minimiser le délai de la procédure du Handover. Le **Handover rapide et sans couture (Seamless Handover)** est le type du Handover qui combine les deux types précédents avec un minimum de délais et de pertes de paquets [11] [4].

Certaines technologies permettent au nœud mobile qui se trouve dans une partie commune des zones de couverture des points d'accès, de communiquer simultanément avec ces points d'accès. Cet avantage peut être exploité, par exemple, pour la réception, via l'ancien routeur d'accès, de données déjà envoyées au nœud mobile avant le Handover ou pour la signalisation avec le nouveau routeur d'accès avant le Handover. Quand cela est possible, le Handover est appelé **Soft Handover** ; dans le cas contraire on parle d'un **Hard Handover**[11].

Dans certaines gestions de Handover, le mobile utilise la connexion avec le routeur d'accès courant afin d'échanger des informations avec le réseau pendant la phase de décision de la procédure du Handover. Un **Handover planifié**, contrairement à un **Handover non-planifié**, est considéré quand le Handover requiert l'émission de messages de signalisation avant que le nœud mobile ne soit connecté au nouveau retour d'accès [11]. Cette signalisation est par exemple utilisée pour la construction d'un tunnel temporaire de l'ancien routeur d'accès vers le nouveau routeur d'accès afin de récupérer les données de l'ancienne route vers la nouvelle route.

Le **Handover contrôlé par le réseau** est considéré quand, au moment de l'initiation et de la décision du Handover, des entités dans le domaine gèrent l'établissement de la nouvelle liaison entre le nœud mobile et le (ou les) point(s) d'attachement(s) déterminé(s) par le réseau**.** Le **Handover contrôlé par le mobile** est considéré quand le nœud mobile est responsable de la détermination de son (ou ses) nouveau(x) point(s) d'attachement et de la gestion de la phase d'exécution [58]. Quand c'est le mobile qui collecte les informations qui seront utilisées pour décider de l'exécution du Handover, ce Handover est un **Handover assisté par le mobile** ; dans le cas où c'est le réseau qui fait la collecte d'informations, on parle d'un **Handover assisté par le réseau**[11]. Si le Handover est initié via le routeur d'accès courant, on parle d'un **Handover Backward**, sinon on parle d'un **Handover Forward** [11] .

Dans certains cas du Handover comme celui d'Intra-AR, le changement de point d'attachement peut être transparent à la couche IP. Le Handover est géré juste au niveau de la ressource radio. Un Handover est un **Handover du niveau 2** (couche liaison) quand la couche réseau n'est pas impliquée ; un Handover est un **Handover du niveau 3** quand la couche

réseau est impliquée [11][4]. Quand le mobile se détache physiquement d'un point d'attachement pour s'attacher à un autre, le Handover de niveau trois n'est exécuté qu'à la fin du Handover du niveau deux. Pour que le nœud mobile exploite les ressources du réseau, il doit attendre la fin du Handover du niveau trois.

Dans certains cas, la couverture d'une zone par le réseau n'est pas réalisée avec la même technologie. Par exemple une zone à forte densité de population, comme les aéroports, peut être couverte en utilisant la technologie WLAN et, dès que le mobile sort de cette zone celui-ci va se retrouver dans une zone de couverture avec une autre technologie assurant la couverture de cellules plus grandes. Un **Handover horizontal** ou **vertical** est considéré quand le nouveau et l'ancien routeurs d'accès ont respectivement la même ou différentes interfaces (technologie)[58] [11].

4.3. Macro Mobilité (inter-domaine):

Le réseau Internet est considéré comme un ensemble de réseaux. La macro-mobilité est quand le nœud mobile se déplace d'un réseau à un autre. La gestion de la mobilité rend le mouvement du mobile transparent aux couches transport et application qui s'exécutent indifféremment sur les nœuds fixes et mobiles. Cela permettra la mobilité sur Internet. Cette partie présente le protocole Mobile IP qui est conçu par l'IETF pour gérer la macro-mobilité. Nous présenterons comment ce protocole, en introduisant des entités fonctionnelles dans le réseau, apporte des solutions à la gestion de la localisation et du Handover. Nous présenterons ensuite après cette sous-section les solutions qui sont proposées dans la micro-mobilité afin de pallier les manques du Mobile IP au niveau local des domaines.

4.3.1. Le protocole Mobile IP :

Le protocole Mobile IP est conçu par l'IETF pour introduire la mobilité dans Internet tout en gardant la même pile de protocole IP. Les applications qui s'exécutent au-dessus du protocole standard IP s'exécuteront sans aucune modification sur le protocole Mobile IP. Il y a deux versions du protocole Mobile IP : Mobile IPv4 et Mobile IPv6.

4.3.1.1. Mobile IPv4 :

Afin d'exposer le protocole, quelques définitions [1] préalables de nouveaux concepts sont introduites :

Nœud Mobile "Mobile Node" (MN) est hôte ou routeur qui peut changer de point d'accès d'un sous-réseau à un autre tout en gardant la même adresse IP initiale.

Réseau d'origine "Home Network" (HN) est le sous-réseau qui a un préfixe réseau correspondant au préfixe réseau de l'adresse IP du MN. C'est le réseau où est inscrit le MN.

Agent mère ou Agent d'origine "Home Agent" (HA) est un routeur sur le Home Network auprès duquel le MN enregistre son adresse temporaire (Car of Address) quand il est loin de son Home Network (ou dans un réseau visité).

Réseau visité "Foreign Network" (FN) est tout réseau qui n'est pas Home Agent par rapport à un MN.

Agent visité "Foreign Agent" (FA) est un routeur sur le réseau visité par le MN qui co-opère avec le Home Agent afin de router les datagrammes au MN quand il ne se trouve pas dans son Home Network.

Adresse permanente "Home Address" et adresse temporaire "Care-of Address" (CoA) est l'adresse temporaire (CoA) attribuée au MN quand il est dans un Foreign Network.

Association de mobilité "Mobility Binding" est l'association de l'adresse temporaire, de l'adresse permanente et de la durée de vie de cette association.

Agent de mobilité "Mobility Agent " est un Home Agent ou un Foreign Agent.

Nœud Correspondant "Correspondent Node" (CN) est un Hôte (mobile ou fixe) qui dialogue avec un mobile.

4.3.1.1.1. Fonctionnement du protocole :

Mobile IP définit deux messages envoyés avec le protocole UDP et qui sont : *Registration-Request* et *Registration-Reply*. Le message *Registration-Request* est envoyé par le MN pour enregistrer sa CoA (demander la création ou la mise à jour de l'association de mobilité) auprès de sont HA à chaque fois que cela est nécessaire. Le HA utilise le message *Registration-Reply* pour répondre et informer le MN sur le résultat du traitement de sa requête et la cause de l'échec de la requête s'il y a échec.

Dans Mobile IP (MIPv4), trois entités fonctionnelles interagissent pour offrir les services de mobilité : le HA, le FA et le MN (Figure II-11). Le MN se déplace de réseau en réseau et s'attache à des à des Points d'Accès "Access Point" (AP). Il reçoit des messages d'annonce (Advertisement Message) émis par des Advertisement Agents installés dans les FA et utilise ces messages pour découvrir s'il est dans son HN ou dans un FN. Dans le premier cas, il communiquera avec d'autres machines sans utiliser les services de mobilité (rien est changé); dans l'autre cas, le MN envoie un message *Registration-Request* au FA. Le FA ensuite attribue une adresse temporaire (CoA) au MN et transmet *Registration-Request* au HA du MN. Le HA traite la requête, met à jour l'association de mobilité (l'adresse temporaire et la CoA) et renvoie un message *Registration-Reply* au FA. Le FA traite la notification du HA et retransmet le message *Registration-Reply* au MN (Diagramme de séquence II-7). A partir de ce moment, le HA intercepte tous les datagrammes à destination du MN et les envoie dans un tunnel qui a comme terminaison la CoA. La terminaison du tunnel est soit le FA qui, à son tour, décapsule les datagrammes et les transmet au MN, soit le MN lui même [1].

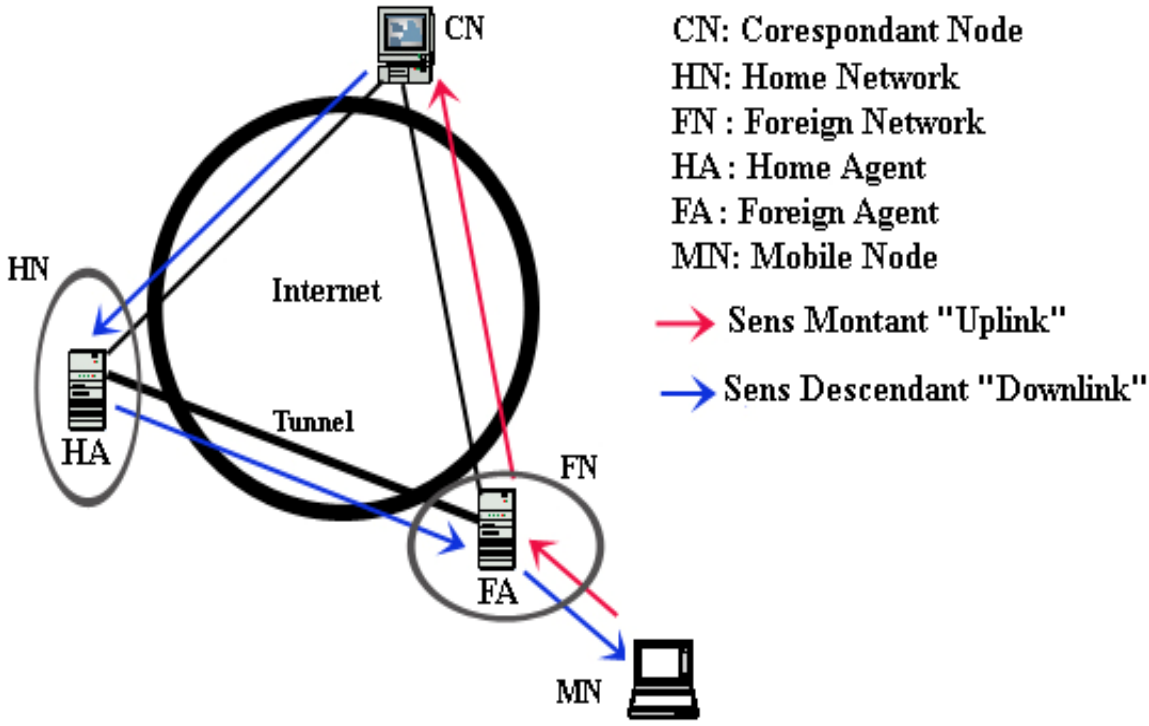


Figure II-11 : Architecture du Mobile IP

Quand le mobile envoie des paquets, il exploite le mécanisme standard IP en utilisant comme adresse source l'adresse permanente. Les datagrammes envoyés par le MN ne passent pas par le HA et sont délivrés directement au destinataire (Figure II-11).

4.3.1.1.1.1. Gestion de la localisation :

Quand le MN change de réseau, il doit détecter son mouvement. Les agents de mobilité (FA ou HA) signalent leurs présences en diffusant des messages d'annonce à des intervalles réguliers. Le MN écoute ces messages, les interprète et prend des décisions. Toutefois, si le mobile a besoin d'une CoA, il peut diffuser des sollicitations auxquelles répondront les Agents de Mobilité qui les ont reçues[2].

Lorsque le MN se trouve sous un nouveau réseau, il va essayer d'acquérir une nouvelle adresse temporaire qui sera la terminaison du tunnel de données envoyées par le CN. Cette adresse peut être obtenue soit par le nouveau FA soit dynamiquement par un mécanisme d'auto-configuration tel le DHCP (Dynamic Host Configuration Protocol [9]). Dans le premier cas, l'adresse est appelée FA-CoA (Foreign Agent Car-of Address) et c'est l'adresse du FA ; La fin du tunnel est, par conséquent, le FA lui-même. L'enregistrement de cette adresse par le MN auprès du HA se fait par l'intermédiaire de ce FA. Ce mode d'acquisition d'adresse temporaire a l'avantage de permettre le partage d'une seule CoA par plusieurs mobiles. Dans le second cas, l'adresse est appelée CCoA (Co-located Care-of Address) et est routable. Le MN enregistre cette nouvelle adresse temporaire auprès du HA avec des messages de demande et réponse d'enregistrement en utilisant comme adresse source la CCoA. La terminaison du tunnel est dans ce cas le mobile lui-même. Ce mode permet au MN de fonctionner sans le FA mais il pose un problème au niveau de l'espace d'adressage d'IPv4 donc du MIPv4.

Une fois la CoA obtenue, elle doit être enregistrer auprès de son HA pour accomplir la procédure de la mise à jour de la localisation. Cette procédure d'enregistrement crée ou met à jour l'association de mobilité " Binding Mobility" maintenu dans le HA. Elle la crée, si le MN quitte son HN pour rejoindre un réseau visité ; sinon elle la met à jour si le MN s'est déplacé d'un FN à un autre. Cette procédure associe l'adresse permanente et la CoA pour une durée pré-spécifiée. Pour la maintenir, le MN rafraîchit l'association de mobilité à des intervalles réguliers avec les messages *Registration-Request*[1].

Mobile IP définit deux méthodes d'enregistrement selon le type de CoA. Dans le cas d'une FA-CoA, le FA joue le rôle d'intermédiaire entre le mobile et son HA. Dans le cas d'une CCoA, le mobile n'aura pas obligatoirement besoin du FA pour s'enregistrer auprès du HA.

La première méthode se déroule comme suit (Diagramme de séquence II-7):

a) le MN envoie une demande d'enregistrement au HA par l'intermédiaire du FA,
b) Le FA traite la demande d'enregistrement et la transmet ensuite au HA,
c) HA envoie la notification d'acceptation ou de refus de la demande d'enregistrement au FA avec le message réponse d'enregistrement,
d) FA traite la réponse d'enregistrement et la transmet au MN.

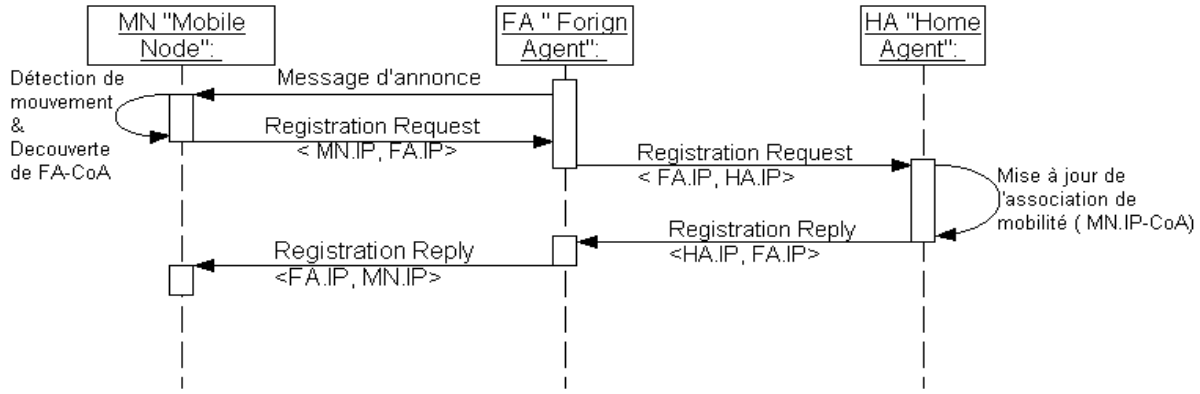


Diagramme de séquence II-7 : enregistrement de la FA-CoA dans Mobile IP.

Dans la deuxième méthode l'échange de messages se fait directement entre le MN et le HA comme suit (Diagramme de séquence II-8):

a) le MN procure une adresse IP (CCoA) chez le serveur DHCP,
b) le MN envoie une demande d'enregistrement au HA,
c) le HA envoie la notification d'acceptation ou de refus de la demande d'enregistrement directement au MN avec un message réponse d'enregistrement.

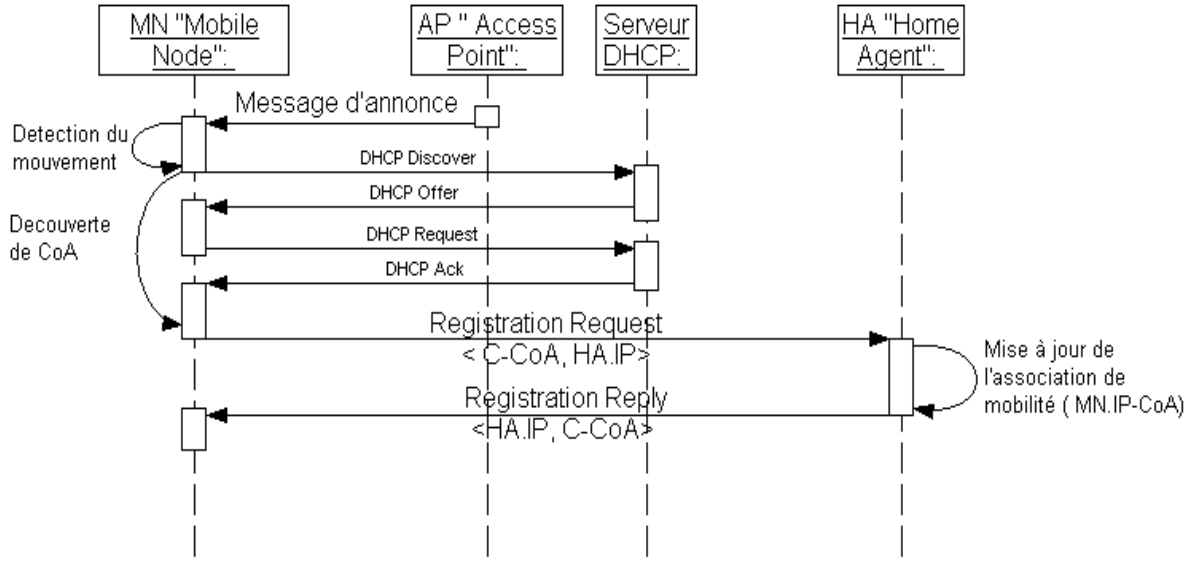


Diagramme de séquence II-8 : enregistrement de la CCoA da ns Mobile IP.

4.3.1.1.1.2. Gestion du Handover dans Mobile IP :

Dans Mobile IP (c'est à dire dans la macro mobilité IP), le Handover est profond. De ce fait, l'exécution d'un Fast Handover devient quasi-impossible. L'exécution d'un Fast Handover dépend beaucoup plus des performances et de la topologie du réseau que de la gestion du Handover. Dans ce qui suit, nous exposerons une solution avec un Smooth Handover [3] qui peut assurer pour le mobile une satisfaisante qualité de communication.

Les schémas du routage optimisé dans MIPv4 offre la possibilité au FA de garder la liaison avec ses anciens visiteurs MN par l'intermédiaire de leurs FAs. Les paquets arrivant à l'ancienne CoA sont reçus par l'ancien FA qui les transmet ensuite au nouveau FA après les avoir éventuellement empilés (Figure II-12.A). Le MN peut donc recevoir des paquets pendant la procédure Handover avec son ancienne CoA. Cependant, des paquets peuvent se perdre car le buffer est une ressource limitée. Dans ce cas, l'ancien FA va envoyer les paquets empilés vers le HA du MN concerné s'il n'a pas encore eu la nouvelle CoA (rafraîchissement) et que la durée de vie de l'association de mobilité est expirée. Le HA va à son tour transmettre ces paquets à la dernière CoA enregistrée par le MN (Figure II-12.B). Si le HA utilise toujours l'ancienne CoA, il ne doit pas décapsuler les paquets pour les ré-envoyer au même FA. Cela va créer un flot inutile de messages. C'est pour cela qu'un tunnel IP particulier est utilisé. Lorsque le FA envoie un datagramme, il l'encapsule avec son adresse comme adresse IP source. A sa réception, le HA compare l'adresse IP source du datagramme avec la plus récente CoA du MN. Si les adresses sont identiques, il ne va pas ré-envoyer le paquet vers le FA ; si ces adresses sont différentes, il le décapsule le datagramme et l'envoie vers la CoA actuelle du MN.

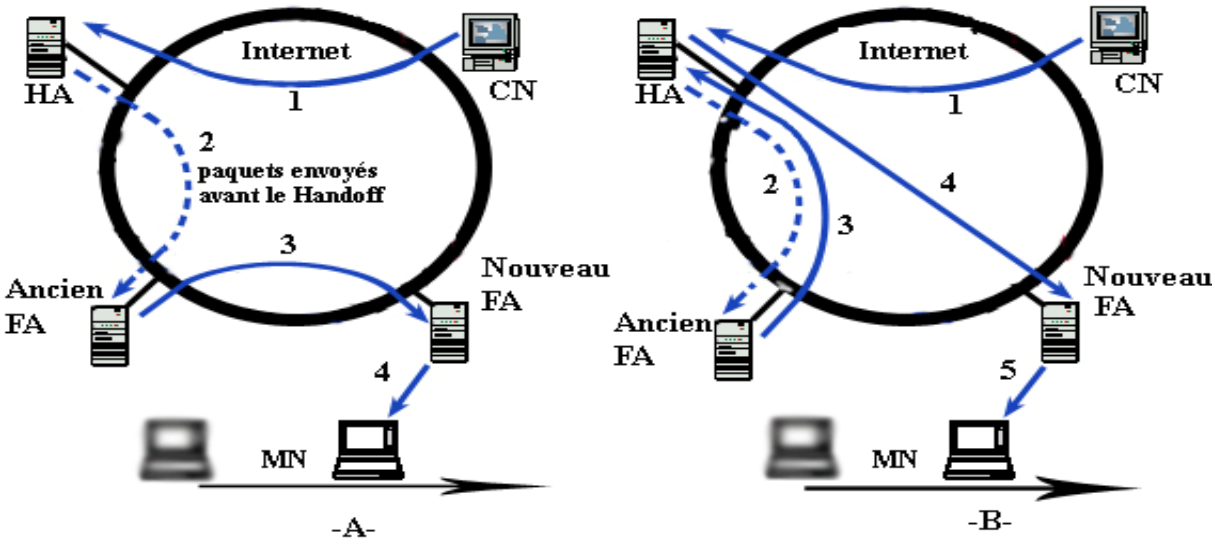


Figure II-12 : Smooth Handoff dans MIP
-A- **Rafraîchissement de Binding au FA** -B- **sans rafraîchissement de Binding au FA**

4.3.1.1.1.3. Routage dans Mobile IP :

En plus des fonctions de gestion de localisation et de gestion de Handover, une autre fonction de gestion de mobilité est le routage des paquets IP venant du CN vers le MN. Le mouvement du MN est complètement transparent à ses CN. Lorsqu'un CN envoie des paquets au MN, il utilise l'adresse permanente du MN comme adresse du destinataire. Si le mobile est dans son HN, le HA ne procède à aucun traitement particulier et les paquets sont acheminés normalement, sinon, quand le MN est dans un FN, le HA intercepte ces paquets (Figure II-13); il les redirige ensuite vers le FA dans le cas d'une FA-CoA ou vers le MN dans le cas d'une CCoA en utilisant un tunnel IP qui prend l'une des ces deux adresses comme terminaison. Pour réaliser le tunnel, Mobile IP utilise le protocole d'Encapsulation IP dans IP "IP Encapsulation within IP"[3]. Avec ce protocole la source du tunnel (HA) encapsule tout datagramme à destination du MN en ajoutant seulement un nouvel en-tête IP devant l'en-tête IP du datagramme. Le nouvel en-tête IP a pour adresse source l'adresse du HA et pour adresse destination la CoA.

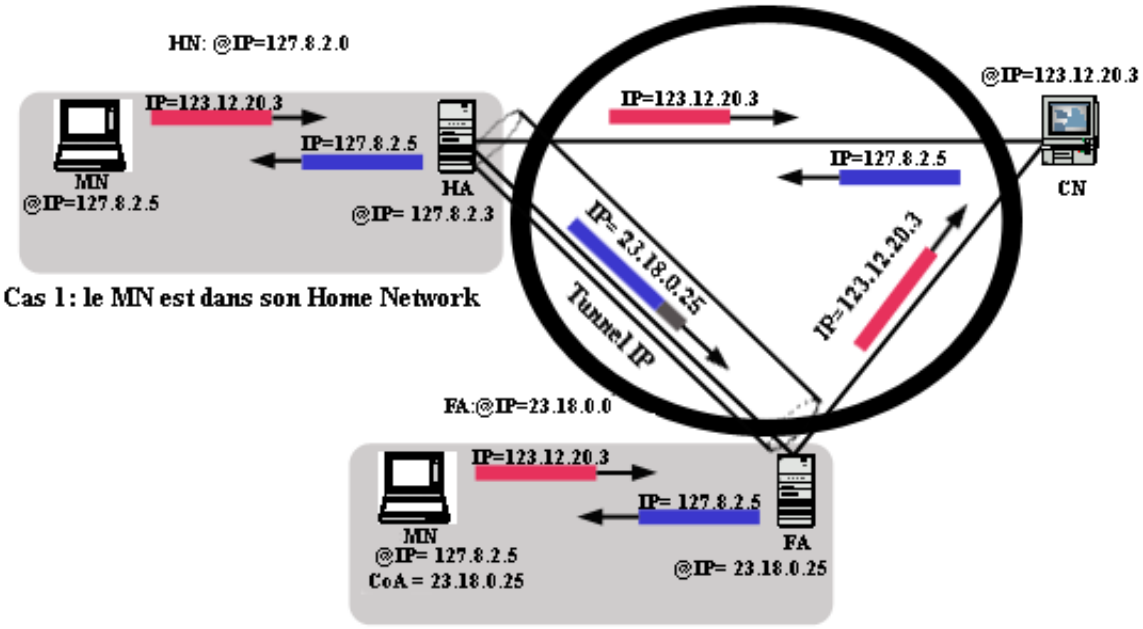


Figure II-13 :Tunnelling et routage dans Mobile IP

Ce routage est appelé le routage triangulaire et est l'une des faiblesses du protocole Mobile IP. Le passage obligatoire des paquets destinés au MN par le HA rend le chemin très long. Cela prend tout son poids quand le CN est dans le même sous-réseau que celui du MN après le déplacement. En effet, les paquets envoyés par ce CN vers le MN transiteront par le HA quelque soit l'emplacement de ce dernier. Afin d'éviter tout ce long chemin que les paquets empruntent, le Routage Optimisé [6] est proposé pour permettre le routage direct des paquets de CN vers le MN sans passer par le HA. Ce mécanisme, qui est une extension de Mobile IP, introduit de nouvelles méthodes pour les nœuds IP. En utilisant ce mécanisme, le CN reçoit des mises à jour de l'association de mobilité qui contiennent la CoA du MN de la part du HA. Le CN sauvegarde cette association et l'utilise pour construire son propre tunnel qui a comme terminaison la CoA. Toutefois, dans la phase d'initiation, les premiers paquets envoyés par le CN passent par le HA jusqu'à ce que l'association de mobilité envoyée par le HA soit reçue par le CN. Cette solution supprime la transparence du Mobile IP car le CN doit savoir s'il est en train de communiquer avec un nœud mobile ou avec un nœud fixe et elle modifie la pile de protocoles TCP/IP.

4.3.1.2. Mobile IPv6 (MIPv6) :

IETF espère que le protocole IPv6 [10] (ou IPng : IP Next Generation) remplacera IPv4 dans le futur proche. L'ordinateur portable mobile représentera sûrement une partie importante de la population d'Internet. Il est par conséquent indispensable d'offrir une gestion de mobilité dans IPv6. Le Mobile IPv6 proposé par IETF est le protocole de gestion de mobilité sur le IPv6. Les opportunités offertes par la version 6 du protocole IP comme le nombre d'adresse IP disponible ainsi que l'expérience acquise par Mobile IPv4 font profiter la conception du MIPv6 pour résoudre presque tous les problèmes et les limitations rencontrés dans la macro-mobilité. Toutefois, MIPv6 partage presque le même mécanisme que celui du MIPv4 mais il est complètement intégré dans IPv6 et offre plusieurs améliorations par rapport à MIPv4. Dans cette section, nous présenterons les différences entre le MIPv4 et le MIPv6 et les solutions apportées par MIPv6 [5][7].

- Le FA n'est plus indispensable pour MIPv6 quand le MN se trouve dans un FN. Le MN utilise un point d'accès pour s'attacher à son nouveau FN, le nombre d'entités de mobilités se trouve ainsi réduit. Les fonctionnalités de FA peuvent être accomplies par les caractéristiques de IPv6 tel Neighbour Discovery [8]et les Addresses Auto-configuration [9] (DHCP).
- Dans MIPv6, la Route Optimisée et la procédure d'enregistrement auprès de HA sont toutes deux réalisées par des nouveaux messages **"**Binding Updates**".** MIPv6 intègre le mécanisme du Routage Optimisé qui, auparavant, était dans MIPv4 une extension non standardisée qui n'est peut être pas supportée par tous les routeurs. C'est la solution qui est adoptée par MIPv6 pour résoudre le problème du routage triangulaire. Il est attendu que le routage optimisé soit déployé à l'échelle globale entre tous les MN et les CN.
- IPv6 introduit le concept d'en-tête d'option. Le protocole peut insérer des en-têtes entre l'en-tête de base d'IPv6 et les données. Dans MIPv6, la plupart des paquets destinés au MN, quand il n'est pas dans son HA, sont envoyés avec un en-tête d'Option Routing "Routing Header" sans avoir recours à l'encapsulation. Ainsi, la charge et le temps du traitement des paquets sont réduits considérablement par comparaison à MIPv4 qui, lui, est contraint de n'utiliser que l'encapsulation.
- L'en-tête d'Option de Destination de IPv6 qui transporte des informations optionnelles qui ne sont traitées que par le destinataire permet au trafic du contrôle de MIPv6 d'être transporté en superposition avec n'importe quel paquet IPv6 existant ; la charge est alors réduite. Par contre, dans MIPv4 et son extension de Routage Optimisé, des paquets UDP (User Data Protocol) sont utilisés séparément du trafic de données pour chaque message de contrôle.

▪ Dans IPv6, une nouvelle procédure de routage appelée "anycast" est introduite. Cette procédure est utilisée dans MIPv6 pour le mécanisme de découverte automatique du HA. Ce mécanisme permet de ne retourner qu'un seul message au MN alors que, dans le mécanisme de MIPv4 correspondant qui utilise une adresse broadcast, des messages séparés sont retournés au MN par chaque HA potentiel qui en a reçu la demande. Le mécanisme de MIPv6 est plus efficace et plus fiable car une seule réponse est retournée.

▪ Dans MIPv6, une option "Advertisement Interval" sur le "Router Advertisement" (équivalant à "Agent Advertisement" dans MIPv4) est définie. Elle permet au MN de décider lui-même combien de "Router Advertisments" sont autorisés à ne pas être reçus avant de considérer que son routeur est injoignable.

4.3.2. Le protocole HIP :

Comme nous l'avons étudié précédemment, dans l'Internet actuel, un terminal est identifié par son adresse IP qui dépend de la localisation topologique de ce terminal ; autrement dit, les adresses IP sont sémantiquement surchargées puisqu'elles identifient les terminaux et leurs localisations topologiques. Le protocole HIP (Host Identity Protocol [65][67]) introduit un moyen de séparer ces informations de localisation et de l'identité du terminal. Il introduit un nouvel "espace de nommage" de nature cryptographique pour identifier les terminaux indépendamment des adresses IP qui, elles, continueront à être utilisées pour le routage des paquets. Ce principe peut être exploité pour résoudre les problèmes liés à la gestion de mobilité. Dans cette section nous décrirons HIP et comment il est utilisé pour la gestion de la mobilité.

4.3.2.1. Description du Protocole HIP :

Une adresse IP sert à accomplir deux fonctions dans les réseaux IP. Elle sert de descripteur de la localisation topologique d'un nœud dans le réseau. Dans ce cas, elle est utilisée pour router les paquets d'un nœud source jusqu'au nœud destinataire. En même temps, elle est aussi utilisée pour identifier d'une façon unique le nœud dans tout le réseau. Le protocole HIP introduit un nouvel « espace de nommage » pour identifier de façon sécurisée les extrémités (autrement dit les terminaux) d'une communication. Concrètement, il s'agit d'attribuer à chaque extrémité, donc à chaque machine, un identifiant sécurisé et dérivé de sa clé publique qui sera ensuite utilisé, par exemple, au niveau des applications pour identifier les extrémités d'une communication. Cet identificateur est appelé HI (Host Identity). Le HI, étant une clé publique longue, n'est pas pratique dans toutes les actions. Pour une utilisation commode, le HI est passé par une fonction de Hache pour avoir un identifiant de 128 bits, appelé HIT (Host Identity Tag) ; un HIT correspond donc à un seul HI. Etant d'une longueur de 128 bits, le HIT peut être utilisé comme une adresse IPv6 pour les applications puisque l'adresse IPv6 est de même taille.

Quand HIP est utilisé, les couches supérieures, y compris la couche application, ne verront plus l'adresse IP ; ils verront plutôt le HIT du destinataire comme " adresse " de la destination. Les informations de localisation sont cachées dans la nouvelle couche introduite par HIP. De ce fait, l'adresse IP n'identifiera plus les nœuds et ne servira que pour le routage de paquets dans le réseau. De cette façon, ces couches supérieures deviennent indépendantes des adresses IP (v4 ou v6) et par conséquent de la localisation ; elles utilisent des identifiants sécurisés. Tout ceci est rendu possible grâce au protocole HIP qui assure la conversion entre les identifiants et les adresses IP. Le terminal recevant un paquet identifie la source en prenant la clé adéquate et décrypte le paquet. L'adresse IP qui est dans le paquet reste sans aucun intérêt pour l'identification de la source.

4.3.2.1.1. La nouvelle couche dans la pile TCP/IP :

Habituellement, les applications n'ont pas besoin de connaître les informations de localisation de leurs correspondants. Elles ont juste besoin de connaître leurs identités. C'est cet aspect de l'adresse IP que les couches supérieures prennent en considération. L'aspect du routage de l'adresse IP est considéré dans les couches inférieures (la couche IP). Le protocole HIP re-architecturer la pile TCP/IP autour de la nouvelle couche "Host Identity Layer" (Figure II-14) qui découplera les couches de transport haut niveau de la couche réseau. En particulier, la couche HI sera responsable du maintien du nouvel espace de nommage d'identificateurs pour leurs utilisations dans les couches de niveau supérieur.

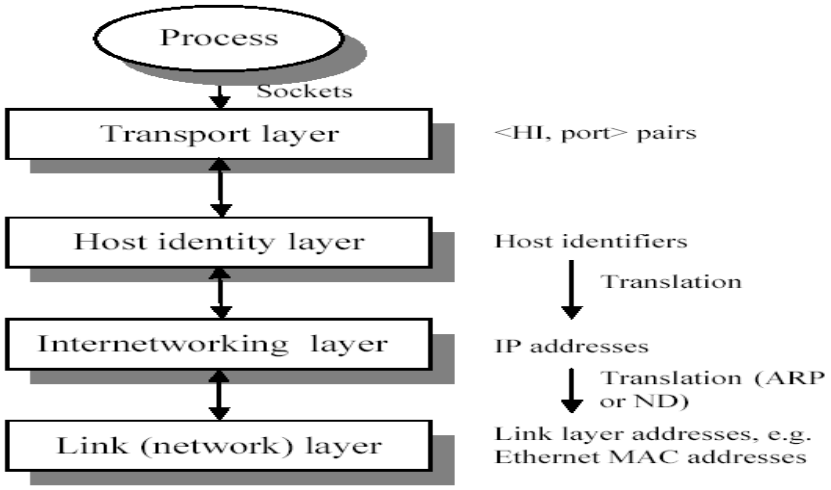


Figure II-14 : La nouvelle architecture de TCP/IP.

Les informations utilisées pour faire la correspondance entre le HIT et la localisation sont sauvegardées dans la couche HI ; celle-ci effectue un mapping entre ces identificateurs HIT et les adresses IP. Les HIT que les applications utilisent doivent être convertis aux adresses IP correspondantes avant que les paquets ne soient transmis par le terminal et après les avoir reçus. Dans cette couche, le HIT destinataire est donc converti à l'adresse IP destination correspondante et l'adresse IP source est convertie au HIT source correspondant.

4.3.2.2. HIP et la mobilité :

La mobilité IP est définie comme la situation où un terminal se déplace pendant qu'il garde ses communications actives. Cela veut dire que le terminal change de localisation topologique décrite par l'adresse IP en maintenant toutes les connexions existantes actives. En d'autres termes, bien qu'il change de localisation, le mobile reste toujours le même pour ces correspondants. On a vu dans les sections précédentes que la mobilité n'est pas assurée avec les mécanismes et les protocoles d'Internet actuels. Puisque l'adresse IP joue le rôle de l'identifiant du terminal, elle doit rester inchangée pendant toute la connexion d'un terminal. Toutefois, puisque l'adresse IP décrit la localisation, il est nécessaire qu'elle change avec le changement de la localisation du terminal dans le réseau. Ces deux fonctions sont contradictoires pour l'adresse IP.

Concernant la mobilité, la solution retenue à l'heure actuelle par l'IETF est le protocole Mobile IP (v4 et v6). Dans le cas de ce protocole, on a vu que la solution consiste à utiliser une localisation mère fixe offrant une adresse permanente pour le nœud mobile. L'adresse permanente joue le rôle de l'identifiant et permet une localisation stable pour le nœud mobile quand il quitte son HN. Les informations de localisation courante sont disponibles dans l'adresse temporaire (CoA) qui est utilisée pour le routage pour atteindre le nœud mobile qui a quitté son HA. Une autre solution à ce problème posé par la mobilité consiste à rendre

l'identification de terminaux indépendante de la localisation. C'est ce moyen que le protocole HIP rend possible[66]. L'exploitation de ce protocole qui sépare la localisation et l'identification dans la mobilité IP permet une gestion simple et évite l'utilisation du tunnel dans Internet.

Un nœud mobile HIP (HMN) se déplaçant dans le réseau peut changer continuellement de point d'attachement à l'Internet. Quand le routeur d'accès est changé, l'adresse IP change aussi. Le changement des informations de localisation (adresse IP) doivent être envoyées au nœud correspondant HIP (HCN). Cette nouvelle adresse pourrait aussi être envoyée à l'ancien FA du HMN. Le HMN peut de ce fait être joignable momentanément par l'ancien FA et les paquets déjà envoyés à ce HMN seront retransmis par ce FA.

La mobilité dans HIP définit un message de signalisation *REA* (readdress) qui contient l'adresse IP courante du HMN. Quand le HMN change de routeur d'accès et acquiert une nouvelle adresse IP, il génère un paquet *REA*, crypte le paquet avec la clé privé correspondant au HI utilisé et envoie ce paquet au HCN et à l'ancien FA (Diagramme de séquence II-9). Quand le HCN reçoit un paquet *REA*, il doit procéder à la vérification de l'adresse IP incluse dans ce paquet. Cela a pour objectif d'éviter l'acceptation de fausses adresses. Le HCN envoie à cette adresse incluse dans le paquet *REA* un message *AC* (Address Check). Quand le HMN reçoit un message *AC* qui correspond à un *REA* déjà envoyé, il répond avec un message *ACR* (Address Check Reply) pour confirmer la nouvelle adresse. Après avoir reçu le message *ACR*, la vérification de l'adresse est accomplie et le HCN peut mettre à jour les informations de la localisation (nouvelle adresse IP) du HMN.

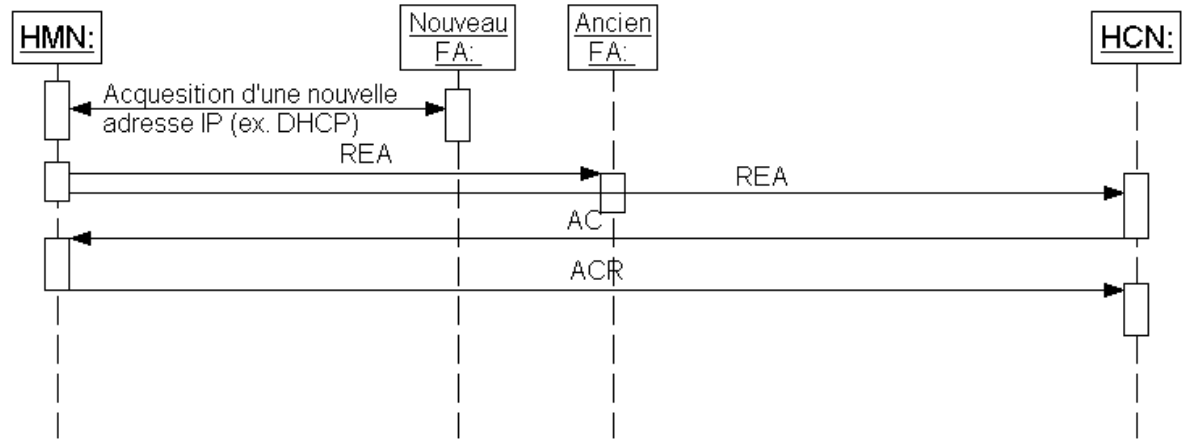


Diagramme de séquence II-9 : gestion de mobilité avec HIP.

4.4. Micro Mobilité (intra-domaine) :

Un domaine est un sous-réseau qui couvre une zone géographique bien déterminée. A l'intérieur du domaine, le mouvement du MN pourrait être très fréquent. L'architecture de Mobile IP n'est pas adéquate à ce genre de mouvement. Si le HA est loin du FN auquel le MN est connecté, la procédure d'enregistrement qui implique l'interaction des entités de mobilité génère une grande quantité de signalisation dans le backbone de l'Internet ; ce qui induit une longueur de la durée du Handover et l'augmentation de pertes de paquets destinés au MN. Le principal but de Mobile IP étant d'offrir une gestion de mobilité et d'assurer la continuité du service après le déplacement du MN. Mobile IP résout le problème de la macro mobilité ; il permet au mobile de se déplacer d'un sous-réseau (domaine) IP à un autre. Cependant, dans la gestion des déplacements à l'intérieur d'un domaine, il s'est avéré que les solutions de Mobile IP ne sont pas optimales. L'étape suivante consiste à optimiser la signalisation dans le réseau et les performances du Handover et la micro mobilité tente de réaliser ces attentes. La micro

mobilité à pour objectif de rendre le mouvement dans le domaine transparent au HA et/ou aux CN en utilisant une gestion locale de la mobilité. La gestion de la micro mobilité permet à des MN de s'enregistrer localement à l'intérieur du réseau qu'ils visitent. Ce type d'enregistrement réduit la quantité et surtout le délai de signalisation ; ce qui améliore les performances du Handover. Dans cette section nous exposerons les protocoles les plus connus de la gestion de la micro mobilité.

Le nombre de protocoles de micro mobilité est important et, afin de permettre une simplicité d'observation, on introduit une classification qui distingue deux types d'architecture [11] qui sont présentées sur Figure II-15 : l'Architecture à base d'agent de Proxy (Proxy-Agent Architectures PAA) et l'Architecture à base de routage localisé (Localized Enhanced-Routing Schemes LERS). La

Figure **II-15** montre les différents protocoles de la micro mobilité qui seront brièvement expliqués dans cette sous-section.

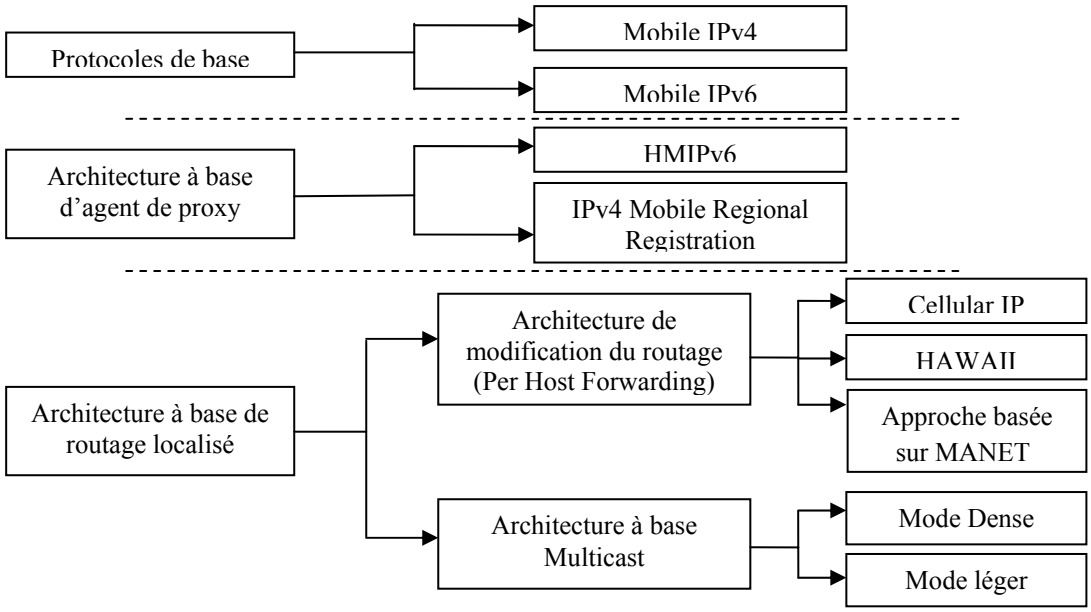


Figure II-15 : Classification des protocoles de la mobilité IP [11]

4.4.1. Architecture à base d'agents de proxy :

Afin d'éliminer la signalisation générée par la procédure d'enregistrement entre le HA et le FA, l'architecture à base d'agents de proxy (PAA) [11] met en place une hiérarchie de FA dans chaque domaine. Le seul FA qui communique avec le HA quand le MN met à jour sa localisation est le FA du plus haut niveau de la hiérarchie. Pour contacter le MN, chaque FA d'un niveau quelconque sait quel FA du niveau inférieur joindre et le MN est attaché à l'un des FA du niveau le plus bas de la hiérarchie dans le domaine. Le principe du routage à l'intérieur d'un domaine est simple : Un tunnel IP est construit de chaque FA d'un niveau donné vers le FA du niveau inférieur suivant. Quand le HA envoie les paquets destinés au MN vers le FA le plus haut, les paquets sont routés dans ces tunnels jusqu'au mobile.

Il est clair que le routage utilisé dans cette architecture est moins performant surtout quand le nombre de niveaux commence à augmenter ; par contre, le fait d'éviter la propagation de l'information d'enregistrement implique la réduction de la durée de la procédure du Handover.

Deux protocoles de cette famille d'architecture seront présentés dans ce qui suit. Le premier est Hierarchical MIPv4 et le deuxième est Hierarchical MIPv6.

4.4.1.1. Enregistrement Régional dans Mobile IP (HMIPv4) :

L'enregistrement Régional dans Mobile IP est une extension de MIPv4 et introduit deux niveaux de FA. Un domaine comporte une passerelle qui occupe le plus haut niveau de la hiérarchie des FA. Sous cette passerelle (GateWay Foreign Agent :GFA) qui relie le domaine à Internet, il y a un ou plusieurs FA régionaux (Regional Foreign Agent : RFA). Quand le MN se déplace à l'intérieur d'un domaine, il enregistre sa nouvelle CoA locale auprès de la GFA par l'intermédiaire de son nouveau RFA ; il fait un enregistrement local. Par contre quand le MN entre dans un nouveau domaine, la CoA qui est l'adresse du nouveau GFA est enregistrée auprès du HA. A l'intérieur du domaine, le mouvement du MN est complètement invisible au HA qui, pour lui, le MN est toujours localisé sous la GFA. Cette nouvelle procédure d'enregistrement implique des extensions au message d'Agent Advertisement et l'introduction de nouveaux messages d'enregistrement[12].

En plus des avantages offerts par l'enregistrement local comme la diminution de la durée de la procédure d'enregistrement ainsi que celle du Handover lorsque le MN se déplace à l'intérieur du FN, cette architecture nécessite peu de modifications par rapport au MIPv4.

4.4.1.2. HMIPv6 (Mobile IP Hiérarchique):

Une amélioration de MIPv6 est HMIPv6. Cette architecture met en place une passerelle dans le réseau visité. Sur cette passerelle, un FA est installé pour former un agent de mobilité appelé MAP (Mobility Anchor Point). Le MAP joue le rôle du HA au niveau local. Il assure la procédure d'enregistrement régionale et, tant que le MN est dans son domaine, le mouvement du MN reste transparent au HA. Le MN aura trois adresses au lieu de deux : L'adresse permanente qui ne change pas pendant toute la connexion, une adresse transitoire CoA qui, attachée au MAP, reste valide pendant la présence du MN dans le réseau visité et une adresse routable CCoA attribuée au niveau du réseau visité. Le seul moment où le MN s'enregistre auprès du HA est celui où il change de MAP, sinon pendant tous les déplacements au sein du même réseau visité, il s'enregistre auprès du MAP. Le MAP garde la correspondance entre la CCoA et la CoA et le HA garde la correspondance entre la CoA et l'adresse permanente. Le routage des paquets reste le même avec celui de MIPv6 puisque le HIMPv6, lui aussi, utilise les avantages de IPv6[4].

4.4.2. Architecture à base de routage localisé :

Le principe de l'architecture à base de routage (Localised Enhanced-Routing Schemes : LERS) dans la micro mobilité est de changer le routage IP par un routage particulier seulement au niveau du domaine ; ainsi, pendant le séjour du MN à l'intérieur d'un domaine, son adresse IP reste inchangée. Dans cette architecture, on peut distinguer trois approches : l'une basée sur la modification de routage localisé, l'autre basée sur l'adressage multicast et la dernière est basée sur le routage ad hoc (Manet) [4]. Nous présenterons dans ce qui suit ces protocoles et leurs architectures.

4.4.2.1. Approche à base de modification de routage localisé :

Cette architecture utilise un protocole de routage particulier de niveau 3. Ce protocole construit une route pour chaque terminal qui se trouve dans le domaine en utilisant la signalisation du terminal. Ces routes sont de type « soft-state » ; elles ne sont plus valide après un time-out et doivent être rafraîchies. Une route est une chaîne d'entrées de table de routage des routeurs. Dans le sens descendant, elle commence de la passerelle qui relie le domaine à Internet et finie par le routeur auquel le MN est attaché. Dans le sens montant, la route du MN vers la passerelle est prédéfinie par défaut puisque le routeur transmet les paquets vers une

interface de sortie correspondant à la passerelle. Cette technique est celle utilisée par les protocoles Cellular IP et HAWAII qui seront présentés.

4.4.2.1.1. Cellular IP :

Cellular IPv6 se base sur Cellular IP (CIP) dont il est très proche[4][5]. La principale différence étant que Cellular IPv6 est basé sur IPv6. En particulier, les messages de signalisation de Cellular IPv6 sont des paquets IPv6 avec une option d'en-tête « hop-by-hop ». Cellular IP est un protocole qui interagit avec le Mobile IP pour accomplir la gestion de la macro mobilité. Il est inspiré de la gestion de mobilité dans les réseaux Télécom cellulaires et est conçu pour être placé à un niveau local. Un réseau CIP est connecté à Internet par l'intermédiaire d'une passerelle (Gateway : GW). Il est constitué d'un ensemble de routeurs particuliers et d'un ensemble de stations de base (Base Station : BS).

L'une des particularités de CIP est qu'il fait la distinction entre les mobiles actifs (active) qui se déplacent dans le réseau et qui sont en communication et les mobiles oisifs (idle) qui ne sont ni en transmission ni en réception de paquets mais veulent rester joignable pour une réception. Les routeurs CIP maintiennent deux caches : le cache de routage (routing cache) et le cache de paging (paging cache). Le cache de routage est utilisé pour localiser les MN actifs pendant que le cache de paging est utilisé pour localiser les MN oisifs. Les deux caches d'un routeur sont créés et mis à jour en utilisant des paquets envoyés par le MN et qui montent jusqu'à la passerelle. Dans le sens montant, les paquets sont routés de la BS jusqu'à la passerelle saut-par-saut avec une route optimale sans prendre en considération l'adresse du destinataire. Les routeurs présents le long du chemin de ces paquets construisent une correspondance (mapping). Cette correspondance est l'association de l'adresse permanente du MN, de l'interface par lequel le paquet IP est reçu et du temps timeout de validité. De cette façon, la séquence de ces correspondances forment la route inverse pour atteindre le mobile à partir de la passerelle.

Quand le MN veut envoyer des paquets, il les transmet vers la BS la plus proche. Dans le sens descendant, la passerelle reçoit des paquets qui sont destinés à un MN qui se trouve dans son domaine. En utilisant le cache du routage, chaque routeur connaît l'interface vers laquelle il doit envoyer les paquets pour qu'ils atteignent le MN. Tous les paquets destinés au MN dont le cache de routage n'est plus valide, sont routés en utilisant le cache paging et diffusés sur l'ensemble de cellule de la zone paging de ce MN et, si le cache paging n'est aussi pas disponible, les premiers paquets sont diffusés dans tout le domaine.

Le Handover dans le réseau Cellular IP est initialisé par le MN. Quand le MN se déplace et s'approche d'une nouvelle BS, il redirige ses paquets de l'ancienne BS vers la nouvelle BS. La reconfiguration du cache de routage est accomplie par le premier paquet IP qui est redirigé vers la nouvelle BS. Les nouveaux routeurs traversés créent une correspondance de cache de routage vers ce mobile. Au nœud d'attachement (le cross-over router) où se rejoignent l'ancienne et la nouvelle route, la correspondance vers la nouvelle interface de sortie à destination du mobile est mise à jour. Pendant le temps de timeout qui reste de l'ancien cache de routage, tous les paquets destinés au MN vont être routés vers la nouvelle et l'ancienne BS en même temps. L'ancienne route est systématiquement éliminée après que le timeout est expiré. La nouvelle BS continue à recevoir les paquets IP du MN pour les lui délivrer.

Cellular IPv6 est un protocole avantageux pour l'amélioration des performances du Handover et est efficace dans l'utilisation des ressources du réseau. Cependant le fonctionnement du réseau dépend d'une seule passerelle. Si elle tombe en panne, le domaine complet est non opérationnel. Aussi, les paquets envoyés par un CN qui se trouve sous le même réseau que le MN passeront obligatoirement par la passerelle. Un autre inconvénient de CIP est qu'il n'est pas transparent pour un mobile qui utilise le protocole Mobile IP.

4.4.2.1.2. HAWAII :

HAWAII (Handoff-Aware Wireless Access Internet Infrastructure)[4] est un protocole transparent au MN qui n'a besoin de connaître que Mobile IP. L'architecture du réseau HAWAII est divisée en domaines organisés en arbres. Chaque domaine a une passerelle appelée "Demain Root Router" qui le connecte à Internet. Un MN a une adresse IP permanente et un Home Domain. Dans le cas où le MN reste dans son Home Domain, son adresse IP est maintenue. Quand le MN se déplace vers un Foreign Domain, il obtient une CoA qui reste inchangée pendant son séjour dans ce domaine même s'il change de routeur d'accès. Les paquets destinés au MN sont reçus par Domain Root Router à l'aide du mécanisme classique de Mobile IP. Ces paquets sont routés en utilisant un chemin spécifique au MN et qui est établie dynamiquement au fur et à mesure que le MN se déplace à l'intérieur du domaine. Dans cette approche, des entrées de routage spécifiques à chaque MN sont installées dans les routeurs du domaine. L'installation de ces entrées est réalisée en utilisant trois messages différents appelés " path setup schemes". Ces messages sont : path *setup power-up* pour la création du chemin, *path setup update* pour la mise à jour du chemin et *path setup refresh* pour le rafraîchissement d'un chemin existant.

Lorsque le MN est mis sous tension ou entre dans un nouveau Foreign Domain, il entame sa procédure d'enregistrement de CoA habituelle en émettant un message *MIP Registration-Request* à la BS à laquelle il va s'attacher. La BS envoie un message *Path Setup Power-up* au Domain Root Router qui est traité saut-par-saut pour construire le chemin inverse vers le MN. Le Domain Root Router transmet la demande auprès de HA ensuite il répond par un acquittement vers la BS qui transmet à son tour le message *MIP Registration-Reply*. A l'intérieur d'un domaine, le MN renouvelle ses enregistrements MIP en envoyant un message *MIP Registration-Request* vers la BS. La BS transmet un message *Path Setup Refresh* au Domain Root Router saut-par-saut. Ce message maintient les entrées de commutation vers le mobile dans les tables de routages de chaque router appartenant au chemin de Domain Root Router vers le MN. Le message utilisé pour mettre à jour le chemin quand le MN change le router d'accès est le message *Path Setup Update*. En envoyant ce message par les BS et les routeurs qui sont entre le chemin des deux BS, la route est changée à partir du routeur d'attachement (cross-over router) à l'endroit où l'ancienne route et la nouvelle se rejoignent.

Il existe deux schémas pour la procédure de mise à jour :

Le schéma avec transmission (forwarding scheme : FS) :

Ce schéma est utilisé pour réaliser un Smooth Handover. Après que le MN s'est attaché à la nouvelle BS, l'ancienne BS retransmet tous les paquets qu'elle a reçus vers la nouvelle BS. Il y a deux variantes de ce schéma. La première est celle de "Multiple Stream Forwarding" (MSF) : Un message de contrôle est envoyé directement à l'ancienne BS pour qu'elle change l'entrée de commutation (adresse IP → interface de sortie) afin d'envoyer les paquets destinés au MN vers une nouvelle interface. Cette méthode ne fait pas perdre de paquets pendant le Handover mais a l'inconvénient de désordonner fortement les paquets IP qui arrivent à la nouvelle BS. En fait, les routeurs entre l'ancienne BS et le routeur d'attachement peuvent déjà avoir changé d'interface de sortie après avoir envoyé des paquets destinés au MN à l'ancienne BS et de ce fait ils retransmettent des paquets plus récents vers la nouvelle BS.

Pour pallier ce désagrément une deuxième variante de FS, qui est "Single Stream Forwarding" (SSF), est mise en place au détriment du délai du Handover. Dans cette méthode, les entrées de commutation sont modifiées de telle sorte qu'elles contiennent l'interface de sortie pour une adresse IP et une interface d'entrée (interface d'entrée, adresse IP → interface de sortie). Elles sont modifiées en commençant par le nouveau routeur d'accès allant vers l'ancienne BS. Le routeur d'attachement et ceux qui sont entre le routeur d'attachement et l'ancienne BS continuent à router les paquets destinés au MN vers

l'ancienne BS jusqu'à ce que celle-ci commence à retransmettre ces paquets vers la nouvelle BS. A partir de ce moment, le routeur d'attachement redirige le flot de paquets vers l'interface de sortie destinée au MN.

Le schéma sans transmission (non-forwarding scheme : NFS) :

Le NFS est un schéma sans retransmission de paquets de l'ancienne BS vers la nouvelle BS. Le flot des paquets est immédiatement redirigé vers la nouvelle BS par le routeur d'attachement dès la réception du message "*Path Setup Update*". Son principal but est de réduire le délai du Handoff. Deux variantes de ce schéma existent. La première est celle "Unicast Forwarding Scheme" (UFS) dans laquelle le routeur d'attachement redirige le trafic seulement vers la nouvelle BS. Les paquets déjà envoyés vers l'ancienne BS sont reçus par celle-ci et émis sur l'interface radio. UFS est optimisé pour les mobiles capables d'écouter sur plusieurs interfaces simultanément. Pour les mobiles dont ce n'est pas le cas, il existe la deuxième variante "Multicast Forwarding Scheme" (MNF). Dans cette variante, le routeur d'attachement diffuse les paquets vers l'ancienne et la nouvelle BS en même temps (bi-cast) jusqu'à ce que l'ancienne BS lui envoie un message de contrôle.

HAWAII investit principalement dans la gestion du Handover. Il offre plusieurs techniques de Handoff et on peut en choisir une selon type du Handover qu'on a à privilégier.

4.4.2.1.3. Approche basée sur MANET :

Un réseau ad hoc est une collection d'entités (nœuds) interconnectées par une technologie sans fil formant un réseau temporaire sans l'aide de toute administration ou de tout support fixe ; tout nœud du réseau restant libre d'intégrer ou de quitter le réseau. A condition qu'il y ait suffisamment de nœuds dans une zone, le réseau s'adapte spontanément pour répondre au besoin de communiquer et se configure de façon complètement autonome et dynamique en fonction des possibilités de connexions existantes entre ces nœuds. Lorsque les nœuds des réseaux ad hoc sont mobiles, on parle de MANET (Mobile Ad hoc NETwork). La topologie de MANET peut changer à tout moment ; elle est donc dynamique et imprévisible. Ce qui fait que les déconnexions des nœuds peuvent être très fréquentes. L'absence d'infrastructure ou de réseau filaire composé de stations de base oblige les unités mobiles à se comporter comme des routeurs qui participent à la découverte et la maintenance des routes pour les autres nœuds du réseau ; chaque nœud communique directement avec son voisin. Lorsqu'une machine veut communiquer avec une autre se trouvant hors de sa portée radio, chaque nœud actif du réseau sert de routeur pour ses voisins. Pour cela, les entités se situent les unes par rapport aux autres et construisent des routes entre elles[4][60].

Suivant la manière de création et de maintenance des routes pour l'acheminement des données, les protocoles de routage MANET peuvent être séparés en deux catégories : les protocoles proactifs et les protocoles réactifs[4].

Les protocoles proactifs calculent les routes à l'avance en continu. Chaque nœud met à jour une ou plusieurs tables de routage par échange de paquets de contrôle entre voisins. Grâce à la connaissance en permanence de la topologie du réseau, le transfert des données est plus rapide. En contrepartie, le trafic généré par l'échange de paquets de contrôle fait baisser la bande passante disponible. L'un des principaux protocoles de cette famille est l'OLSR (Optimised Link State Routing).

Les protocoles réactifs ne calculent une route que sur demande de transmission. Lorsqu'une machine-source souhaite envoyer un message à une machine-cible, elle envoie une requête à tout le réseau. Après réception de la requête, la machine-cible renvoie un paquet réponse qui va remonter vers la source et ainsi tracer le chemin pour l'échange des données. Dans ce type de protocole, la bande passante est économisée mais le délai d'établissement de la route est allongé. Le principal protocole réactif est l'AODV (Ad hoc On-demand Distance Vector).

Les protocoles MANET peuvent être modifiés afin d'assurer la fonction du routage dans la micro mobilité. Celle-ci étant considérée comme un cas simplifié de réseaux ad hoc (MANET) où l'ensemble des routeurs du réseau d'accès sont considérés comme étant des nœuds du réseau ad hoc mais qui sont fixes et les seuls nœuds mobiles sont les terminaux mobiles. La proposition d'une telle solution a été proposée ; il s'agit d'une extension du protocole TORA (Temporally-Ordered Routing Algorithm) nommé MER-TORA (TORA-based Mobile Enhanced Routing) [4].

TORA est basé sur l'utilisation de la propriété appelée "orientation destination" des graphes acycliques orientés (DAG : Directed Acyclic Graph). L'algorithme TORA consiste à établir un DAG dont la racine est la destination. Le nœud destination est le seul nœud sans arc sortant. Ainsi, à partir de chaque nœud, on peut retrouver la destination en suivant l'orientation du graphe (DAG). Chaque nœud a pour attribut une hauteur et le flot de données sera routé du point le plus haut au point le plus bas. Si le flot est bloqué à un nœud, la hauteur de ce dernier est augmentée de telle sorte qu'elle soit supérieure à celle de ses voisins et que les paquets de données puissent continuer leur route.

Quand un nœud a besoin d'une route vers une destination, il émet un paquet de requête contenant l'adresse de la destination. Ce paquet se propage dans le réseau jusqu'à arriver à un nœud voisin du nœud destination. Celui-ci émet alors un paquet de mises à jour contenant sa hauteur qui est égale à 1 -la hauteur de la destination étant 0-. Quand un nœud reçoit le paquet de mises à jour, il s'attribue une hauteur supérieure à celle contenue dans le paquet ; ce qui à pour effet d'orienter les arcs du graphe de façon à avoir une route vers le destinataire depuis tous les points du réseau. Quand un paquet de données doit être émis, il va suivre le graphe grâce à la différence entre sa hauteur et celle de ses voisins. Quand un nœud découvre que la route vers une destination n'est plus valide, il réajuste sa hauteur et transmet un paquet mis à jour pour actualiser le graphe. La connaissance des arcs descendant de la hauteur des voisins est obtenue par diffusion régulière de l'information de hauteur par les nœuds[60].

Le protocole TORA est originairement un protocole proactif mais, puisqu'il traite indépendamment le routage vers chaque destination, il est possible qu'il réagisse comme algorithme de routage proactif pour certaines destinations et comme algorithme de routage réactif pour d'autres[59].

Pour gérer la micro mobilité, le protocole MER-TORA ([59]) utilise l'algorithme TORA et distingue deux types de construction de routes : le routage Inter-AR (Access Router) et le routage spécifique au terminal (MN). Dans le routage Inter-AR, la route dans un domaine est continuellement maintenue ; dans ce cas MER-TORA réagit comme un protocole de routage proactif. Dans le routage spécifique au MN, la route est maintenue à la demande ; dans ce cas, le protocole réagit comme un protocole réactif. L'exécution du Handover dans MER-TORA exploite le coté réactif du protocole et le ré-établissement de la route vers le mobile dans le domaine exploite le coté proactif du protocole[59].

Dans un domaine chaque nœud connaît ses voisins selon le principe de MANET. Au moment du Handover, l'attachement du mobile au nouveau AR crée un tunnel de l'ancien AR vers le nouveau AR qui sont évidemment voisins. Les paquets reçus par l'ancien AR vont être redirigés vers le nouveau AR qui va les transmettre à son tour au mobile. Le mécanisme du protocole TORA fonctionne avec la réaction des liens entre les nœuds (lien rompu ou lien établi). De ce fait, à l'exécution du Handover, la détection de la rupture du lien de l'ancien AR vers le mobile déclenchera la mise à jours du DAG. Par conséquent, la route sera rétablie à partir du routeur « cross-over » qui relie l'ancienne route et la nouvelle route. Par la suite, le tunnel sera détruit quand le message de mise à jour de la route atteindra l'ancien AR. Ce processus se répète à chaque déplacement du mobile [59].

4.4.2.2. Architecture à base Multicast :

Pour qu'ils se communiquent dans Internet, les hôtes utilisent ce qu'on appelle les adresses unicast. Ce type d'adresse sert pour un routage point-à-point. L'adresse unicast a une seule destination possible. Un autre type de routage utilise les adresses multicast pour créer des connexions point-à-multipoint : un émetteur pour plusieurs destinataires. Les destinataires des paquets forment un groupe multicast.

L'architecture à base Multicast [4] exploite ce principe de routage pour la gestion de la micro mobilité en créant un groupe multicast regroupant l'ensemble ou seulement une partie des localisations des mobiles. On distingue deux modes dans cette architecture.

4.4.2.2.1. Mode dense (Daedalus) :

Ce protocole [4] s'inspire en partie de Mobile IP et est particulier pendant la procédure où le HA transmet les paquets vers la CoA d'un MN qui est dans un FN. Lorsque le mobile entre dans un FN, en plus de son adresse permanente, il obtient une adresse temporaire multicast qu'il va enregistrer auprès de son HA. Quand ce dernier intercepte les paquets destinés au MN, il les encapsule et les envoie vers tous les membres du groupe multicast. C'est les BS (ou les AP) qui forment le groupe multicast et non le MN lui-même.

La localisation courante du MN est déterminée de la manière suivante : Périodiquement, chaque BS diffuse un message balise (beacon) qui est reçu par tous les mobiles qui sont dans sa zone de couverture. En mesurant la force du signal reçu, le MN peut identifier les BS voisines proches et estimer vers quelle cellule va-t-il faire le Handover dans le futur. En se basant sur les informations fournies par les messages balise, le MN configure le routage multicast entre son HA et certaines BS. Il peut demander aux BS qui sont identifiées comme cibles probables pour le Handover de rejoindre son groupe multicast. On définit deux types de BS. La BS primaire est la BS de la cellule sur laquelle se trouve le MN. Les BS voisines qui appartiennent au groupe multicast sont des BS secondaires. A n'importe quel moment, il y a au plus une seule BS primaire. Chaque paquet envoyé par le HA et transmis par la BS primaire au MN. Pour les BS secondaires, même si elles reçoivent les paquets destinés au MN, elles ne font que sauvegarder les quelques derniers paquets reçus dans un buffer.

Le Handover est initialisé par le MN en envoyant un ensemble de messages de contrôle vers plusieurs BS. Ces messages demandent pour chaque BS de commencer à transmettre et à arrêter de stocker les paquets ou le contraire. Une liste des derniers paquets reçus par le MN est incluse dans les messages de contrôle qui activent la transmission dans la nouvelle BS primaire. Cette liste informe la nouvelle BS primaire sur les paquets déjà reçus par le MN. La nouvelle BS primaire commence à transmettre les paquets se trouvant stockés dans son buffer et qui ne sont pas encore transmis par l'ancienne BS vers le MN. Les paquets IP en transit envoyés par le CN sont délivrés directement au MN par l'intermédiaire de la nouvelle BS primaire sans que l'ancienne BS ne les envoie.

Ce protocole permet d'exécuter un Seamless Handover et d'éviter la mise à jour de la route au niveau du HA. Le point faible de cette approche est qu'elle ne permet pas le passage à l'échelle et la maintenance de table de routage est difficile. Ajoutée à cela la nécessité de mettre en place des routeurs Multicast à travers le backbone Internet puisque le passage en multicast se fait à partir du HA ;ce qui n'est pas possible actuellement.

4.4.2.2.2. Mode léger (spare mode) MMP :

Pour éviter de construire l'arbre de multicast à partir de HA, MMP (Multicast for Mobility Protocol [4]) combine le protocole de routage multicast CBT (Core Base Tree) avec Mobile IP. CBT construit un arbre de routage partagé dont le centre (core-point) se trouve à distances égales des destinataires. On choisit comme core-point la passerelle du domaine vers l'Internet.

Le MMP est transparent pour le MN et le HA puisqu'il n'est utilisé que pour acheminer les paquets IP à l'intérieur du FN.

Le core-point reprend les fonctionnalités du FA. Quand le MN entre dans un FN et obtient une adresse IP multicast, il envoie un message *Registration-Request* pour l'enregistrer. La demande est interceptée par le core-point pour modifier la CoA par son adresse et ensuite la transmettre au HA. A partir de là, le core-point qui a une table d'association adresse permanente et multicast CoA intercepte tous les paquets destiné au MN, les décapsule et les encapsule ensuite pour les envoyer au groupe multicast. Le mécanisme de gestion de la localisation et du Handover est le même que celui en mode dense, sauf que ce protocole gère le multicast à partir du core-point (FA).

5. Conclusion :

Dans ce chapitre, avant de décrire la gestion de la mobilité, nous avons présenté en détail la gestion de la QoS et les protocoles AAA. Dans la gestion de la QoS, nous avons vu les deux architectures fondamentales qui sont à l'origine de toutes les architectures de la QoS. La première est l'architecture IntServ qui se base sur le principe de réservation des ressources de réseau pour un utilisateur ou une application en utilisant le protocole de signalisation RSVP. La deuxième architecture est celle de DiffServ qui se base sur la différentiation de service ; chaque ensemble d'utilisateurs ou d'applications partage un service en fonction des exigences et des contraintes de ces applications. Cette architecture se voit comme une solution au problème de passage à l'échelle qui surgit dans l'architecture IntServ. Toutefois, chacune d'elles a ses propres avantages et ses propres inconvénients ; seulement, ce problème de passage à l'échelle freine le déploiement de l'architecture IntServ dans Internet. Une idée originale qui a été présentée dans ce chapitre est celle d'utiliser les deux architectures ensemble. La gestion des ressources dans un réseau offrant la QoS est problématique car les ressources sont limitées et le réseau est partagé par plusieurs utilisateurs. Pour cela le concept de la gestion de réseau par politique est introduit. Le protocole COPS qui est présenté dans ce chapitre conçu au départ spécialement pour la gestion par politique des réseaux IntServ s'avère efficace dans d'autre architecture tel DiffServ grâce à son caractère générique.

Dans les architectures AAA, nous avons exposé les deux protocoles les plus connus qui sont RADIUS et Diameter. On peut voir Diameter comme une amélioration de RADIUS du fait qu'il peut prendre en considération des applications spécifique en exploitant le protocole de base Diameter ; l'une de ces applications est la mobilité. Le protocole COPS pourrait satisfaire aussi parfaitement les fonctionnalités de l'architecture AAA [57] seulement la particularité de COPS est qu'il est orienté vers la gestion des QoS. Nous avons vu aussi le protocole d'authentification EAP qui permet d'utiliser n'importe quelle méthode d'authentification même celles qui seront proposées dans le futur ; ce qui le place en tant que meilleur candidat des protocoles d'authentification. Nous avons expliqué comment le protocole IEEE 802.1X permet d'autoriser l'accès physique à un réseau filaire ou sans fil en utilisant le protocole EAP et un nouveau mécanisme qui lui est propre.

Enfin dans une dernière partie, nous avons présenté la gestion de mobilité. Le protocole Mobile IP est conçu pour gérer la macro mobilité qui concerne le déplacement de l'utilisateur mobile d'un domaine à un autre. Nous avons conclu que ce protocole n'est pas performant quand l'utilisateur mobile se déplace au sein d'un même domaine à cause de la quantité d'informations qui doit être transportées dans le réseau pour la gestion de la localisation et du Handover ; ce qui mène à une gestion plus localisée qui est la gestion de la micro-mobilité. Nous avons vu que les protocoles de la gestion de la micro-mobilité visent principalement à réduire la signalisation et à optimiser le Handover. Un autre protocole qui peut être utilisé dans la gestion de la mobilité est le protocole HIP. Ce protocole, très ambitieux, change l'architecture de la pile TCP/IP. Il introduit un espace de nommage pour les identificateurs

afin, principalement, de séparer l'identification des terminaux, utilisés par les applications, de leurs adresses IP utilisées que pour le routage. Cela représente un grand avantage pour la gestion de la mobilité et nous avons expliqué comment la mobilité exploite ce mécanisme.

L'exploitation du concept de la mobilité devient indispensable dans le monde de la communication. L'utilisateur d'Internet devient de plus en plus mobile et les opérateurs espèrent avoir la possibilité de mettre en œuvre une architecture idéale afin de conquérir cette catégorie de clientèle. En outre, l'intégration de la QoS dans cette architecture est l'élément clé visant à satisfaire le client en lui garantissant un service meilleur et adéquat aux applications de l'utilisateur mais la gestion des ressources dans un réseau de QoS introduit, surtout à cause des contrôles effectués, une charge qui peut influencer la gestion de la mobilité et particulièrement la gestion du Handover. Le service devient par conséquent dégradé contrairement aux attentes du concept de la QoS. De plus, la sécurité dans les réseaux mobiles sans fil est plus difficile que dans les réseaux fixes. Dans les réseaux fixes, l'utilisateur ne s'authentifie que dans le début de la session et qu'auprès du serveur AAA de son fournisseur. Il ne demande l'autorisation d'utilisation du service et l'établissement de la QoS que dans le début de cette session. Dans les réseaux sans fil et mobiles ces fonctions sont indispensables pour chaque changement de point d'attachement dans la même session. En plus des problèmes rencontrés dans la mobilité pour assurer un Handover optimal, l'introduction de la QoS et l'assurance des fonctions AAA ne fait évidemment que rendre ces problèmes plus difficiles à solutionner et à optimiser. A l'issue de l'étude de ces trois architectures, on constate que ces protocoles peuvent générer une grande quantité de signalisation s'ils sont utilisés en même temps que le Handover ; ce qui motive à penser qu'une signalisation commune à la QoS, à la AAA et à la Mobilité est vraiment indispensable. Aussi, proposons-nous dans le chapitre suivant une analyse sur l'interaction entre ces architectures pour assurer la mobilité.

Chapitre III

Interaction entre la QoS, la AAA et la Mobilité

1. Introduction :

L'enjeu économique que l'Internet mobile peut représenter dans notre société a poussé la communauté de Télécom à offrir l'accès à Internet par l'intermédiaire de son réseau sans fil. Les systèmes, tels les systèmes GPRS (General Paket Radio Service) et UMTS (Universal Mobile Telecommunication System), sont le fruit des efforts de cette communauté. Ces systèmes sont capables de transporter des paquets IP en utilisant un réseau à commutation de paquets en parallèle avec le réseau téléphonique à commutation de circuits. Ces architectures qui utilisent des protocoles propriétaires sont soumises à des lois de licence strictes en matière d'utilisation et de prix -ce qui contribue à freiner leur évolution- et le service d'accès à Internet est indépendant des réseaux conçus spécialement pour la téléphonie mobile.

Les protocoles de l'Internet définis par l'IETF ont été conçus originairement pour des réseaux fixes. Utilisés dans des environnements sans fil ou mobiles, ils voient leurs performances ou, tout au moins, leurs comportements très affectés. Avec l'adoption rapide de la communication à base d'IP pour l'informatique mobile, l'utilisateur espère recevoir des services par les réseaux sans fil similaires à ceux qui sont fournis par les réseaux filaires. Les utilisateurs s'attendent à un accès flexible aux services Internet qui ne se limitent pas seulement aux services de données mais aussi aux services multimédias et téléphoniques. Dans les réseaux à commutation de paquets tels les réseaux IP, la solution ne consistera qu'à l'utilisation du concept de la QoS et, en dépit de la technologie du réseau d'accès ou de la mobilité du terminal, la QoS offerte doit être garantie. Cela pose de nouveaux problèmes et de nouveaux défis pour la délivrance de la QoS dans les réseaux mobiles sans fil puisque celle-ci doit prendre en considération la mobilité des terminaux. Plusieurs architectures de QoS ont été définies dans le contexte des réseaux mobiles mais aucune n'est considérée comme étant la solution complète pour fournir la QoS dans ce type de réseau.

La signalisation AAA, bien qu'indispensable, est, elle aussi, problématique dans les réseaux sans fil mobiles. Le mobile aura un seul fournisseur d'accès dans lequel il y aura toutes les informations concernant l'authentification, l'autorisation et la comptabilité de l'utilisateur. Lors de la connexion des mobiles aux réseaux d'accès visités, la procédure AAA n'implique pas seulement le réseau visité mais aussi le réseau d'origine et cela créera une grande latence. Or pour garantir la QoS, cette procédure doit être très rapide à cause principalement de la mobilité qui rend les déconnexions/connexions fréquentes. Le délai de

l'interruption et la charge de signalisation causés par ces opérations ne pourraient pas être négligeable car ils sont liés à la fréquence du Handover dans la micro et la macro mobilité. En outre, dans ce type de réseau, la sécurité est plus délicate que dans les réseaux fixes ; ce qui rend sa gestion complexe dans un réseau qui exige une très grande souplesse dans tous les types d'opération.

Ce chapitre va identifier et analyser les principaux problèmes qui surgissent quand on exploite la QoS dans la mobilité IP et quelques solutions seront ensuite exposées. Cette analyse globale se focalise particulièrement sur les architectures Mobile IP, DiffServ et IntServ. Nous analyserons aussi comment la gestion de AAA est organisée dans les réseaux sans fil mobile et la solution adoptée par le protocole Diameter par la suite.

2. Interopérabilité entre La QoS et La Mobilité :

L'interaction entre la gestion de la QoS et la mobilité dans les réseaux IP est assez problématique car ces deux gestions ont évoluées séparément l'une de l'autre. Les études menées dans le domaine de la gestion et du maintien de QoS ont concerné plus les réseaux filaires. Par conséquent, quand on tente d'utiliser la QoS dans les réseaux sans fil mobiles, les résultats seront différents que ceux qu'on peut avoir dans les réseaux fixes. Cette partie est consacrée, en premier lieu, à l'analyse du comportement de la QoS offerte quand elle est appliquée sur les réseaux mobiles.

2.1. L'impact de la mobilité sur la QoS :

Les problèmes et les obstacles majeurs auxquels se heurte la gestion de la QoS quand celle-ci est appliquée dans les réseaux sans fil mobiles, peuvent être analysés sur les trois volets suivants : la gestion du Handover, qui est elle-même liée à la topologie du réseau, la macro-mobilité et à la micro-mobilité.

2.1.1. L'impact du Handover :

Si on prend en considération la QoS telle qu'elle est perçue par l'utilisateur, l'interruption ou la perturbation de la communication ou de la réception de données causée par le Handover se traduit par une dégradation du niveau QoS même si l'utilisateur ne recevait que le service Best-Effort. La résolution de ce problème relève aussi bien de l'analyse topologique que de l'analyse technique. Pour pallier cet inconvénient, la topologie du réseau doit être conçue de manière à ce que le Handover soit le moins fréquent possible quel que soit le mouvement du MN. Le maintient du niveau de la QoS revient également à réduire au maximum le délai du Handover et la perte de paquets due au Handover. Ce qui veut dire que l'exécution d'un Seamless Handover est la situation la plus adéquate pour la stabilité du niveau du service.

Un autre problème pour la gestion de la QoS sur la mobilité est, en dépit du mouvement du mobile dans les réseaux d'accès sans fil et des schémas du routage pour la mobilité, l'approvisionnement du service demandé qui doit être assuré quand le mobile change son point d'attachement au réseau d'accès. Le Handover entre les points d'accès, le changement d'adresse IP ainsi que les mécanismes associés à la micro mobilité peuvent créer des situations où le service assuré pour le mobile ne pourrait pas être correctement fourni ; ce qui induit la violation du niveau de service assuré. La violation de la QoS peut aussi survenir même à cause d'un simple refus du service par le nouveau point d'attachement. Dans le cas où le mobile demande seulement une priorité relative pour ses flots, une violation assez courte de la QoS serait acceptable à une certaine limite mais si ce mobile réserve explicitement des ressources, le nouveau point d'accès et la route du flot jusqu'au bord du domaine doivent fournir les mêmes ressources ; ce qui ne peut pas être toujours possible et tout à fait garanti.

Dans l'exécution du Handover et dans le cas de l'architecture IntServ, RSVP aura des problèmes pour garantir les réservations de ressources parce que la phase de réservation, celle de routage et celle de transmission de données sont indépendantes. Les nœuds RSVP ont besoin d'envoyer périodiquement les messages *Path* et *Resv* pour chaque flot pour le rafraîchissement de la réservation de bout en bout. Quand la route vers le mobile change, les paquets vont seulement recevoir le service BE jusqu'à ce que l'état de la réservation soit mise à jour dans la nouvelle route par les messages périodiques *Path* et *Resv* du protocole RSVP. Ce problème sera de plus en plus important si le Handover est profond. Il causera un temps de réparation de la QoS plus long et par conséquent une dégradation du niveau de service. Afin de réduire la période où il n'y a pas de réservation, le message de rafraîchissement devra être envoyé immédiatement après le Handover [11].

Faute de manque de mécanisme de signalisation dans l'architecture DiffServ et sans un gestionnaire de bande passante (Bandwidth Broker) pour co-ordonner et réguler le partage de ressources, l'arrivée d'un nouveau mobile dans une cellule peut perturber le partage de ressources entre les mobiles existants dans cette cellule. De ce fait, les mobiles se trouvant dans une cellule chargée verront leurs niveaux de service se dégrader au fur et à mesure que les mobiles arrivent dans cette cellule[11].

2.1.2. L'impact de la profondeur du Handover :

On a vu dans le chapitre précédent (Chapitre II4.2.1) que selon les points d'accès entre lesquels le mobile se déplace, le type de Handover sera différent et le taux de signalisations relatives à la gestion de la mobilité dans le réseau d'accès le sera aussi. Ainsi plus le Handover est profond dans le réseau d'accès plus le temps sera long pour rétablir la QoS de bout en bout avec la nouvelle localisation du nœud mobile. Le niveau le plus haut du Handover se propage dans le réseau d'accès et demande plus de temps pour installer la route vers le mobile et pour mettre en place la QoS.

Dans le Handover Intra-AR, le contrôle du Handover consiste seulement à gérer les ressources radio puisque le flot du mobile qui déclenche le Handover ne change pas de route entre le AR et ANG et la phase du contrôle d'admission pourrait être omise si elle est déjà faite avec le AR quand le nœud mobile initie le Handover. Ce Handover est généralement transparent à la couche IP si l'interface au mobile n'a pas était changée. La dégradation du niveau de service due au Handover pourrait être négligeable.

Si, dans un Handover Inter-AR, le AR change sans le changement de la ANG, le Handover affecte la disponibilité de ressources radio et du réseau d'accès. Le nouveau AR pourrait avoir besoin d'exécuter le contrôle d'accès et, peut être aussi, le contrôle d'admission au même moment. La coordination des ressources est plus affectée quand le Handover concerne le changement de ANG. La ANG peut être changée quand le MN se déplace dans un grand réseau d'accès qui contient plus d'une passerelle ou quand le MN se déplace vers un autre réseau d'accès. Dans les deux cas, une nouvelle adresse IP est attribuée au mobile ; ce qui implique le changement de route qui peut compromettre la garantie de la QoS. Quand la ANG change, les flots RSVP subiront une plus longue dégradation de QoS à cause du temps nécessaire aux messages de rafraîchissement pour atteindre un routeur qui a un état de réservation et peut donc initialiser une mise à jour de la réservation sur la route.

Le Handover le plus complexe se réalise quand le MN change de domaine administratif. En plus de l'établissement de la nouvelle route et de la gestion de l'allocation de la QoS, le MN aura besoin de se ré-authentifier et d'être autorisé à utiliser les ressources du réseau d'accès et les enregistrements de comptabilité doivent être aussi initialisés ; le mobile sera forcement en communication avec son réseau mère qui est peut être très loin. Tout cela

provoquera une surcharge en trafic de signalisation qui va affecter le délai de rétablissement des ressources pour le mobile[11].

2.1.3. L'impact de la Macro Mobilité :

C'est dans la macro mobilité que le Handover le plus profond est impliqué. La macro mobilité nécessite un délai de rétablissement des routes plus important que dans le cas d'une micro-mobilité. En plus des opérations de la gestion de la localisation ou de re-routage, les opérations de la re-authentification, de l'autorisation d'utiliser les ressources et/ou des services et même d'initialisation des enregistrements de comptabilité sont toujours indispensables. Ces procédures requièrent un certain temps qui doit s'écouler avant que la procédure du rétablissement et de la reconfiguration de la QoS sur le nouveau chemin commence laquelle, à son tour, nécessite un certain temps pour s'exécuter. Tout cela, entraîne une violation de la QoS très longue et non négligeable.

Une incommodité particulière pour la gestion de la QoS concerne la gestion des adresses. Puisque Mobile IP exige au MN le changement d'adresse (FA-CoA ou Co-CoA) à chaque fois qu'il change de sous-réseau, la réservation de ressources nécessite la réfection de toute l'opération sur tout le chemin de bout en bout. Ce processus introduit un temps d'attente de rétablissement de la QoS important affectant ainsi la garantie de QoS. Si le RSVP est utilisé pour faire une réservation de ressources pour un niveau de service, une nouvelle réservation doit être installée tout au long de la route du flot de données à chaque changement de CoA. Même en utilisant le schéma de la route optimisé dans mobile IP, la transmission directe de la mise à jour des associations aux CN cause une grande latence avant de commencer la mise à jour de la localisation ; dans ce cas, l'interruption de communication pendant le Handover sera longue. Cette latence augmente si le MN et le HA ou le CN sont séparés par un nombre assez grand de sauts dans le réseau.

Dans Mobile IP, les mises à jour de la localisation ont besoin d'être transportées du MN au HA/CN avant que le changement de la localisation du mobile ne soit effectivement communiqué et que les connections courantes ne soient restaurées. Comme on l'a déjà vu, sans un mécanisme adapté, les données en transit seront perdues jusqu'à l'accomplissement du Handover et, par conséquent, jusqu'à l'installation de la nouvelle route vers le MN. Toutefois, le schéma de la route optimisée peut assurer un Smooth Handover en utilisant l'ancien FA et le HA dans certains cas pour éviter la perte de ces données et diminuer la dégradation du niveau de service.

La mise à jour de la localisation est toujours générée quand le MN change de sous-réseau dans le FN. La hausse de la mobilité de l'utilisateur peut générer des notifications trop fréquentes au HA ; ce qui va engendrer une surcharge du réseau en trafic de signalisation. Dans le cas où la population d'utilisateurs mobiles serait large, la charge de la signalisation devient une portion significative du trafic et par conséquent elle affecte la réservation de ressources.

Un autre problème est engendré directement par le routage triangulaire. Plusieurs architectures de la QoS opèrent mieux quand les paquets suivent le même chemin pour les deux directions de transmission. Pour ces architectures, le routage triangulaire affecte la QoS offerte. Afin d'éviter ce problème, il est possible d'utiliser un tunnel qui a comme entrée le MN ou le FA du MN et comme sortie le HA du MN dans le sens contraire du tunnel descendant vers le MN : c'est ce qu'on appelle le tunnel inverse "Reverse Tunneling". De cette façon, le routage triangulaire est évité mais un autre problème surgit : les routeurs qui se trouvent dans le tunnel ne pourront pas reconnaître les paramètres de la QoS. Par exemple, quand les messages RSVP sont encapsulés dans un en-tête IP, l'option « Router Alert » devient invisible au routeurs RSVP de la route des messages si elle est utilisée par les paquets

RSVP pour indiquer aux routeurs que les paquets nécessitent un traitement particulier. Même si des solutions qui résolvent ce problème ont été proposées tel l'extension de RSVP pour la mobilité, ces solutions augmentent la complexité des protocoles de QoS pour opérer dans un environnement mobile [13].

Dans le cas de Mobile IPv6, le problème tel que le routage triangulaire ou de Tunneling sont résolus grâce à la « Route Optimization » pour le premier et au « Routing Header » pour le second. Ce dernier diminue la surcharge du réseau grâce à l'élimination de l'encapsulation IP. De plus, un mécanisme de détection de mouvement est intégré pour permettre ainsi une meilleure performance durant le Handover. En effet plus tôt le réseau d'accès détecte que le mobile va changer de cellule, plus tôt il déclenche la procédure Mobile IP et va minimiser la durée du Handover. Malgré la volonté de Mobile IPv6 d'atteindre les objectifs de transparence et de support de Handover, cependant, il reste non optimisé pour offrir une gestion optimisée d'un Seamless Handover dans un réseau supportant des applications demandant la QoS à cause du nombre important d'enregistrement de la localisation et aussi de la latence de rétablissement des routes après le Handover[13].

2.1.4. L'impact de la Micro Mobilité :

Les schémas de la micro mobilité peuvent utiliser les différents mécanismes du Tunnelling, du multicast ou du routage spécifique. Dans ces architectures, le mouvement local du mobile affecte différemment ces différents types d'architecture de la QoS. Dans l'architecture IntServ, chaque routeur garde un état par flot, de ce fait le mouvement du mobile va déclencher des réparations localisées de la route ainsi que de la réservation de ressources dans le réseau. D'un autre coté, DiffServ n'a pas de mécanisme de signalisation ; ce qui signifie qu'il n'y a pas d'états à mettre à jour dans le réseau mais le niveau de service fourni va varier en fonction du nombre de mobiles arrivant ou partant dans une cellule. Des améliorations ont été proposées pour mieux supporter le changement de l'adresse IP et supporter aussi l'encapsulation IP dans IntServ et dans DiffServ[13].

L'approche Multicast peut avoir un effet négatif sur la disponibilité des ressources, par exemple, parce que le groupe du multicast peut varier dynamiquement et de ce fait les ressources seront épuisées presque inutilement. D'un autre coté, l'utilisation du tunnel peut affecter la transmission des flots recevant une QoS puisque le paquet original est encapsulé dans un autre paquet IP. Toutefois, tant que les terminaisons du tunnel sont capables de fournir des ressources pour le trafic qui traverse le tunnel, la garantie de la QoS peut ne pas être voilée. Le tunnel peut avoir par contre l'avantage d'agréger plusieurs flots pour une seule réservation de ressources. Enfin, la micro mobilité qui se base sur Per-Host Forwarding ne simplifie pas le support du routage à base de la QoS puisqu'il y a seulement une seule route possible pas saut.

Rappelons que le but principal de la gestion de la micro mobilité est de gérer le mouvement du mobile localement afin de minimiser l'interruption due au Handover et la charge causée par le trafic de la signalisation. Certaines architectures assurent l'exécution d'un Seamless Handover ; ce qui est optimal pour la gestion de la QoS. Cependant, ces architectures utilisent des protocoles et des infrastructures spécifiques qui ne tiennent pas compte des paramètres utilisés dans les architectures de la QoS telle que IntServ et DiffServ. Ainsi, les solutions de la gestion de la QoS dans la micro mobilité seront nécessairement spécifiques aux architectures de la micro mobilité. Dans l'ensemble, une conception d'une architecture qui combine la micro-mobilité et la QoS doit au moins tenir compte des points suivants [11] :

- L'utilisation de l'encapsulation IP ou de tout autre tunnel particulier cache les informations de l'en-tête du paquet original ; ce qui va affecter la classification de celui-ci.

- Le changement de l'adresse CoA durant la session va affecter le temps de rétablissement du chemin et des ressources nécessaires.
- Le Multicast des paquets à plusieurs routeurs va consommer beaucoup de ressources réseau.
- La route fixe vers le réseau extérieur à travers la même passerelle ne permet pas le passage à l'échelle.
- L'adaptabilité au changement des routes est nécessaire.
- Avoir une route optimale de la passerelle vers le routeur d'accès réduit le temps d'exécution des procédures relatives à la QoS.
- Le support du routage à base de QoS est une très bonne vision pour la combinaison de la QoS et de la micro mobilité.

2.2. Concepts résolutoires :

Ci après, quelques solutions préconisées par l'IETF et par la communauté académique sont présentées ; l'objectif final étant de combiner la gestion de la macro mobilité, la micro mobilité et la QoS pour une garantie de service de bout en bout dans les réseaux mobiles. Par fois ces solutions sont applicables à une architecture QoS spécifique.

2.2.1. Contrôle d'admission et priorité :

Dans une architecture comme DiffServ, il est possible de mettre en place un mécanisme de contrôle d'admission qui va limiter le trafic Best Effort pour ne pas utiliser toutes les ressources restantes et prévoir aussi un certain taux de ressources pour pouvoir supporter le trafic des nœuds mobiles arrivant (mobile en Handover) [11]. Pour réaliser cela, il n'est pas nécessaire d'installer de nouveaux dispositifs ; il suffit juste d'appliquer des règles de contrôle d'admission au niveau des routeurs d'accès au réseau pour limiter le trafic dans chaque classe de service.

Un mécanisme de priorité est aussi utilisé pour admettre en priorité le trafic des mobiles en Handover que le trafic des nœuds qui commence une nouvelle session dans la cellule. Cette technique est appliquée dans les réseaux cellulaires de télécommunication. Le problème est de connaître le taux de bande passante à réserver pour le trafic des mobiles qui demanderont la connexion et pour le trafic des mobiles qui exécuteront le Handover. Il existe quatre stratégies de réservation [11] : la première est la stratégie fixe qui réserve un pourcentage fixe de capacité de ressources du routeur d'accès pour les connexions du Handover, la deuxième est la stratégie statique qui se base sur les connexions effectives des mobiles et qui peut utiliser l'historique de leurs trafics, la troisième est la stratégie dynamique qui réserve dynamiquement les ressources du routeur d'accès pour les connexions du Handover en se basant sur le nombre de mobiles connectés dans les cellules voisines et la dernière est la stratégie dynamique avancée qui utilise un modèle analytique dont lequel les demandes du Handover peuvent différer en quantité nécessaire de ressources.

2.2.2. Réservation de ressource par rapport au Handover :

L'approche la plus simple pour intégrer la QoS dans les réseaux sans fil mobile est de faire exécuter la gestion de la QoS et la gestion de la mobilité séparément et indépendamment l'une de l'autre ; l'implémentation de la QoS ne dépend d'aucune architecture ou mécanisme de la mobilité. Les réservations de la QoS sont installées en utilisant la signalisation RSVP et les paramètres de contrôle IntServ. Quant à l'information de routage, elle est distribuée en utilisant un protocole de la micro-mobilité standard ou spéciale ou de la macro-mobilité. En effet, lors du Handover la procédure de gestion de mobilité sera exécutée et à la fin de celle-ci, la procédure de gestion de QoS va se mettre en place pour rétablir les ressources sur le

nouveau chemin établi par la procédure de gestion de mobilité. Quand on utilise IntServ avec le protocole RSVP, le changement de la route du trafic est traité par le caractère soft-state de l'architecture et la réparation de la réservation au long du chemin ou portion du chemin selon la profondeur du Handover est accomplie par les messages de rafraîchissement périodiques.

Le problème avec cette manière de procéder est que le niveau de service risque de connaître une dégradation durant tout le temps nécessaire à RSVP pour rétablir la QoS sur le nouveau chemin puisque l'installation le long de la nouvelle route n'est pas systématiquement immédiate. Si le message de rafraîchissement est généré avant que la nouvelle route du trafic du mobile soit complètement établie, il risque d'avoir des réservations de ressources sur une route qui ne correspond pas à celle du mobile, et celle-ci ne sera malheureusement ni corrigée ni effacée avant le prochain message de rafraîchissement ; cela n'implique pas vraiment une utilisation efficace des ressources du réseau. En outre, la réservation, dans certains cas, peut être rejetée par manque de ressources sur l'un des routeurs sur la nouvelle route du mobile. Enfin, cette perturbation au niveau de la réservation des ressources risque d'arriver aussi souvent que le nœud mobile change de routeur d'accès et cela au cours d'une même session RSVP ; ce qui va induire une QoS globale très pauvre pour l'application s'exécutant sur le mobile.

Afin d'améliorer le comportement de la QoS à base de réservation dans un environnement de mobilité, les mécanismes de la QoS et de la micro-mobilité peuvent être couplés pour s'assurer que les réservations sont installées dès que possible. Ces mécanismes peuvent être réalisés de différentes manières selon que la réservation de ressource a lieu durant le Handover, en même temps que le Handover ou avant le Handover.

2.2.2.1. Réservation durant le Handover :

Dans cette approche, les événements générés par la procédure de gestion de mobilité vont déclencher le processus de réservation de ressources de la procédure de gestion de QoS. Le mécanisme de réservation RSVP sera déclenché dès que la nouvelle route est installée. Avec IntServ, il s'agit du mécanisme "Local Path Repair" défini dans les spécifications de RSVP[15]. Cette option a l'avantage de minimiser la durée de perturbation de service entre le début du Handover jusqu'au rétablissement des ressources dans le nouveau chemin. Elle évite aussi d'effectuer des réservations de ressource dans le réseau avant que l'information concernant la nouvelle route du mobile ne soit disponible. Au lieu d'attendre que la mise à jour de la réservation soit transportée de bout-en-bout ou que le nœud correspondant génère un message de rafraîchissement de la réservation vers le mobile, le délai d'établissement de la nouvelle route peut être minimisé en localisant la réparation de la réservation à la portion du chemin qui a changé ; c'est-à-dire entre le routeur crossover et le routeur d'accès auquel le mobile est nouvellement attaché. Toutefois, dans le cas où la QoS devrait être ré-négociée, la signalisation de bout-en-bout est inévitable. De plus, les ressources réservées sur l'ancienne route doivent être libérées explicitement.

L'un des problèmes de cette approche est la complexité introduite dans le réseau pour intercepter les messages RSVP pour limiter la réparation à la région entre le routeur crossover et le mobile et le fait que le mécanisme RSVP soit déclenché par des événements liés à la mobilité générera une signalisation plus importante comparée au cas de la réservation de ressource indépendamment de la mobilité où la procédure RSVP est déclenchée uniquement par les messages de rafraîchissement d'une manière régulière. Dans le cas de gestion de mobilité à base d'agent Proxy, l'événement déclencheur de la procédure RSVP sera la réception de l'accusé de réception de l'enregistrement de la nouvelle localisation. Dans le cas de gestion de mobilité basée sur Manet, la réception du message de mise à jour de la route sera en revanche le déclencheur de la procédure RSVP. Enfin dans la gestion à base de

modification de route localisée, dès que l'information de routage est distribuée dans tout le réseau, la procédure RSVP sera déclenchée [11].

2.2.2.2. Réservation en même temps que le Handover :

Cette approche suggère l'utilisation du même mécanisme de signalisation pour propager les informations de la mobilité et de la QoS soit par l'extension du protocole de signalisation de la QoS soit via un protocole de routage à base de QoS unique. Cette approche minimise l'interruption du trafic après le Handover en assurant que la réservation des ressources est mise en place dès que c'est possible après le Handover. Toutefois, au lieu d'attendre l'acquittement que la route vers le MN soit mise en place, comme dans le cas avec l'approche présentée plus haut, les ressources nécessaires pour le niveau de service du trafic du mobile sont installées en même temps que les informations de routage ; cela permet d'éviter le problème d'installation d'une réservation avant la propagation vers le MN des information de routage valides et de fournir aussi le moyen d'installer plusieurs réservation en utilisant un seul message de signalisation. Par conséquent, le taux de signalisation de la QoS est considérablement réduit. A moins qu'une renégociation de la QoS de bout-en-bout ne soit nécessaire, la mise à jour de la réservation peut, comme avec l'approche précédente, être localisée pour la zone affectée par le changement de la route. La réservation de ressources au long de l'ancien chemin doit être aussi explicitement effacée.

Le problème majeur de cette approche est d'abord la complexité de traitement introduit dans les nœuds du réseau et le besoin de transporter aussi les informations de QoS par les mécanismes de gestion de la micro mobilité. Dans certains cas, de nouveaux messages sont nécessaires pour assurer cette fonction ou encore un nouveau protocole unifiant la signalisation de la QoS et de la mobilité doit d'être créé pour supporter cette solution.

Dans les approches par agent proxy, les paramètres de QoS seront transportés dans le message d'enregistrement de localisation. Dans l'approche par Manet, l'information de QoS sera transportée dans les messages de la mise à jour de la route. Dans l'approche par modification de routage, les informations de QoS seront transportées par les messages de "path-set-up" et de "refresh" [11].

2.2.2.3. Réservation avant le Handover :

Pour aboutir à une garantie de service indépendante de la mobilité, les techniques présentées précédemment tel que le contrôle d'accès, la priorité ou l'intégration de la signalisation de QoS dans la signalisation de mobilité ne suffisent pas. Le nœud mobile a désormais besoin d'effectuer une réservation de ressources au niveau des cellules qu'il peut visiter durant son déplacement. Ci après deux approches sont présentées, une s'applique à l'architecture IntServ (MRSVP) et l'autre s'applique à l'approche DiffServ (ITSUMO).

2.2.2.3.1. MRSVP :

Mobile RSVP ([63]) est un protocole de réservation de ressources à l'avance dans un réseau mobile supportant l'architecture IntServ. Il propose trois niveaux de services pour les utilisateurs mobiles. Chacun de ces niveaux offre une certaine garantie de latence pour les flots du mobile et chacun est appliqué dans le domaine tant que ces flots restent conformes aux paramètres du trafic. Le premier de ces services est le service MIG (Mobility Independent Guarantees) dans lequel le mobile reçoit un service garanti, le deuxième est le service MIP (Mobility Independent Predictive) dans lequel le service reçu est prédictif et le troisième est le service MDP (Mobility Dependent Predictive) dans lequel le service reçu est prédictif avec une grande probabilité mais la qualité du service peut se dégrader dans des circonstances de grande charge du réseau.

Dans le modèle MRSVP un mobile peut faire des réservations de ressource le long du chemin de son flot, dans le sens descendant et ascendant en joignant les localisations qui sont susceptibles d'être visitées par ce mobile durant sa connexion. Cet ensemble de localisations est spécifiées par le MSPEC. Idéalement, le MSPEC contiendra toutes et seulement toutes les localisations que le mobile va visiter pendant sa connexion. La détermination de cet ensemble de localisations est encore un problème non résolu bien que plusieurs mécanismes soient proposés pour le déterminer d'une manière approximative par le réseau. Dans certaines situations, un mobile peut spécifier son propre MSPEC comme partie de son profil de mobilité. Dans tous les cas, quand le mobile initialise sa réservation, il est supposé acquérir son MSPEC grâce au réseau ou à son profil de mobilité. Avec MRSVP, le MSPEC d'un mobile peut changer dynamiquement quand le mobile est en connexion. Dans un tel cas, des ressources vont être réservées sur les nouvelles localisations ajoutées au MSPEC seulement s'il en a suffisamment sur le chemin du flot joignant ces localisations dans les deux sens.

Deux types de réservations sont supportés dans MRSVP : la réservation passive et la réservation active. Un émetteur ou récepteur mobile fait une réserve active sur sa localisation courante et des réservations passives sur les localisations différentes que la courante figurant dans son MSPEC. La localisation courante est la cellule du routeur auquel le mobile est actuellement connecté pour communiquer à travers le réseau d'accès. Sur la route, les réservations passives et la réservation active sont fusionnées au point de leurs rencontres. Toutefois, une réservation passive ou active d'un même flot peut être désinstallée sans affecter d'autres réservations. Pour une utilisation judicieuse des ressources réseaux, les ressources des réservations passives d'un flot peuvent être utilisées par d'autres flots qui exigent une garantie de QoS plus faible ou par les services Best-Effort. Cependant, quand une réservation passive devient active – quand le flot d'un mobile qui a fait une réservation passive se déplace vers ce lien – ces flots peuvent être affectés.

Dans MRSVP, un paquet unicast est délivré à un mobile en utilisant le protocole de routage Mobile IP. Dans un tel cas, les réservations de ressources pour un mobile sont installées tout au long de la route qui est déterminée par Mobile IP. Cela implique que, quand le mobile est localisé dans un FN et les paquets unicast sont délivrés à ce mobile via le HA par le tunnel IP, les réservations de ressources doivent être aussi installées sur le tunnel (les routeurs sur le tunnel sont des routeur RSVP).

2.2.2.3.2. ITSUMO :

L'architecture ITSUMO (Internet Technologie Supporting Universal Mobile Operation, [62]) a une autre philosophie que celle du modèle précédent. Elle est basée sur un serveur de QoS global (QGS, QoS Global Server) et des serveurs de QoS locaux (QLN, QoS Local Node) installés sur les routeurs. La réservation des ressources n'est pas effectuée par le mobile mais par le serveur QGS. Le mobile, à son arrivée dans un domaine interagit avec le QGS en négociant un niveau de service (SLS) de manière dynamique et en spécifiant sa demande de réservation et son profil de mobilité. En se basant sur des informations internes et sur le SLS, le QGS prévoit la quantité de bande passante devrait être réservée sur chaque QLN. Il envoie cette configuration aux QLN et, par la suite, met périodiquement à jour les QLN qui seront probablement visités par ce mobile.

Dans ce modèle, il semblerait que l'architecture serait plus efficace, sur le plan de l'utilisation de ressources si le principe de la réservation passive (utilisé par le Best-Effort) est utilisé. Le protocole utilisé pour mettre en place cette signalisation de négociation de SLS et de demande de réservation est DSNP (Dynamic Service Negociation Protocol). La principale différence avec l'approche de MRSVP est qu'avec celle-ci c'est au mobile de signaler la réservation d'une manière explicite en s'accordant à son profil de mobilité. Ce profil est connu et calculé par le mobile. Dans l'approche ITSUMO, ces informations sont mises à jour

périodiquement par le QGS, en s'accordant par le profil de mobilité fourni par le mobile mais calculé par le QGS[58].

2.2.3. Transfert de contexte :

La réservation de ressources dans les réseaux sans fil mobiles a pour objectif de garantir le niveau de service aux utilisateurs mobiles. Cependant elle induit aisément une utilisation non optimisée des ressources réseaux. Pour contourner le schéma de réservation à l'avance qui exige beaucoup de ressources, l'événement du Handover et le changement associé à la nouvelle cellule doivent être liés à la disponibilité effective des ressources dans la nouvelle cellule. A cette fin, quand le réseau ou le mobile juge que le Handover est imminent, le mobile doit diffuser une requête de ressources aux routeurs d'accès voisins et demande quelques indications sur la disponibilité de ressources. Pour cela une architecture spécifique est définie ([61]). Dans cette architecture, chaque routeur d'accès maintient un contexte du nœud mobile ; dans le contexte sont définis les caractéristiques du mobile tel la QoS ou la sécurité liée au traitement du trafic du mobile au niveau du routeur d'accès auquel il est connecté. Lors du déplacement d'une cellule à une autre, ce contexte sera transféré, en utilisant le protocole CXTP "context protocol" ([61]) vers la nouvelle cellule pendant le Handover pour justement, préparer la QoS dans la nouvelle cellule. Aucune autre signalisation n'est nécessaire pour mettre en place la QoS puisque le traitement correspondant au trafic du mobile est déjà en place au niveau de la nouvelle cellule grâce au transfert de contexte du mobile.

Le transfert de contexte durant le Handover semble être une bonne alternative pour résoudre le problème de coordination de ressource dans les réseaux mobiles. Cela sous-entend que le mobile peut communiquer avec deux routeurs d'accès ou plus en même temps pour demander la disponibilité des ressources et faire la réservation avant l'exécution du Handover. Cela est possible dans les réseaux cellulaires hot-spot (aéroport, gare, hôtel…), tel que, par exemple, les réseaux locaux sans fil [58]. Concevoir que le mobile communique avec plusieurs routeurs d'accès en même temps implique aussi que plusieurs routeurs d'accès couvrent la même région ; de ce fait le coût d'une telle installation est élevé.

Initialement, le transfert de contexte conçu pour améliorer le Handover entre les routeurs d'accès en leurs permettant de faire communiquer entre eux, directement ou par l'intermédiaires du mobile, le contexte la QoS de ce mobile en mouvement. Un raffinement de ce concept permettra aux routeurs d'accès et aux passerelles de communiquer le contexte du mobile durant le Handover ; cela permettra la réduction du temps dans lequel le mobile n'a pas de ressources qui lui sont allouées. Les extensions de cette architecture ont aussi pour but de faire communiquer les routeurs d'accès et les passerelles des domaines pour rétablir la QoS sur tout le chemin en envoyant le contexte du mobile à tous les routeurs jusqu'à la passerelle[58].

3. Interopérabilité entre la AAA et la Mobilité :

Mobile IPv4 permet à un MN de changer son point d'attachement à Internet pendant qu'il maintient une adresse permanente fixe. Les paquets IP envoyés à l'adresse permanente sont interceptés par le HA, transmis au MN par l'intermédiaire de son point d'attachent courant en utilisant un tunnel IP. Le FA, quand il est installé dans le point d'attachement pour qu'il serve entre autres comme le bout du tunnel IP, peut aussi servir au contrôle d'accès pour le réseau visité ; l'une des ses fonctionnalités est l'authentification de tout MN qui pourrait se présenter et qui appartiendrait au même domaine administratif ou à un domaine administratif différent. Le FA doit vérifier si le MN est autorisé à s'attacher à lui et à utiliser les ressources et/ou le service du réseau visité. Il devrait aussi, dans l'absence d'un mécanisme spécifique, fournir au

domaine d'origine du MN des informations relatives aux ressources utilisées par ce MN pendant qu'il est attaché au domaine visité.

Dans une architecture AAA de mobilité, il y a le serveur AAA du réseau d'origine (le serveur AAAH) d'un MN et le serveur AAA du réseau visité (le serveur AAAF) Figure III-1. Quand un MN se déplace et entre dans la zone de couverture d'un réseau visité, il aura besoin d'un accès sans fil. Il demande ensuite, par l'intermédiaire du réseau visité, une authentification chez son serveur AAA - qui est dans ce cas le AAAH- dans le réseau d'origine. Si l'authentification avec le AAAH est un succès, le MN est desservi par le réseau visité avec un accès au réseau et est autorisé à utiliser les ressources et/ou les services qu'il demande s'il en a le droit. Si l'authentification échoue, l'accès demandé par le MN est rejeté.

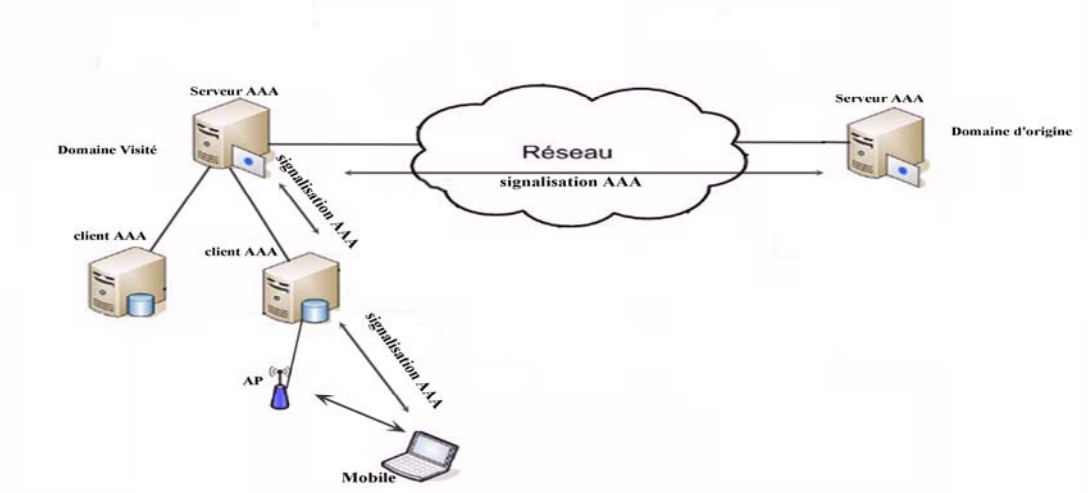


Figure III-1 : La signalisation AAA dans la mobilité.

La AAA est indispensable à la gestion de la mobilité mais elle introduit une difficulté supplémentaire pour assurer la QoS dans des réseaux sans fil mobiles. Le problème majeur de la gestion de la mobilité est d'achever l'opération de la mobilité le plus rapidement possible car, pour la même session, un mobile sera contraint de s'authentifier à chaque fois qu'il change de réseau et, dans la plupart du temps, à chaque fois qu'il change de point d'attachement. Selon le mécanisme d'authentification utilisé, la procédure d'authentification peut nécessiter plusieurs échanges de requêtes/réponses. Ajoutant à cela, La signalisation AAA entre le AAAF et le AAAH qui pourraient être géographiquement loin quand le MN se trouve dans un réseau visité pourrait être transportée à travers plusieurs domaines grâce à des agents de proxy/relais. La durée du de la procédure de l'authentification est dans ce cas rallongée introduisant un délai supplémentaire considérable à la durée d'exécution du Handover. Ce processus marquera de ce fait l'allongement de la durée de l'interruption qui, comme on la déjà vu précédemment, influence négativement la QoS dans les réseaux sans fil mobiles. Par conséquent, la QoS globale assurée pour un mobile serait très dégradée si les déplacements de ce mobile sont assez fréquents.

La principale préoccupation des protocoles COPS, RADIUS et Diameter est d'offrir une infrastructure pour la gestion de la signalisation AAA des utilisateurs connectés à un réseau mais non pas d'optimiser le processus d'authentification et d'autorisation pour réduire le temps d'exécution de ces procédures. Dans ce qui suit nous décrirons une application Diameter conçue pour la gestion de mobilité en utilisant les serveurs AAA, les agents de mobilité existants, le protocole Mobile IP et le protocole de base Diameter. Nous exposerons ensuite une solution utilisant cette application Diameter combinée à l'authentification USIM connue dans le monde télécommunication et visant à optimiser le temps de la procédure de l'authentification dans la mobilité.

3.1. L'application Mobile IP Diameter :

L'application Mobile IP Diameter [37] exploite le protocole Diameter de base et utilise les agents de mobilités HA et FA et le protocole Mobile IPv4 pour offrir le service de mobilité. Dans cette application le HA et le FA réagissent comme des clients Diameter. Les MN, parce qu'ils n'implémentent pas Diameter, interagissent avec les FA et leurs HA en utilisant seulement Mobile IP. Le domaine d'origine contient un serveur Diameter d'origine appelé AAAH et le domaine visité contient un serveur Diameter visité appelé AAAF **Figure III-2**. Le AAAF et le AAAH peuvent établir entre eux une session Diameter directe ou par l'intermédiaire d'un réseau de proxy/relais Diameter tel qu'il est défini dans le protocole Diameter de base[35].

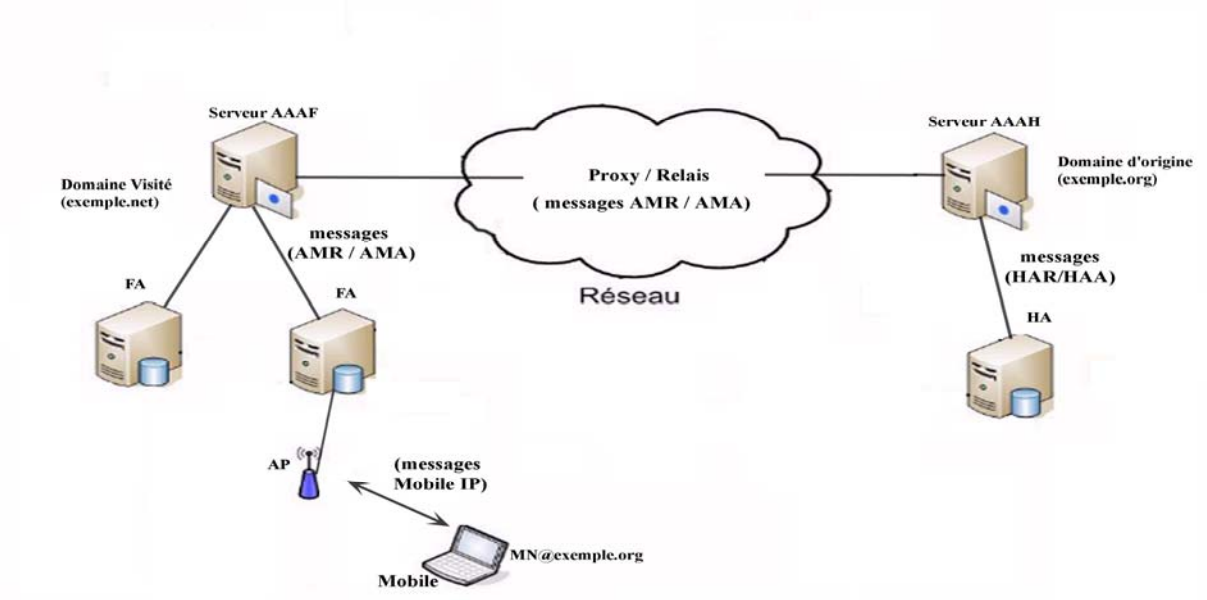


Figure III-2: L'architecture Diameter mobile IP

3.1.1. Fonctionnement du protocole :

L'application Mobile IP Diameter crée quatre nouveaux quatre nouveaux messages au protocole Diameter de base qui sont échangés entre les entités de mobilité et les serveurs AAA et qui transportent les nouvelles informations nécessaires à la gestion de la mobilité. Les quatre nouveaux messages sont le *AA-Mobile-Node-Request* (AMR), le *AA-Mobile-Node-Answer* (AMA), le *Home-Agent-MIP-Req*uest (HAR) et le *Home-Agent-MIP-Answer* (HAA).

Pour qu'un MN demande une mise à jour de la localisation auprès de son HA, il lui envoie un message *Registration-Request* de MIPv4 par l'intermédiaire du FA. Dans Diameter Mobile IP, quand le FA reçoit ce message, il crée un message AMR qui comprend des AVP spécifiques à l'application Diameter Mobile IPv4 (Diagramme de séquence III-1). L'adresse permanente, l'adresse du HA et d'autres champs importants sont prélevés du message *Registration-Request* pour être encapsulés dans ces AVP. Le message AMR est ensuite envoyé par le FA au serveur Diameter local qui est un AAAF ou un AAAH. A sa réception, le AAAF saura, selon le processus du protocole Diameter, si le message est traité localement ou transmis à un autre serveur Diameter qui est le AAAH du MN. Il n'est pas indispensable que le FA sollicite les services AAA à chaque fois qu'il reçoit un message *Request Registration* mais cela est nécessaire quand la période d'autorisation définie dans la communication Diameter est expirée. Ce délai de l'autorisation est toujours supérieur ou égal au délai de validité de l'enregistrement du MN défini dans le message *Registration-Request*.

Selon l'application d'authentification utilisée entre le MN et son AAA, le FA entame la procédure d'authentification et peut envoyer un challenge au MN qui renvoie ensuite une réponse afin de permettre son authentification et son autorisation par le AAAH. L'application d'authentification typique qui peut être utilisée est l'application EAP Diameter. Si les données d'authentification fournies par le MN sont fausses, le AAAH peut renvoyer un message AMA signalant le rejet de la demande. Si le MN est authentifié avec succès et autorisé, le AAAH envoie au HA un message HAR contenant les données du message *Registration-Request* de Mobile IPv4. A la réception du message HAR, le HA le traite et crée le message *Registration-Reply* de MIPv4 ; il l'encapsule ensuite dans le message HAA avant de l'envoyer au AAAH. Quand le AAAH reçoit le message, il envoie le message AMA au AAAF qui le transmet au FA. Ce dernier construit un message *Registration-Reply* et le transmet au MN en lui autorisant l'accès et en lui offrant le service.

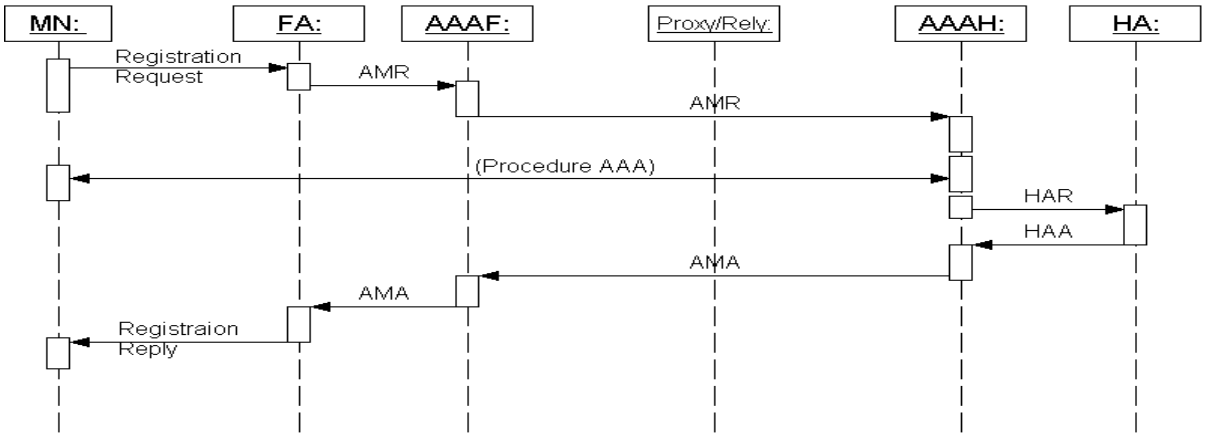


Diagramme de séquence III-1 : communication Diameter Mobile IPv4.

3.2. Optimisation de l'authentification dans les réseaux IP mobiles :

Dans [64], on propose une méthode d'authentification qui combine l'infrastructure AAA et la sécurité UMTS. Les concepts AAA et les mécanismes USIM (Universal Subscriber Identity Module) sont combinés afin d'offrir une authentification efficace et rapide dans l'environnement des réseaux IP sans fil mobiles. Le protocole AAA choisi est le protocole Diameter Mobile IP. Dans cette proposition, il s'agit d'exploiter l'infrastructure Diameter et le processus d'authentification fonctionnera selon le mécanisme USIM. Diameter supporte l'environnement mobile mais avec la procédure d'authentification dans Mobile IP, les messages peuvent transiter par plusieurs agents de proxy ou de relais en ajoutant un délai au temps d'exécution de l'authentification. En s'inspirant du protocole USIM, cette solution optimise l'authentification en ne laissant les messages transiter jusqu'au réseau d'origine que dans le cas de la première connexion au domaine par le mobile.

Les réseaux sans fil tel le IEEE802.11 incluent les protocoles d'authentification EAP/MD5 et EAP/TLS. Ces algorithmes utilisent le mécanisme challenge/réponse qui exige plusieurs échanges de messages. Dans la mobilité, ils ne sont pas performants à cause de la latence qu'ils exigent parce qu'à chaque fois que le mobile change de cellule et exécute un Handover, une authentification est exécutée avec plusieurs échanges de messages auprès de son AAAH qui pourrait être géographiquement loin.

3.2.1. Le mécanisme d'authentification USIM :

La spécification de USIM définit le mécanisme de sécurité des réseaux 3GPP (les systèmes de télécommunication mobile de 3ème génération) [64] . L'algorithme USIM est composé d'un

protocole de challenge/réponse et d'un protocole d'établissement de clé combiné à un protocole de numéro de séquence pour l'authentification. Il peut être utilisé pour n'importe quel type de réseau sans fil mobile et aussi pour l'accélération du nomadisme. Il est déployé sur une infrastructure hiérarchique divisée en station mobile (le mobile) qui est le niveau « 0 », en domaine visité qui est le niveau « 1 » et en environnement d'origine qui est le niveau « 2 ».

La station mobile commence par envoyer son identité au serveur visité (visited server) dans le domaine visité (visited domain). Le serveur visité cherche ensuite le serveur d'origine (home server) dans l'environnement d'origine (home environment) et lui transmet la requête d'authentification (Diagramme de séquence III-2). Le serveur d'origine vérifie l'identité et génère des AV (Authentication Vectors) qui contiennent les informations nécessaires à l'authentification et les clés du cryptage ; il transmet ensuite ces AV au serveur visité. Un AV contient un numéro aléatoire « RAND » pour le challenge, la réponse attendue pour ce challenge « XRES », une clé de chiffrement « CK », une clé d'intégrité et un "Token d'authentification" « AUTN ». A la réception de ces AV, le serveur visité sélectionne un AV et prend de l'AV sélectionné deux éléments qui sont « RAND et AUTN ». Il envoie ensuite ces deux éléments à la station mobile qui vérifie en utilisant « AUTN », la plage de numéro de séquence. Si celle-ci est vérifié avec succès, la station mobile calcule, en utilisant « RAND », la réponse « RES » et l'envoie au serveur visité. Celui-ci compare la réponse « RES » calculée par la station mobile à la réponse « XRES » envoyée par le serveur d'origine. Si elles correspondent l'une à l'autre, le serveur visité envoie un message de réponse positive à la station mobile. L'authentification est ainsi accomplie et la communication s'établit avec le chiffrement en utilisant la clé « CK ».

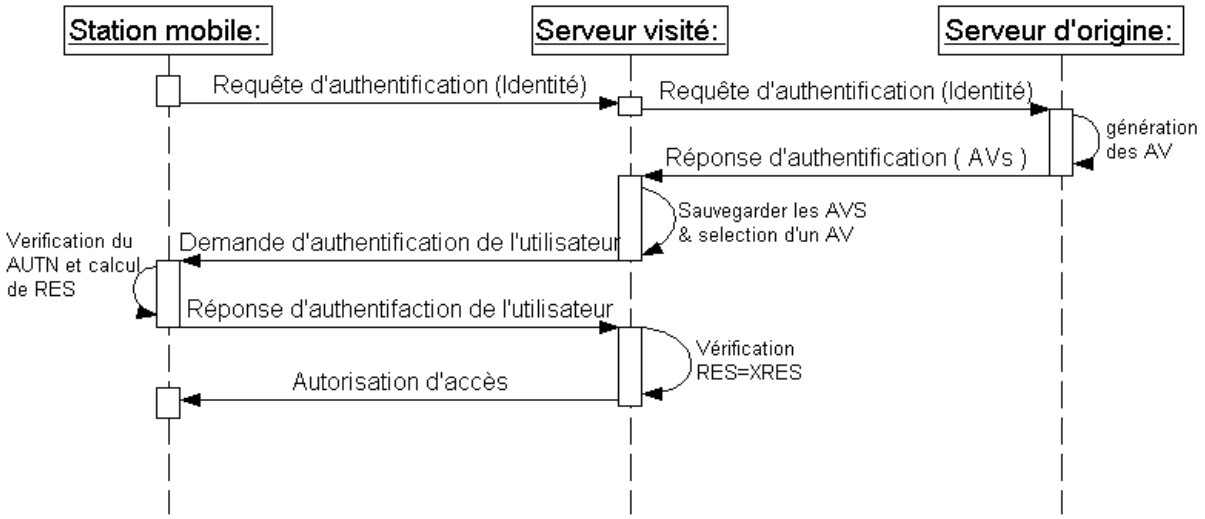


Diagramme de séquence III-2 : Authentification USIM.

3.2.2. Combinaison du Diameter et du USIM :

Avec le protocole USIM, le serveur d'origine n'est pas consulté à chaque fois qu'une authentification est requise. L'ensemble des éléments d'authentification nécessaires est chargé dans le serveur visité qui se trouve volontairement près de la station mobile ; ceci améliore les performances de la mobilité et du nomadisme. Dans la combinaison du Diameter et du USIM [64], le processus d'authentification USIM est appliqué sur l'infrastructure AAA de Diameter. L'objectif de cette combinaison est d'offrir une authentification AAA rapide et optimale, particulièrement avec le protocole Diameter, sans se soucier de la longue distance géographique entre le mobile et le serveur d'origine et du nombre d'agents de proxy/relais qui

les séparent. Dans cette proposition[64], l'authentification du type challenge/réponse est préconisée. C'est l'algorithme MD5 qui est utilisé pour calculer la réponse attendue et la vérification de la réponse du mobile. En effet, les échanges de l'authentification de type challenge/réponse introduisent une latence significative particulièrement dans les réseaux sans fil mobile parce que les Handover sont fréquents.

L'infrastructure de cette authentification se compose du centre d'authentification qui est le serveur AAA lequel est le seul générant des vecteurs d'authentification, d'un proxy uniquement sélectionné au moment de la première authentification appelé "broker" et supportant le nouveau mécanisme et d'un client AAA qui reçoit la demande d'authentification du mobile. Le broker est sélectionné d'une manière à ce qu'il soit le plus proche possible du mobile et supporte le type et le mécanisme de l'authentification. Cette architecture permet d'éliminer la charge du réseau due au mécanisme d'authentification challenge/réponse.

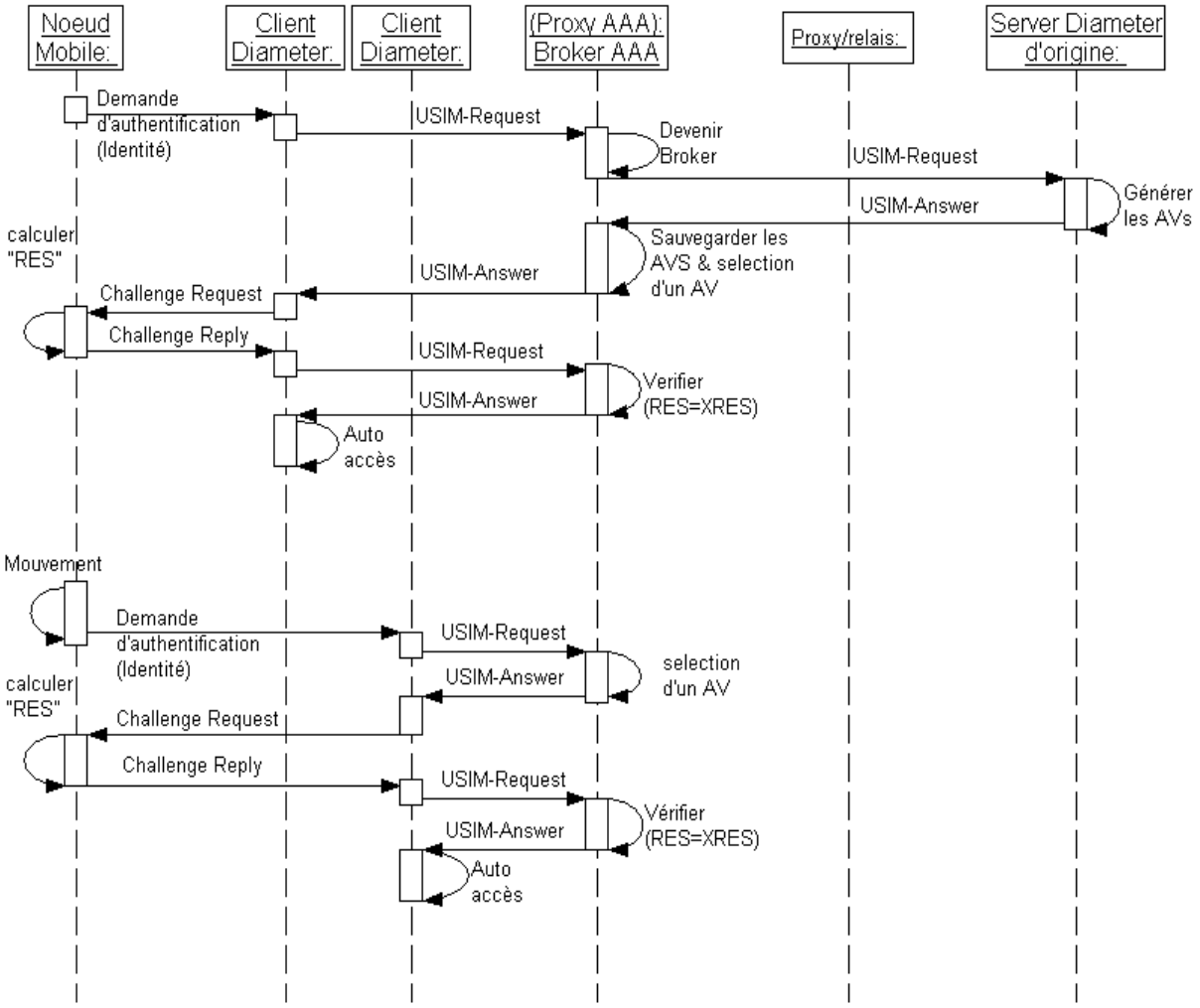


Diagramme de séquence III-3 : combinaison du Diameter et du USIM.

Quand le mobile visite un domaine, il demande une authentification en envoyant son identité. A la réception de la requête, le client AAA envoie un message Diameter "*USIM-Request*" au serveur d'origine correspondant par l'intermédiaire des agents de proxy/relais (Diagramme de séquence III-3). Le message *USIM-Request* indique que le challenge est exigé. Un proxy recevant le message *USIM-Request* vérifie le message pour savoir s'il peut devenir un broker. S'il n'y a pas dans le message *USIM-Request* l'indication de la présence d'un broker, le proxy lui-même devient, s'il le peut, un broker, et modifie le message et le

transmet ensuite. Dans l'autre cas, il le transmet sans modification et un autre proxy deviendra un broker si cela est possible.

Quand le serveur AAA d'origine reçoit le message *USIM-Request*, il génère les numéros aléatoires « RAND » et calcule leurs réponses « XRES » respectives en utilisant l'algorithme MD5 et la clé partagé avec le communicant direct ou avec le mobile. Il construit alors un ensemble d'AV (Authentication Vector) dont chacun est composé de l'User-ID, du « RAND » et du « XRES ». Avant d'envoyer le message *USIM-Answer*, le serveur AAA d'origine vérifie le message *USIM-Request* reçu. Le broker reçoit le message *USIM-Answer* et sauvegarde les AV qui seront utilisés pour authentifier le mobile correspondant quand il se déplace d'une cellule à une autre. Le broker extrait un « RAND » d'un AV et l'envoie au client AAA pour le challenge. A la réception du « RAND », le client AAA envoie le challenge au mobile. Ce dernier calcule une réponse « RES » en utilisant le « RAND » et la clé partagée selon l'algorithme MD5 et l'envoie au Client AAA qui envoie par la suite un message *USIM-Request* contenant la réponse « RES ». Le broker, quand il reçoit le message, compare la réponse « RES » et la réponse « XRES » correspondante envoyée par le serveur et vérifie si ces réponses sont égales. Après la vérification, il répond avec un message de rejet ou d'acceptation. L'AV utilisé pour l'authentification est immédiatement éliminé.

Quand le mobile se déplace vers une nouvelle cellule, il demande une authentification (Diagramme de séquence III-3). Une fois le mobile identifié, le broker renvoie à ce mobile un challenge en utilisant l'AV suivant. La procédure d'authentification continue sans solliciter le serveur AAA d'origine et l'AV utilisé sera ensuite éliminé.

4. Conclusion

Afin de montrer la difficulté d'exécuter la QoS et la AAA dans les réseaux sans fil mobiles, ce chapitre analyse l'interaction entre ces trois architectures. La première partie de ce chapitre concerne l'influence de la mobilité sur la QoS. Les réseaux sans fil mobiles, par leurs caractéristiques, contraignent la QoS à prendre en considération certains aspects de la mobilité pour que son installation soit utile dans ce type de réseau. L'analyse faite dans ce chapitre explique comment l'exécution du Handover, en fonction de son exécution et de sa profondeur, peut faire dégrader le niveau de la QoS. Le Handover est lié au mouvement du mobile ; d'autres mécanismes liés à la macro-mobilité peuvent empêcher le fonctionnement normal et rapide de la QoS. Selon leurs types d'architectures et des mécanismes de routage qu'elles utilisent, la macro-mobilité et la micro-mobilité influencent la QoS différemment sur IntServ ou sur DiffServ. Par la suite, des mécanismes et des architectures, qui peuvent permettre à la QoS d'être intégrée convenablement dans les réseaux mobiles, sont exposés. Ces solutions sont les plus discutées dans la communauté de la mobilité et certaines d'entres elles ont été simulées et/ou réalisées ; elles ouvrent aussi des perspectives intéressantes pour la QoS dans les réseaux mobiles.

La deuxième partie de ce chapitre est consacrée à l'interopérabilité entre la AAA et la mobilité. Faute d'architecture et de Handover, les réseaux sans fil mobiles rendent la signalisation AAA difficile et délicate. L'exécution de la signalisation AAA dans ce type de réseaux influence indirectement la QoS par le délai qu'elle introduit. La solution de ce problème, exposée dans ce chapitre, exploitant l'infrastructure de la AAA sur la mobilité offre une bonne alternative pour réduire le temps d'exécution de la procédure de l'authentification en s'inspirant du mécanisme d'authentification utilisé dans les réseaux de télécommunication de troisième génération. L'analyse faite dans ce chapitre va servir de base à notre proposition qui consiste à unifier les trois architectures de mobilité, de QoS et de AAA en utilisant un seul protocole et qui sera abordée dans le prochain chapitre.

Chapitre IV

Une architecture unifiée

1. Introduction :

L'analyse faite dans le chapitre précédent nous a permis de constater que si l'établissement de la route du flot d'un mobile, en tenant compte de l'installation de la QoS, se fait très rapidement au moment de la connexion du mobile au nouveau routeur d'accès, cela engendrerait une amélioration conséquente du service, autrement dit de la QoS fournie aux mobiles en mouvement. Cette amélioration suggère l'optimisation du temps d'exécution du Handover tout en assurant le bon fonctionnement de la communication et des protocoles de signalisation. Les solutions nécessaires pour arriver à cette fin sont liées à la signalisation des protocoles d'authentification, de la QoS et de la mobilité car ces multiples signalisations causent une grande latence dans l'établissement de la session, tout simplement parce qu'elles sont indépendantes et que le commencement d'une procédure est lié à l'achèvement d'une ou d'autres procédures. Cette indépendance est due essentiellement au fait que ces protocoles et ces architectures ont évolué séparément.

L'exploitation de la QoS dans les réseaux mobiles, tout en assurant la sécurité et la QoS elle-même, rend l'introduction d'une architecture unifiant les protocoles de signalisation relatifs à la QoS, à la mobilité et à la AAA vitale et incontournable. Ce type d'architecture que nous appelons architecture unifiée permettra immanquablement, en utilisant une signalisation unifiée, de réduire la charge du réseau et d'optimiser le temps d'exécution du Handover. La signalisation unifiée ne minimisera pas seulement la latence dans l'établissement de la session mais apportera également une facilité de contrôle dans le réseau car elle permet de prendre en considération les contraintes d'une signalisation spécifique au moment de l'exécution d'une autre signalisation. Aussi, sera-t-elle probablement une première étape vers le processus de standardisation d'une signalisation inter-domaine.

Ce chapitre expose notre solution qui consiste en une architecture de signalisation unifiée de la AAA, de la QoS et de la mobilité en utilisant un protocole unique et une infrastructure adéquate. Après avoir formulé les différents objectifs de notre travail, nous présenterons notre architecture unifiée en commençant par la description de l'infrastructure déployée pour installer les entités participantes à l'exécution et à la gestion du protocole. Par la suite, nous poursuivrons par la description des messages du protocole et de son fonctionnement sur cette infrastructure.

2. L'architecture UNISIG :

Afin de mettre en place une signalisation commune à la signalisation d'authentification, de la gestion de la QoS et de la gestion de la mobilité une architecture se composant de plusieurs entités fonctionnelles et statiques est proposée. Cette architecture, appelé UNISIG (UNIfied SIGnaling) permet l'exécution du protocole UNISIG Diameter lequel est conçu pour

transporter les trois signalisations comme étant une application Diameter. Dans cette architecture une méthode EAP appelée méthode EAP/UNISIG est introduite pour la signalisation entre le mobile et le réseau d'accès. Nous présenterons d'abord, dans cette section, l'infrastructure qui est la composition matérielle de l'architecture et l'emplacement des entités fonctionnelles qui interagissent ou qui communiquent, puis nous poursuivrons par la description des fonctionnalités de ces entités et le rôle de chacune d'elles pour accomplir la signalisation du protocole.

2.1. Infrastructure :

L'architecture proposée considère l'Internet comme un ensemble de domaines dont chacun est un réseau sous le contrôle d'une administration (opérateur). Une administration peut avoir un ou plusieurs domaines ; un domaine est constitué d'un réseau filaire qui le relie à Internet et auquel se connectent plusieurs sous réseaux sans fil (Figure IV-1). Un réseau sans fil se constitue d'un routeur d'accès qui le relie au réseau filaire du domaine et d'un ou plusieurs points d'accès (AP/BS) qui assurent au mobile la connectivité sans fil.

Un domaine est considéré comme un seul ou un ensemble de sous domaine DiffServ avec ses routeurs de bord et ses routeurs de cœur. Sur chaque routeur de bord un client QoS Diameter (cf. ci-dessous) est installé. Notons qu'un routeur d'accès est aussi un routeur de bord qui reçoit et transmet des paquets des mobiles et vers les mobiles. Sur chaque point d'accès (AP) un MiMM (Micro-Mobility Manager) est implémenté pour gérer les mobiles et le flot de signalisation avec ces mobiles. Sur chaque routeur d'accès un MaMM (Macro Mobility Manager) est installé pour gérer le réseau sans fil et, dans chaque domaine, il y a un RMM (Roaming Mobility Manager) installé dans le serveur pour gérer l'authentification, la QoS et la mobilité dans le domaine (cf. ci-dessous). Pour assurer la signalisation entre le mobile et le domaine, un MC (Mobility-Client) est installé sur le mobile.

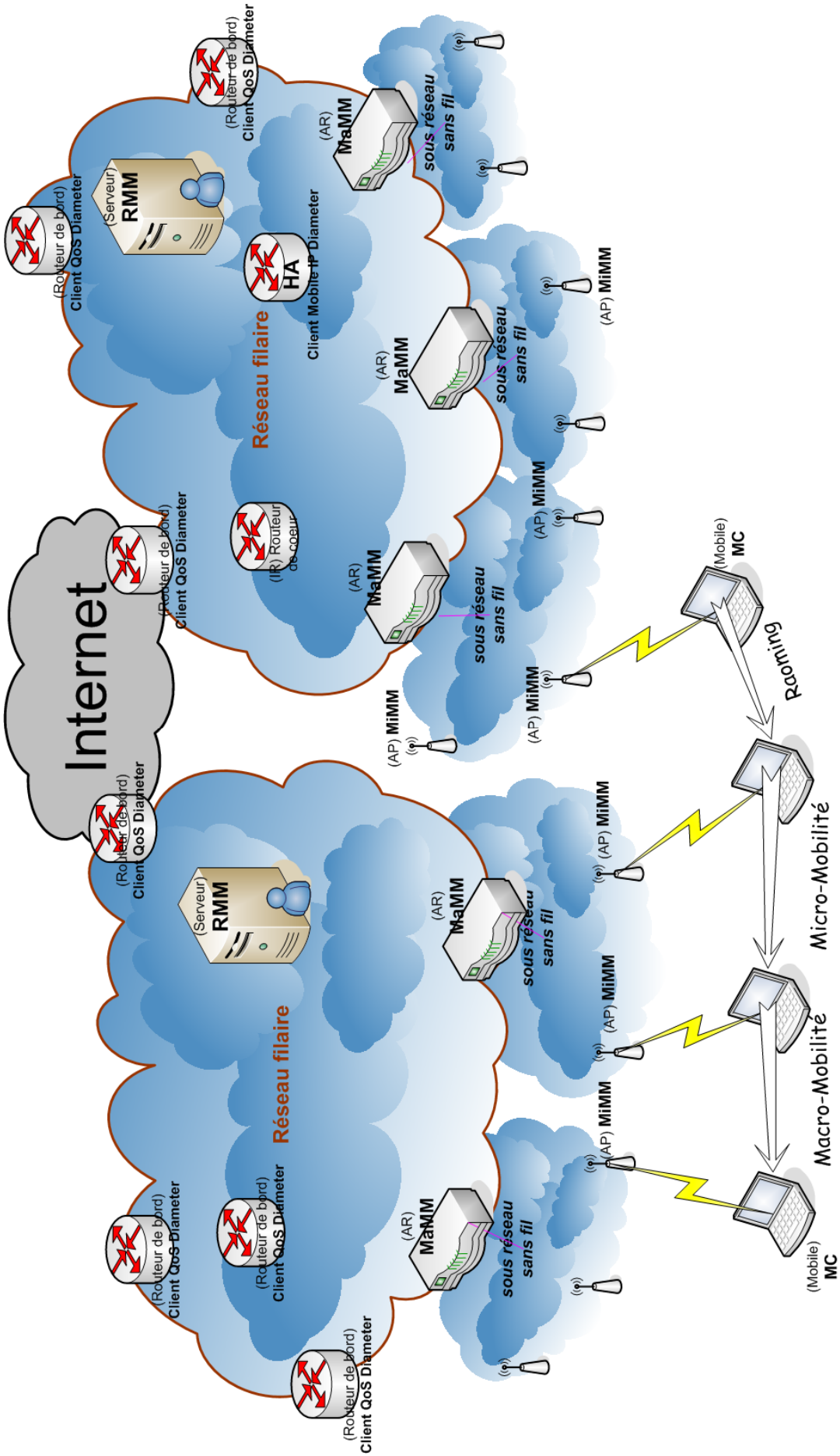


Figure IV-1 : Infrastructure de
e IV-1 : Infrastructure de

2.2. Les agents Diameter :

Afin d'utiliser l'application UNISIG Diameter dans cette architecture, trois nouveaux agents Diameter sont définis (Figure IV-2). Ces agents sont considérés comme des éléments de base utilisés par les entités fonctionnelles de l'architecture pour accomplir une signalisation unifiée. Ils sont le client UNISIG Diameter, le client QoS Diameter et le serveur UNISIG Diameter qui pourrait être en même temps un proxy Diameter dans certains cas. Le client Mobile IP Diameter qui est introduit dans l'application Diameter Mobile IP[37] est aussi utilisé dans cette architecture. Cette sous-section décrit le rôle de chacun de ces agents.

2.2.1. Client UNISIG Diameter :

Le client UNISIG Diameter assure les fonctionnalités du client Diameter de base mais traite aussi les nouveaux messages de l'application UNISIG Diameter. Il communique avec le serveur UNISIG Diameter pour accomplir l'exécution de l'application UNISIG Diameter. Sa fonction principale est d'assurer le rôle d'authentificateur et de relier la signalisation entre le mobile et le serveur UNISIG Diameter. Il construit des messages Diameter à partir des messages de la méthode UNISIG/EAP envoyés par le mobile et des messages UNISIG/EAP à partir des messages Diameter envoyés par le serveur/proxy Diameter.

2.2.2. Client QoS Diameter :

Le client QoS Diameter est chargé d'installer les configurations de la QoS des mobiles sur un routeur de bord d'une architecture DiffServ. Ces installations de configuration (règles) sont demandées par le serveur UNISIG Diameter en lui envoyant des messages de requêtes Diameter transportant les AVP appropriés. Cet agent interprète tous les AVP et les nouveaux messages de l'application UNISIG Diameter spécifique à la QoS. Il utilise le mécanisme du PEP (Policy Enforcement Point) pour installer les états de configuration des requêtes dans une gestion de réseau par politique selon le modèle d'approvisionnement COPS (COPS-PR[40]).

2.2.3. Client Mobile IP Diameter :

L'agent Client Mobile IP Diameter est l'agent Diameter introduit dans la spécification de l'application Mobile IP Diameter ([37], section II.3.1). Il est utilisé dans cette architecture pour interagir avec le serveur UNISIG afin d'accomplir la gestion de la mobilité. Pour communiquer entre eux, le serveur UNISIG Diameter et le client Mobile IP Diameter utilisent les deux messages *HAR* et *HAA* de l'application Mobile IP Diameter. Cet agent est installé sur chaque HA du domaine d'origine.

2.2.4. Serveur UNISIG Diameter :

Le serveur UNISIG Diameter assure les fonctionnalités d'un serveur d'authentification Diameter. Il peut aussi agir comme un proxy UNISIG Diameter dans certaines procédures pour assurer l'authentification intra domaine et inter domaine quand il est dans un domaine visité par un mobile. Pour assurer l'installation des configurations de la QoS sur les routeurs de bord et la gestion de tous les clients QoS Diameter, le serveur UNISIG Diameter implémente les mécanismes du PDP. Sur ce plan, il prend des décisions pour assurer la gestion du réseau par politique afin de garantir la QoS contractée avec le mobile et transmet les requêtes aux clients QoS Diameter du domaine. Quand il est dans un domaine d'origine, en plus de ces fonctionnalités, il assure la communication avec le client Mobile IP Diameter afin d'accomplir la mise à jour de la localisation.

2.3. Les entités de l'architecture UNISIG :

Les entités de l'architecture unifiée sont les entités fonctionnelles qui utilisent principalement les agents Diameter afin d'effectuer la gestion de la QoS et de la mobilité et l'authentification (Figure IV-2). Elles peuvent avoir à gérer aussi plusieurs bases de données et bases d'informations ainsi que des tables d'association ou de correspondance. Quand on présente l'exécution du protocole UNISIG Diameter, on parle de ces entités qui déroulent l'exécution sans avoir à préciser si elles vont utiliser le serveur ou le proxy ou bien le client ou l'authentificateur. Elles sont responsables de la synchronisation de la transmission et du traitement des messages Diameter selon la phase dans les procédures d'authentification, d'installation de la QoS et de la mise à jour de la localisation qui seront décrites par la suite. Elles se chargent aussi d'inviter le mobile à envoyer les informations adéquates selon le type du Handover qu'il va effectuer. Ces entités sont le RMM, le MaMM, le MiMM et le MC.

2.3.1. MiMM :

Le MiMM (Micro-Mobility Manager) est installé sur le point d'accès/station de base (Figure IV-2). Il implémente un client UNISIG Diameter afin d'assurer la signalisation avec le mobile (avec le MC) et avec le MaMM. Il assume le rôle du relais entre le mobile et le serveur en utilisant le mécanisme de IEEE802.1X [48]. Le MiMM ne communique qu'avec le MaMM installé sur le routeur d'accès qui gère le AP sur lequel il est installé.

2.3.2. MC:

Pour interagir avec le MiMM installé dans le AP/BS auquel il s'accroche au moment du Handover, le mobile utilise le MC (Mobility-Client). Le MC implémente la méthode EAP/UNISIG qui permet de construire les messages de requête/réponse EAP pour cette méthode EAP. Ces messages transportent simultanément les informations d'authentification et de la mobilité quand cela est possible et selon la phase de la procédure d'authentification et de la mise à jour de la localisation.

2.3.3. MaMM :

Le MaMM (Macro-Mobility Manager) garantit les fonctionnalités d'un FA pour le routage Mobile IP et la gestion de la mobilité. Il implémente un serveur UNISIG Diameter pour l'authentification Intra-Domaine et Intra-AR (Intra-Access Router). Etant aussi sur un routeur de bord, il implémente un client QoS Diameter pour l'installation de la configuration de la QoS (Figure IV-2). Il est responsable de la gestion d'un ensemble de MiMM installés sur les AP reliés à son routeur d'accès et appartenant à son sous réseau sans fil (Figure IV-1). Il participe avec son proxy Diameter à l'authentification intra-domaine et inter-domaine et à la mise à jour de la localisation du mobile.

Le MaMM gère une base de données d'authentification locale pour l'authentification Intra-AR et une table d'association de MiM (Micro-Mobility) pour manager les mobiles qui sont connectés à son sous réseau sans fil. Quand le mobile circule dans son sous réseau sans fil, le MaMM maintient l'association de MiM. Une association de MiM contient l'identité du mobile, l'identifiant du AP/BS auquel le mobile est attaché et son adresse permanente. Pour la première connexion du mobile au routeur d'accès, le MaMM crée une entrée de la table d'association MiMM qui correspond à ce mobile.

2.3.4. RMM :

Le RMM (Roaming Mobility Manager) est responsable de l'administration du domaine pour assurer l'authentification des mobiles connectés, la gestion de la QoS et la gestion de la mobilité ; il est installé sur le serveur du domaine (Figure IV-2). Grâce à son serveur UNISIG

Diameter, le RMM communique avec les autres entités de l'architecture qui sont les H-RMM, les MaMM et les clients QoS Diameter. Il est responsable de la gestion d'un ensemble de MaMM installés sur les différents routeurs d'accès de son domaine et interagit avec eux pour exécuter l'authentification intra-domaine et participer à la mise à jour de la localisation des mobiles. Il gère aussi les clients QoS Diameter installés sur les routeurs de bord ainsi que sur les routeurs d'accès et interagit avec eux pour installer la configuration de la QoS dans un réseau DiffServ. Ce mécanisme est assuré grâce au PDP (Policy Decision Point) intégré dans le serveur UNISIG Diameter du RMM. Pour accomplir l'authentification inter-domaine des mobiles connectés à son domaine, il interagit avec d'autres H-RMM de ces mobiles. Cette authentification inter-domaine utilise le mécanisme de l'application USIM Diameter [64]. Le RMM, quand il est dans le domaine d'origine d'un mobile, il interagit avec client Mobile IP Diameter installé sur le HA du mobile pour la mise à jour de la localisation et aussi avec les clients QoS Diameter pour l'installation de la configuration de la QoS dans le domaine d'origine.

Le RMM gère une base de données d'authentification pour sauvegarder les informations d'authentification des utilisateurs abonnés à l'administration de son domaine. Dans cette base de données, il sauvegarde aussi les données d'authentification des utilisateurs abonnés à d'autres administrations conventionnées avec son administration. Il gère également une base de données politiques pour sauvegarder les règles à appliquer dans les routeurs de bord pour les flots des utilisateurs utilisant les services du réseau DiffServ. Il sauvegarde par ailleurs dans cette base de données les SLS des abonnées à son administration et à d'autres administrations conventionnées. Il utilise en outre une base d'informations de disponibilité de ressources de tout le domaine pour la gestion de la QoS et principalement pour appliquer le contrôle d'admission au niveau du domaine.

Il gère une table d'association MaM (Macro-Mobility) contenant les informations sur l'identité du mobile et sa localisation effective dans son domaine. Il maintient une association MaM pour tout mobile qui visite son domaine et cela tant que ce mobile circule dans son domaine. L'association MaM se constitue de l'identité du mobile, de son adresse permanente, de sa CoA (ou de sa CCoA quand il s'agit d'une adresse routable) et de l'adresse du routeur d'accès (le MaMM) auquel il est connecté. Pour une nouvelle connexion du mobile au domaine, le RMM crée une entrée dans la table d'association MaM correspondante à ce mobile.

Un RMM est nommé H-RMM (Home RMM) quand il est dans le réseau d'origine d'un mobile et est nommé V-RMM (Visited RMM) quand il est dans un domaine visité par un mobile.

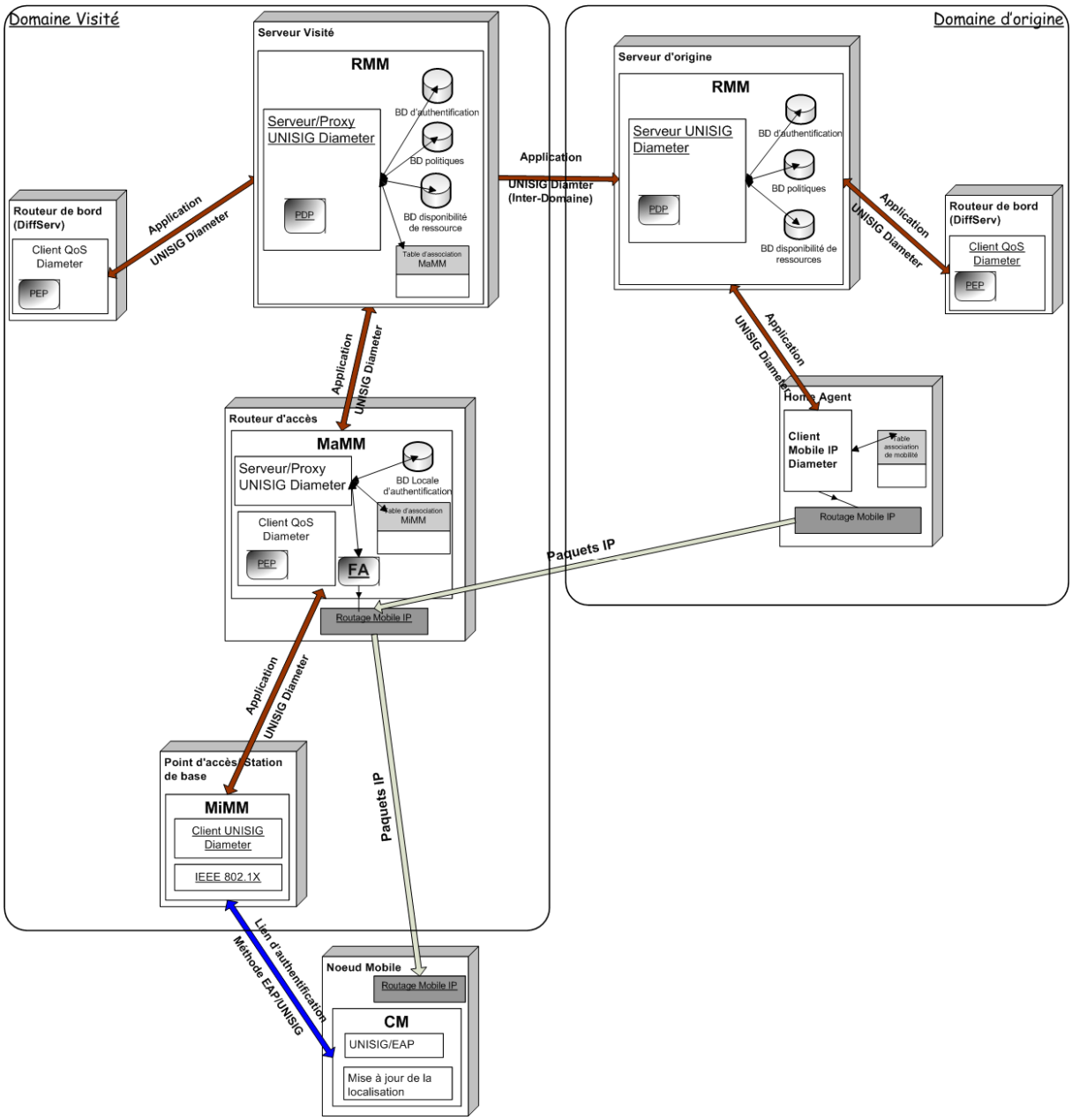


Figure IV-2 : Architecture fonctionnelle pour UNISIG Diameter.

3. Le protocole UNISIG Diameter :

A l'intérieur du domaine et entre les domaines, les entités de cette architecture, pour la signalisation, utilisent l'application UNISIG Diameter. Pour cela, en plus de ce que le protocole Diameter de base définit, de nouveaux messages et de nouveaux AVP y sont introduits. Entre le domaine et le mobile, sur la liaison sans fil, la signalisation s'effectue avec le protocole EAP grâce à une nouvelle méthode EAP qui est définie pour transporter, en utilisant les requêtes/réponses EAP, les nouvelles informations introduites par cette architecture. Dans cette section nous présenterons en premier lieu les nouveaux messages et les nouveaux AVP de l'architecture UNISIG et ensuite nous présenterons le fonctionnement du protocole UNISIG.

3.1. Les nouveaux messages Diameter :

L'application UNISIG Diameter introduit quatre nouveaux codes de commande Diameter (quatre nouveaux messages) afin d'assumer son nouveau mécanisme en transmettant les informations nécessaires et en garantissant les traitements adéquats. Deux autres messages de l'application Mobile IP Diameter sont utilisés dans cette architecture. Ces messages qui suivent la même nomination que dans la spécification Diameter sont les suivants :

- *USR* (*Unified-Signaling-Request*) : est le message requête utilisé pour transporter les AVP relatifs à l'authentification et à la gestion de la mobilité. Il est envoyé par le MiMM au MaMM, par le MaMM au RMM ou par le V-RMM au H-RMM.
- *USA* (*Unified-Signaling-Answer*) : est le message réponse utilisé pour transporter les AVP relatifs à l'authentification et à la gestion de la mobilité. Pour répondre au message *USR*, il est envoyé par le H-RMM au V-RMM, par le RMM au MaMM ou par le MaMM au MiMM. Quand il est envoyé par le H-RMM au V-RMM, ce message transporte aussi le SLS du mobile.
- *PIR* (*Policy-Install-Request*) : est le message requête utilisé pour transporter les AVP relatifs à la QoS. Il est envoyé par le RMM aux clients QoS Diameter installés dans les routeurs de bord afin d'initier une session QoS et installer les états de configuration QoS sur ces routeurs. Il est aussi envoyé pour demander au client QoS Diameter le maintient/rafraîchissement, le changement ou l'effacement de l'état de configuration.
- *PIA* (*Policy-Install-Answer*) : est le message réponse utilisé pour transporter les AVP relatifs à la QoS. Pour répondre au message *PIR*, il est envoyé par les clients QoS Diameter au RMM afin de transmettre les résultats du traitement du message *PIR* et de l'installation de la configuration de la QoS.
- *HAR* (*Home-Agent-Request*) : est le message requête utilisé pour transporter les AVP relatifs à la gestion de mobilité mais il envoyé seulement par le H-RMM au client Mobile IP Diameter pour demander au HA la mise à jour de la localisation en encapsulant le message *Registration-Request* dans l'AVP **Registration-Request**. Il est le message défini dans la spécification de l'application Mobile IP Diameter.
- *HAA* (*Home-Agent-Answer*) : est le message requête utilisé pour transporter les AVP relatifs à la gestion de la mobilité. Ce message est envoyé par le client Mobile IP Diameter au H-RMM en réponse à *HAR*. Il transporte l'AVP **Registration-Reply** qui encapsule le message *Registration-Reply* du Mobile IP. Il est le message défini dans la spécification de l'application Mobile IP Diameter.

3.2. Les nouveaux AVP :

- Les AVP relatifs à la gestion de la mobilité

AVP **Registration-Request** : cet AVP encapsule le message *Registration-Request* du mobile IP.

AVP **Registration-Reply** : cet AVP encapsule le message *Registration-Reply* du mobile IP.

- Les AVP relatifs à la gestion de la QoS

AVP **Policy** : cet AVP encapsule les règles envoyées par le RMM pour l'installation de l'état de configuration de la QoS sur les routeurs de bord.

AVP **SLS** : cet AVP encapsule le SLS (Specification Level Service) d'un mobile envoyé par le H-RMM.

AVP **QoS-Install** : cet AVP encapsule soit la constante « **PASSIVE** » pour demander une installation passive de la configuration soit constante « **ACTIVE** » pour demander l'activation

de l'installation passive ou pour le rafraîchissement/mise à jour d'une installation activée soit la constante « **UNUNSTALL** » pour effacer un état de configuration de la QoS.

AVP **Session-QoS-ID** : cet AVP encapsule l'identifiant unique dans un domaine d'une session QoS afin de mettre en relation les mêmes sessions QoS et différentes sessions Diameter et il peut être aussi utilisé pour la comptabilité.

- Les AVP relatifs à l'authentification

AVP **AV** : cet AVP encapsule les AV (Authenticator Vector) générés initialement par le H-RMM. Un AV se constitue de l'identifiant du mobile, d'un challenge RAND, d'une valeur COMPT incrémentée à chaque challenge et de la réponse XRESP calculée avec l'algorithme MD5 ayant pour paramètre le RAND, le COMPT du même AV et le secret partagé (mot de passe) entre le H-RMM et le mobile.

AVP **RAND** : cet AVP encapsule le challenge RAND généré par le H-RMM.

AVP **COMPT** : cet AVP encapsule la valeur COMPT d'un compteur incrémenté.

AVP **RESP** : cet AVP encapsule la valeur de la réponse calculée par le MC du mobile en utilisant l'algorithme MD5 ayant pour paramètres les valeurs RAND et COMPT envoyées par le MiMM et le secret partagé entre le mobile et le H-RMM.

AVP **User-ID** : cet AVP encapsule l'identifiant du mobile qui est un NAI (Network Access Identifier [69]).

Les AVP Diameter de base

Pour la construction d'un message Diameter dans cette application, les règles restent les mêmes quant à l'emplacement des AVP et à leurs présences obligatoires ou optionnelles. L'AVP **Session-ID** est utilisé pour identifier la session Diameter ; ce qui n'est pas la même chose avec l'AVP **Session-QoS-ID** qui est utilisé pour une session de QoS dans le domaine. Une session QoS d'un mobile dans un domaine peut correspondre à plusieurs sessions Diameter différentes et le RMM est responsable de la mise en correspondance de ces deux types de sessions pendant le mouvement du mobile parce que, au Handover Inter-AR, la session Diameter sera sûrement changée mais la session QoS ne le sera seulement que dans un Handover Inter-Domaine.

Pour l'AVP **Result-Code**, plusieurs constantes sont définies afin de signaler les différentes nouvelles erreurs comme par exemple dans la situation où le mobile est authentifié avec succès mais la mise à jour de la localisation ayant échoué ; ou par exemple dans le cas où l'authentification du mobile est un succès mais le contrôle d'admission pour le flot de ce mobile au niveau du RMM estime que les ressources sont insuffisantes pour accepter le Handover. Les nouvelles constantes de l'AVP **Result-Code** déterminent soit l'authentification échouée soit l'authentification succès mais mise à jour de la localisation échouée, soit l'authentification succès et mise à jour de la localisation succès mais l'installation de la configuration de la QoS échouée soit que les trois opérations sont un succès et, dans ce cas, le flot du mobile sera accepté par le MiMM.

L'application UNISIG Diameter est signalée dans la phase de négociation du protocole Diameter en encapsulant la valeur adéquate dans l'AVP **Auth-Application-Id**. Le reste des AVP, obligatoires ou non, du protocole Diameter de base sont utilisés sans aucune nouveauté introduite.

3.3. Les nouveaux messages EAP :

Pour la signalisation EAP entre le point d'accès (MiMM) et le mobile (MC), un type de messages EAP est défini pour la méthode EAP/UNISIG. Dans l'intérêt de distinguer le contenu du bloc de données des messages requête/réponse EAP, on nomme le message EAP

selon ce qu'ils transportent comme informations. Les nouveaux messages sont *EAP-Challenge-RegRq-Request*, *EAP-Challenge-RegRq-Response*, *EAP-Reg-Request* et *EAP-Reg-Response*. En plus de ces messages, cette architecture utilise aussi quatre autres messages qui sont déjà définis dans le protocole EAP [42].

- *EAP-Identity-Request* : ce message est envoyé par le MiMM pour demander l'identité du mobile afin d'entamer l'authentification. Son bloc de données contient l'identifiant de l'AP sur lequel le MiMM est installé et les informations du message d'annonce (Advertisement Agent) pour que le MN détecte son mouvement.
- *EAP-Identity-Response* : ce message est envoyé par le MC du mobile pour transmettre l'identité de celui-ci au MiMM qui a demandé l'identité du mobile. Le bloc de données de ce message contient le NAI (Network Access Identifier [69]) du mobile.
- *EAP-Challenge-Request* : ce message est envoyé par le MiMM pour transmettre le défi envoyé préalablement par le serveur au mobile. Il transporte dans le bloc de données du message EAP le challenge (RAND) et une valeur (COMPT) incrémentée pour chaque génération d'un défi.
- *EAP-Challenge-Response* : ce message est envoyé au MiMM par le MC après qu'il ait calculé la réponse du défi. Le bloc de données de ce message contient la valeur de la réponse, autrement dit le Hash qui est le résultat de l'algorithme du Hashage appliqué sur les valeurs envoyées par le MiMM dans le message *EAP-Challenge-Request* et le mot de passe du mobile (secret partagé).
- *EAP-Challenge-RegRq-Request :* ce message est envoyé par le MiMM dans le cas où il transmet non seulement le défi au mobile mais il l'invite aussi à entamer la procédure de la mise à jour de la localisation. Il transporte dans le bloc de données du message EAP les mêmes données transportées dans le message *EAP-Challenge-Request*.
- *EAP-Challenge-RegRq-Response* : ce message est envoyé au MiMM par le MC du mobile en réponse au message *EAP-Challenge-RegRq-Request* après avoir calculé le défi et construit le message *Registration-Request* de Mobile IP. Il transporte dans le bloc de données du message EAP la réponse du challenge comme dans le message *EAP-Challenge-Response* et le message *Registration-Request* du Mobile IP sans modification.
- *EAP-Success* : ce message est envoyé par le MiMM pour informer le mobile du succès de la procédure et de l'autorisation de commencer la transmission et la réception de données.
- *EAP-Failure* : ce message est envoyé par le MiMM pour informer le mobile de l'échec de la procédure. Cet échec peut être la cause de l'échec de l'authentification, de l'échec de l'installation de la QoS ou de l'échec du contrôle d'admission. Cette information est signalée entre les entités de l'architecture avec l'AVP **Result-Code**.

3.4. Fonctionnement du protocole :

Le protocole UNISIG Diameter permet d'assurer l'authentification du mobile et, quand cela est nécessaire, l'autorisation et l'installation de la QoS ainsi que la mise à jour de la localisation. Tout son fonctionnement peut se résumer en quatre procédures déclenchées selon le Handover exécuté au mouvement du mobile. Dans le Handover Inter-Domaine, la procédure exécutée est la procédure Roaming ; Dans le Handover Inter-AR, la procédure exécutée est la procédure Macro-mobilité ; Dans le Handover Intra-AR, la procédure exécutée est la procédure Micro-mobilité. Les procédures Roaming et Macro-mobilité, quand elles sont exécutées, lancent la procédure de l'installation de la QoS dans le domaine. Dans cette partie nous expliquerons le fonctionnement du protocole UNISIG et l'interaction entre les unités fonctionnelles de l'architecture en détaillant le déroulement de chacune des quatre procédures.

3.4.1. Procédure Roaming :

Le MiMM diffuse des messages *EAP-Identity-Request* pour signaler sa présence et inviter les mobiles à s'authentifier. Quand le Mobile entre dans la zone d'un nouveau AP, il détecte le mouvement en interprétant les informations contenues dans le message *EAP-Identity-Request*. Si le MN ne reçoit aucun message au bout d'un temps, il diffuse le message *EAP-Start*, et le MiMM qui reçoit ce message lui répondra avec un message *EAP-Identity-Request*. Le mobile reçoit le message, l'interprète et conclut qu'il est sous un nouveau domaine et sous un nouveau routeur d'accès (sous réseau sans fil). Le MC réplique avec un message *EAP-Identity-Response* contenant l'identité du mobile qui est son NAI (Diagramme de séquence IV-1). Le MiMM reçoit le message par le port autorisé PAE du mécanisme du standard IEEE802.1X. Il extrait l'identité du mobile et l'encapsule dans un AVP **User-ID**. Il construit un message *USR* contenant AVP **Session-ID** et l'AVP **User-ID**. Il le transmet ensuite au MaMM installé dans le routeur d'accès auquel il est rattaché pour entamer l'authentification du mobile.

Le MaMM reçoit le message *USR* ; il cherche dans la table d'association MiM (Micro-Mobility) en utilisant l'identité du mobile l'association MiM correspondante à ce mobile. Le MaMM ne trouvant pas cette association MiM déduit qu'il s'agit soit d'une procédure de la Macro-mobilité dans laquelle une mise à jour de la localisation du mobile, une mise à jour de l'installation de la QoS et une authentification intra-domaine seront impliquées soit d'une procédure de Roaming dans laquelle une mise à jour de la localisation du mobile, une nouvelle installation de la QoS et une authentification inter-domaine seront impliquées. Le MaMM transmet alors le message *USR* au RMM du domaine en gardant le même AVP **Session-ID**. Le RMM cherche, à son tour, dans la table d'association MaM l'association MaM correspondante au mobile ; ne la trouvant pas, il déduit qu'il s'agit d'une procédure Roaming pour un mobile qui déclenche un Handover Inter-Domaine. Le V-RMM prépare alors le routage du message *USR* vers le RMM du domaine d'origine du mobile (H-RMM) grâce à l'identité du mobile pour procéder à l'authentification inter-domaine. Il envoie ensuite le message *USR* au H-RMM en gardant l'AVP **Session-ID** envoyé par le MaMM.

Quand il reçoit le message *USR*, le H-RMM génère, conformément à l'identité du mobile, un ensemble d'AV correspondant exclusivement au mobile faisant la requête. Le H-RMM extrait aussi de sa base de données le SLS du mobile selon le contrat fait avec l'utilisateur. Il construit ensuite un message *USA* contenant l'AVP **SLS** encapsulant le SLS du mobile, les AVP **AV** encapsulant les AV générés et l'AVP **Session-ID** du message *USR* reçu puis il l'envoie au V-RMM. Le V-RMM sauvegarde les données d'authentification et les données SLS reçues dans le message pour les utiliser dans l'authentification Intra-Domaine et l'installation de la QoS. Il sélectionne ensuite un groupe (ensemble) d'AV pour l'envoyer au MaMM qui a transmis auparavant le message *USR*. Le nombre d'AV qui pourrait y avoir dans le groupe envoyé peut dépendre du routeur d'accès et du nombre de points d'accès qu'il gère. Il peut être configuré manuellement au niveau du RMM ou dynamiquement par le MaMM pendant la phase d'échange de capacités du protocole Diameter. Le V-RMM encapsule ce groupe d'AV dans des AVP **AV** qui sont mis dans le message *USA*. Il ajoute par ailleurs dans le message l'AVP **Session-ID** correspondant et l'envoie au MaMM. Ce dernier extrait le groupe d'AV pour le sauvegarder afin de les utiliser dans l'authentification Intra-AR et sélectionne un AV pour retirer le compteur COMPT et le challenge RAND. Ces deux dernières valeurs sont mises respectivement dans l'AVP **RAND** et l'AVP **COMPT** puis le MaMM les envoie au MiMM dans un message *USA* en gardant toujours le même AVP **Session-ID**. Le MiMM enlève ensuite les deux valeurs RAND et COMPT, construit un message *EAP-Challenge-RegRq-Request* contenant ces deux valeurs et transmet le message au mobile.

Le MC du mobile recevant le défi construit le message Mobile IP *Registration-Request* avec l'adresse CoA sélectionnée et calcule la réponse RESP en utilisant la fonction de Hashage ayant comme paramètres le secret partagé avec le H-RMM et les deux valeurs RAND et COMPT reçues dans le message *EAP-Challenge-RegRq-Request*. Une fois le calcul fini, le MC construit un message *EAP-Challenge-RegRq-Response* contenant la réponse RESP et le message *Registration-Request* et l'envoie ensuite au MiMM. Celui-ci recevant le message toujours par le PAE construit un message *USR* contenant le même AVP **Session-Id** correspondant à la session Diameter se rapportant à ce mobile. Ce message qui est envoyé au MaMM transporte également l'AVP **RESP** qui encapsule la réponse RESP du mobile et l'AVP **Registration-Request** qui encapsule le message *Registration-Request*. Pour procéder à l'authentification du mobile, le MaMM extrait la réponse RESP de l'AVP **RESP** et la compare à la réponse XRESP correspondante (XRESP de l'AV sélectionné). Si le mobile est correctement authentifié - si RESP = XRESP -, le MaMM transmet au V-RMM le message *USR* avec seulement l'AVP **Registration-Request** envoyé par le MiMM.

Si le mobile s'est mal authentifié, le MaMM doit procéder à la ré-authentification en demandant une autre authentification ou arrêter la procédure de la mise à jour de la localisation. Quand il arrête la procédure de la mise à jour de la localisation, le MaMM envoie au MiMM un message *USA* contenant l'AVP **Result-Code** signalant l'échec de l'authentification et le MiMM fera suivre un message *EAP-Failure* au mobile. Le V-RMM, dans ce cas, annulera la procédure Roaming après l'écoulement d'un Timer prédéfini.

Dans le cas de l'utilisation d'une CCoA, le MaMM, après une authentification réussie, procure une adresse routable au mobile en utilisant le DHCP. Il ajoute ensuite la CCoA allouée dans le champ CoA du message *Registration-Request*, encapsule ce message dans un AVP **Registration-Request** et la procédure Roaming continue son exécution.

La réception du message *USR* appartenant à la session Diameter du mobile par le V-RMM avec un AVP **Registration-Request** confirme que le mobile s'est correctement authentifié au niveau du MaMM. A ce moment, le V-RMM exécute le contrôle d'admission et si le flot du mobile n'est pas admis, il va envoyer au MaMM un message *USR* avec un AVP **Result-Code** signalant l'échec de l'admission pour la QoS. Le MaMM transmet le message au MiM qui signale l'échec avec le message *EAP-Failure* au mobile.

Si le flot du mobile est admis, le V-RMM transmet au H-RMM le message *USR* contenant l'AVP **Registration-Request** et l'AVP **Session-ID** utilisé pour le mobile concerné. En parallèle le V-RMM entame la procédure d'installation d'état configuration passive de certains routeurs de bord et du routeur d'accès (section III.3.4.4). Se limiter à certains routeurs par lesquels le flot du mobile authentifié passera est fait dans le souci de diminuer les messages de signalisation et d'assurer une bonne gestion des ressources réseaux dans un domaine qui va desservir, en plus de ses propres abonnés, d'autres clients qui sont abonnés à d'autres administrations.

Quand le H-RMM reçoit le message *USR*, il décapsule l'AVP **Registration-Request** et le met dans un message *HAR* qui est envoyé au HA. Ce HA qui implémente un client Mobile IP Diameter va traiter la requête de mise à jour de la localisation du mobile. Il décapsule à son tour l'AVP **Registration-Request** pour extraire le message *Registration-Request* envoyé originairement par le mobile. Il traite ce message selon le mécanisme Mobile IP et met à jour la localisation ou crée l'association de mobilité dans la table d'association de mobilité. Il construit dans ces conditions le message *Registration-Reply*, l'encapsule dans l'AVP **Registration-Reply**, crée un message *HAA* contenant cet AVP et envoie enfin ce message au H-RMM. Le H-RMM construit un message *USA* qui contient l'AVP **Result-Code** signalant la

mise à jour de la localisation avec succès et envoie ensuite le message au V-RMM correspondant.

Pour ce V-RMM recevant de la part du H-RMM le message *USA* qui transporte l'AVP **Registration-Reply**, deux cas se présentent. La procédure d'installation passive des états de configuration de la QoS sur les routeurs de bord et le routeur d'accès n'est pas encore finie étant le premier cas ; dans ce cas, le V-RMM attendra la fin de la procédure de l'installation passive des configurations (section III.3.4.4) pour poursuivre la procédure Roaming. Le V-RMM a déjà fini l'installation passive sur tous les routeurs de bord et il n'attendait que le message *USA* du H-RMM qu'il vient de recevoir étant de deuxième cas qui est le plus probable. Dans les deux cas, quand les clients QoS Diameter confirment l'installation correcte des états de configuration de la QoS, le V-RMM crée une entrée de la table d'association MaM correspondante au mobile et construit un message *USA* contenant l'AVP **Registration-Reply** et l'AVP **Result-Code** signalant le succès de la mise à jour de la localisation et de l'installation de la QoS. En parallèle, il entame la procédure d'activation des installations des configurations sur les routeurs de bord et le routeur d'accès (section III.3.4.4) ; il envoie ensuite le message *USA* au MaMM.

A la réception du message *USA* transportant l'AVP **Registration-Reply**, le MaMM effectue, selon la valeur de l'AVP **Result-Code**, les traitements de Mobile IP et crée ensuite une entrée dans la table d'association MiM correspondante au mobile. Au même moment, le MaMM modifie l'AVP **Result-Code** en signalant avec cet AVP le succès de l'authentification, de l'installation de la configuration QoS et de la mise à jour de la localisation. Il transmet le message *USA* au MiMM. Celui-ci conclura, selon l'AVP **Result-Code** enlevé du message *USA*, si le mobile est correctement autorisé à accéder au réseau après son Handover. Il débloque le port fermé du standard IEEE 802.1X pour autoriser le trafic du mobile et envoie un message *EAP-Succes* au mobile. Le mobile, à ce stade, peut commencer sa communication à travers ce domaine visité. La réception et la transmission des paquets de ce mobile s'effectuent avec le mécanisme de routage Mobile IPv4.

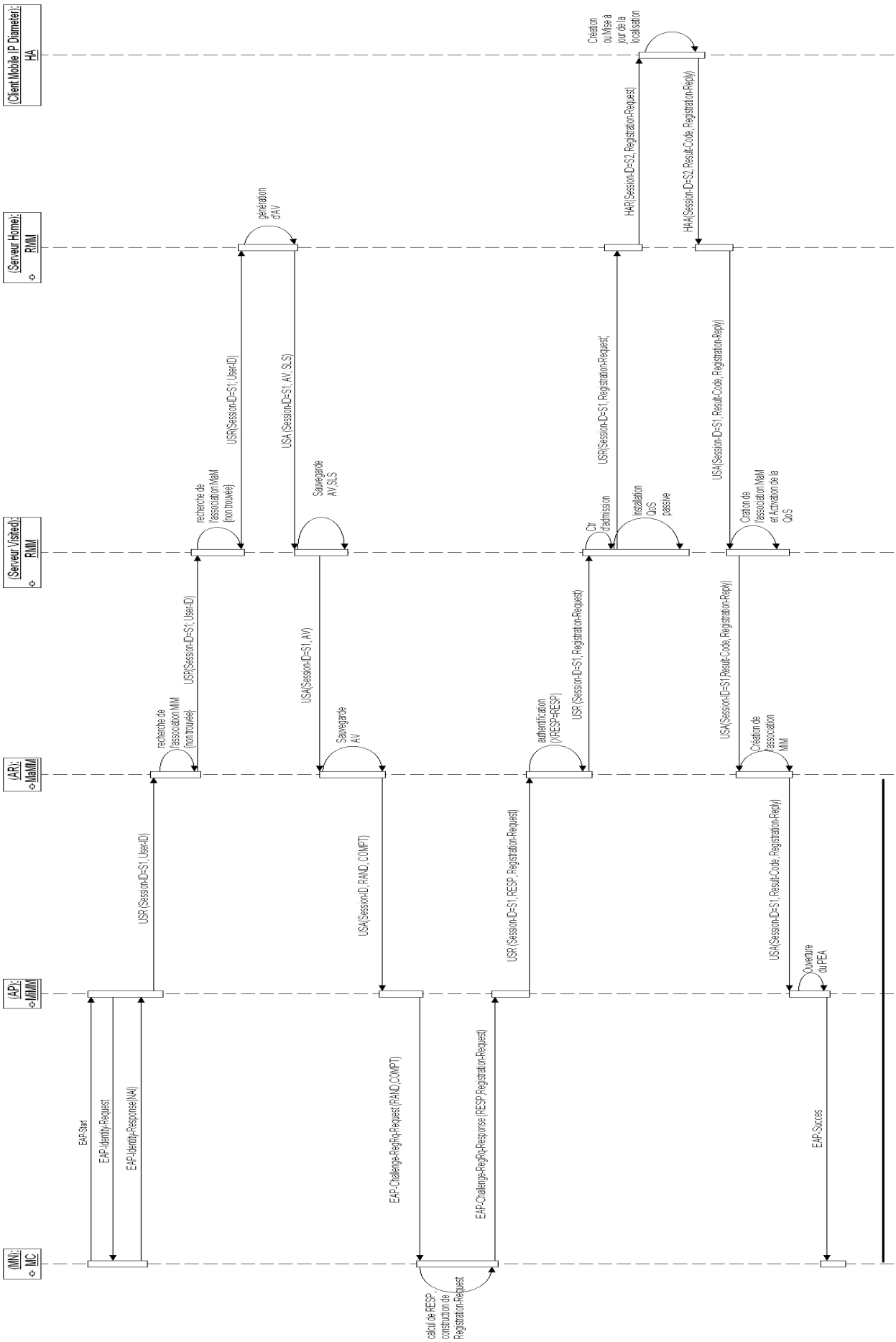


Diagramme de séquence IV-1 :

3.4.2. Procédure Micro-mobilité :

La procédure Micro-mobilité est courte car le Handover exécuté est un Handover Intra-AR et le mouvement du mobile reste à l'intérieur du sous réseau sans fil. Dans cette procédure, la mise à jour de la QoS et de la localisation ne sont pas indispensables ; le mobile n'effectuera qu'une authentification Intra-AR puisque le MaMM dispose à son niveau de toutes les informations d'authentification nécessaires.

La procédure Micro-mobilité commence quand le mobile détecte le changement de point d'accès en recevant *EAP-Identity-Request*. Le MC renvoie au MiMM l'identité du mobile dans un message *EAP-Identity-Response* (Diagramme de séquence IV-2). Le MiMM extrait ensuite cette identité, construit le message *USR* contenant l'AVP **User-ID** transportant l'NAI du mobile et transmet le message *USR* au MaMM pour entamer l'authentification. Ces étapes sont identiques à celles décrites dans le scénario précédent dans la procédure du Roaming.

Le MaMM reçoit le message *USR* et recherche dans la table d'association MiM l'association correspondante au mobile. Il la trouve car le mobile circule dans son sous réseau sans fil du routeur d'accès sur lequel le MaMM est installé. Il déduit que c'est l'exécution de la procédure Micro-mobilité qui nécessite seulement une authentification Intra-AR. Il sélectionne alors un AV du groupe sauvegardé afin prendre la valeur COMPT et le challenge RAND. Il met ensuite la valeur COMPT et le challenge RAND respectivement dans l'AVP **RAND** et l'AVP **COMPT** puis les envoie au MiMM dans un message *USA*.

Le MiMM extrait les deux valeurs RAND et CONT, constitue un message *EAP-Challenge-Request* contenant ces deux valeurs et transmet ce défi au Mobile ; le Mobile calcule la réponse RESP. Une fois le calcul terminé, le Mobile constitue un message *EAP-Challenge-Response* contenant la réponse RESP qui a été calculée ; il envoie ensuite le message au MiMM qui le reçoit par le port PAE. Le MiMM construit un message *USR* contenant l'AVP **RESP** et l'envoie au MaMM. Ce dernier, à la réception du message *USR*, extrait de l'AVP **RESP** la valeur de la réponse et procède à l'authentification du mobile. Il compare la réponse RESP à la réponse XRESP extraite de l'AV sélectionné (fournie auparavant par le H-RMM lors de la procédure de Roaming). Si le mobile est correctement authentifié, le MaMM construit un message *USA* contenant l'AVP **Result-Code** signalant le succès de l'authentification et envoie ce message au MiMM. Il met aussi à jour l'association MiM du mobile authentifié car il a changé de point d'accès. Si le mobile s'est mal authentifié, le MaMM doit procéder à la procédure de ré-authentification ou doit envoyer un message *USA* contenant l'AVP **Result-Code** signalant l'échec. Le MiMM, à la réception du message *USA*, extrait l'AVP **Result-Code**. Si l'authentification est un succès, il débloque le port fermé en autorisant le trafic du mobile à passer et envoie un message *EAP-Success*. L'authentification étant réussie, le mobile continuera l'émission et la réception des paquets IP selon le routage Mobile IP. Dans le cas de l'échec de l'authentification, le MiMM laisse le port bloqué et envoie au mobile un message *EAP-Echec*.

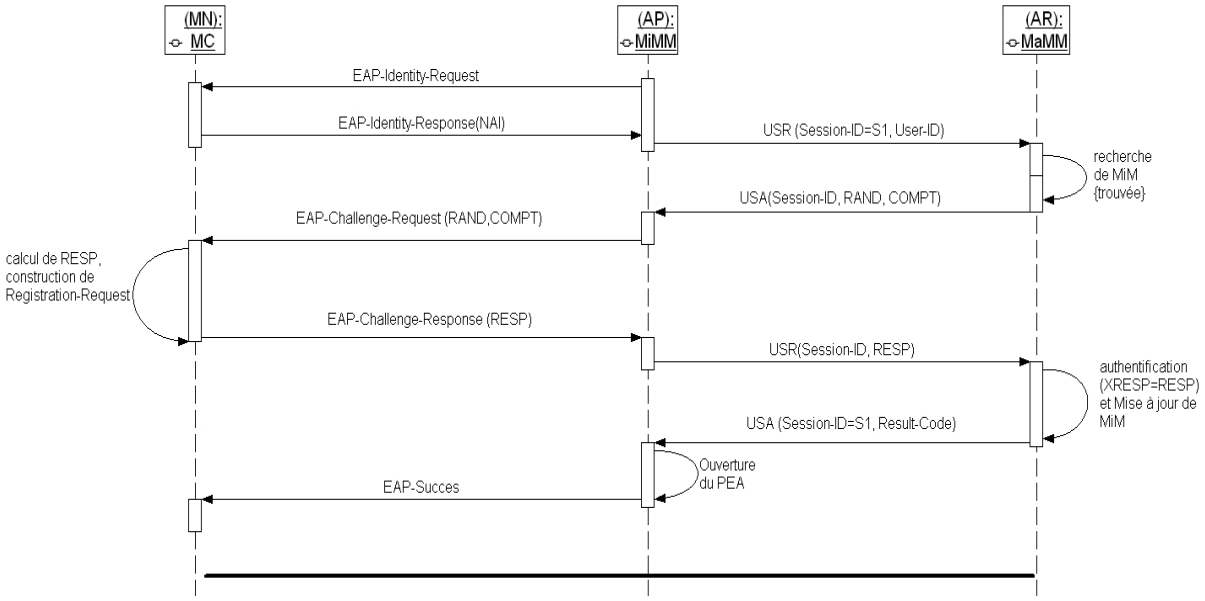


Diagramme de séquence IV-2 : Procédure Micro-mobilité.

3.4.3. Procédure Macro-mobilité :

La procédure Macro-mobilité est en grande partie similaire à la procédure Roaming. La différence majeure est que le mobile n'est authentifié qu'au niveau du domaine visité. Dans cette procédure, le mobile effectuera une mise à jour de la localisation de l'installation de la configuration de la QoS mais va exécuter seulement une authentification intra-domaine. L'opération de la mise à jour de la localisation reste la même avec celle de Roaming. Dans cette sous section la procédure de Macro-mobilité est décrite. Afin de permettre de suivre le déroulement de la procédure, quand la signalisation est semblable à la signalisation dans la procédure Roaming, la description est brève.

Comme dans le cas de la procédure Roaming, quand le Mobile entre dans un nouveau sous réseaux sans fil, il détecte son mouvement à la réception du message EAP-Identity-Request envoyé pas le MiMM. Le MC réplique avec un message *EAP-Identity-Response* contenant l'identité du mobile (Diagramme de séquence IV-3). Le MiMM extrait cette identité, construit un message *USR* et transmet ensuite ce message au MaMM pour entamer l'authentification du mobile. Le MaMM reçoit le message *USR* et cherche dans la table d'association MiM l'association MiM correspondante au mobile. Ne la trouvant pas, il déduit qu'il s'agit soit d'une procédure Macro-mobilité soit d'une procédure Roaming. Le MaMM transfère alors le message *USR* au V-RMM. Celui-ci recherche l'association MaM correspondante au mobile. En la trouvant, il conclut qu'il s'agit d'une procédure Macro-mobilité pour un mobile qui était déjà dans le domaine et qu'il a seulement changé de sous réseau sans fil (de routeur d'accès) en exécutant un Handover Inter-AR.

Le V-RMM entame la procédure d'authentification intra-domaine en sélectionnant un groupe d'AV envoyés déjà par le H-RMM lors de la première authentification du mobile au sein du domaine. Il envoie ce groupe d'AV au MaMM dans un message *USA* qu'il construit. Le MaMM sauvegarde le groupe d'AV, sélectionne un seul AV et retire les deux valeurs RAND et COMPT pour les envoyer dans un message *USA* au MiMM. Ce MiMM constitue un message *EAP-Challenge-RegRq-Request* contenant ces deux valeurs et transmet le message au mobile. Le MC construit le message *Registration-Request* et calcule la réponse RESP comme dans le cas de la procédure Roaming. Il envoie ensuite le message *Registration-Request* et la réponse RESP au MiMM dans un message *EAP-Challenge-RegRq-Response*. Le

MiMM construit un message *USR* qui est envoyé au MaMM et qui transporte également l'AVP **RESP** et l'AVP **Registration-Request**.

Le MaMM procède alors à l'authentification du mobile en comparant la réponse RESP avec la réponse XRESP correspondante. Si le mobile est correctement authentifié, le MaMM retransmet au V-RMM le message *USR* avec l'AVP **Registration-Request** envoyé par le mobile. Sinon il doit déclencher soit la ré-authentification soit l'arrêt de la procédure d'authentification et celle de la mise à jour de la localisation.

Dans le cas de l'utilisation d'une CCoA, le MaMM, après une authentification réussie, procure une adresse routable au mobile en utilisant le DHCP. Dans ce cas, le MaMM ajoute la CCoA allouée dans le champ CoA du message *Registration-Request*, encapsule ce message dans un AVP **Registration-Request** et la procédure Macro-mobilité continue son exécution.

Après avoir exécuté le contrôle d'admission, le V-RMM transmet le message *USR* qu'il a construit au H-RMM du mobile. En parallèle le V-RMM entame la procédure d'installation d'état configuration passive de certains routeurs de bord et du routeur d'accès (section III.3.4.4). Dans la procédure de Macro-mobilité, le V-RMM doit entamer en priorité l'effacement des états de configuration de la QoS (section III.3.4.4) notamment, en toute urgence, l'état installé sur l'ancien routeur d'accès du mobile (l'ancien MaMM). Ceci permettra de libérer le plus tôt possible des ressources qui sont peut être indispensable pour l'admission des Handover des autres mobiles.

Quand il reçoit le message *USR*, le H-RMM décapsule l'AVP **Registration-Request** et le met dans un message *HAR* qui est envoyé au HA. Ce HA traite alors la requête de mise à jour de la localisation du mobile et met à jour l'association de mobilité dans la table d'association de mobilité. Il construit dans ces conditions le message *Registration-Reply*, l'encapsule dans l'AVP **Registration-Reply** et l'envoie au H-RMM dans un message *HAA*. Le H-RMM envoie ensuite au V-RMM un message *USA* qui contient l'AVP **Registration-Reply** et l'AVP **Result-Code**. Quand le V-RMM reçoit ce message *USA*, il procède à la configuration QoS des routeurs de bord et du routeur d'accès exactement comme dans le cas de la procédure de Roaming. Il met ensuite à jour l'entrée de la table d'association MaM correspondante au mobile et transmet le message *USA* contenant l'AVP **Registration-Reply** et l'AVP **Result-Code** signalant le succès de la mise à jour de la localisation et de l'installation de la QoS.

A la réception du message *USA*, le MaMM effectue les traitements de Mobile IP et crée ensuite une entrée dans la table d'association MiM correspondante au mobile. Au même moment, le MaMM modifie l'AVP **Result-Code** en signalant avec cet AVP le succès de l'authentification, de l'installation de la configuration QoS et de la mise à jour de la localisation. Il transmet le message *USA* au MiMM qui déploque le port PAE pour autoriser le trafic du mobile et envoie un message *EAP-Succes* au mobile. Le mobile commence à émettre et à recevoir des paquets IP selon le routage mobile IP.

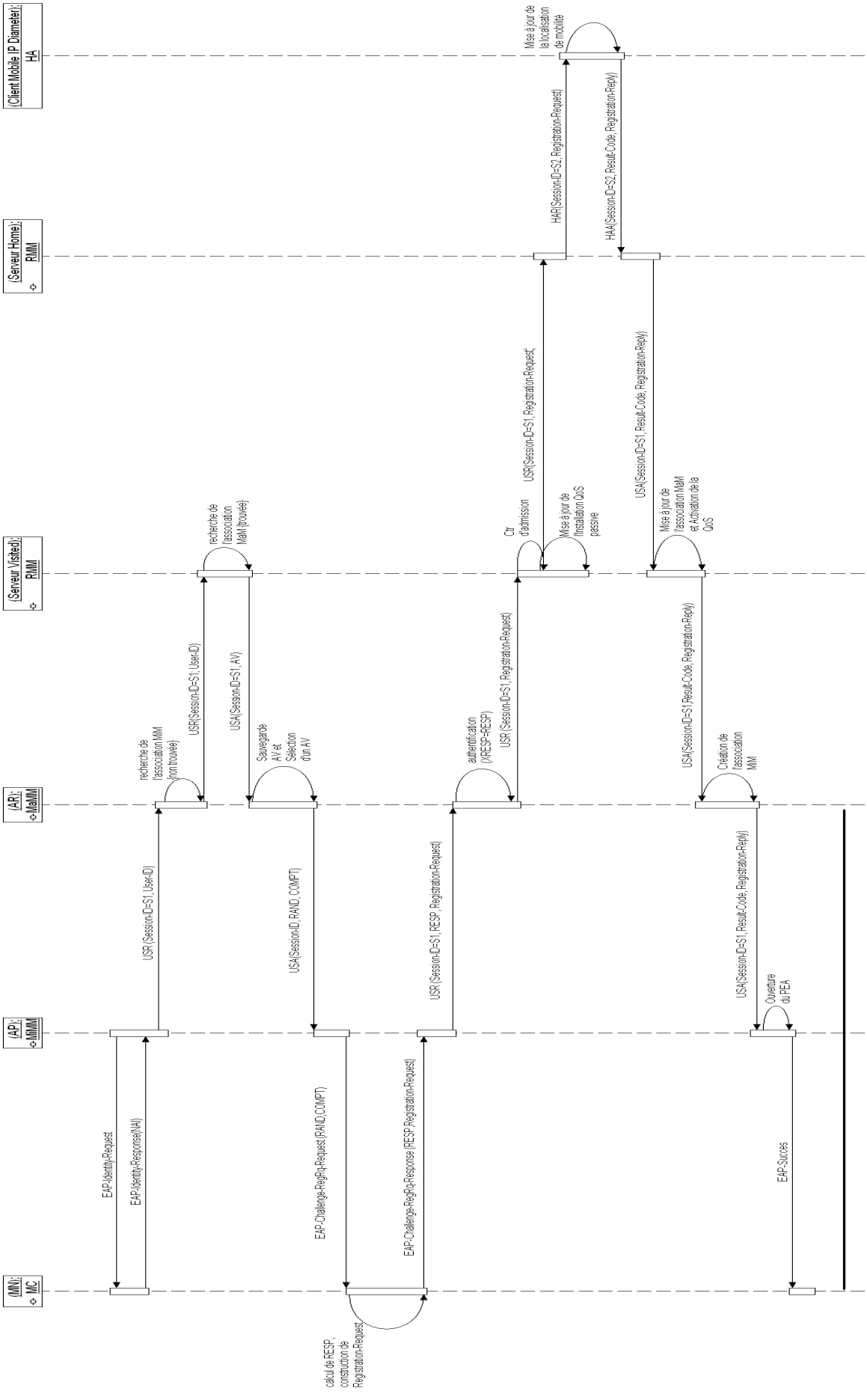


Diagramme de séquence IV-3 :
mme de séquence IV-3 :

Remarques :

Si, dans la procédure Macro-mobilité, le nombre d'AV envoyés au V-RMM par le H-RMM s'avère insuffisant après plusieurs authentifications, le V-RMM n'aura plus d'AV à la prochaine authentification. Dans ce cas, le V-RMM envoie au H-RMM une requête avec l'AVP **Auth-Request-Type** encapsulant la constante signalant que c'est seulement une authentification. Si, dans la procédure Micro-mobilité, le nombre d'AV envoyés au MaMM par le V-RMM s'avère insuffisant après plusieurs authentifications, le V-RMM n'aura plus d'AV à la prochaine authentification. Dans ce cas, le MaMM envoie au V-RMM une requête avec l'AVP **Auth-Request-Type** encapsulant la constante signalant que c'est seulement une authentification.

Dans le cas de la Macro-mobilité où seulement la mise à jour de la localisation est indispensable puisque l'authentification et l'installation des configurations de la QoS au niveau du domaine ont été faites et elles ne changent pas, l'AVP **Auth-Request-Type** encapsulant la constante signalant seulement la mise à jour de la localisation est envoyé au MaMM par le V-RMM.

3.4.4. Procédure de configuration de la QoS :

La procédure de configuration de la QoS s'inspire du protocole COPS sur une architecture DiffServ. Un client QoS Diameter est installé sur chaque routeur de bord. Il reçoit des requêtes de configuration de la part du serveur UNISIG Diameter installé sur le RMM qui gère les ressources du domaine et qui gère aussi la configuration des ressources sur les routeurs d'accès. Le RMM est chargé d'appliquer le contrôle d'admission des flots des utilisateurs pour limiter leurs nombres en cas de charge et, de ce fait, il ne doit pas dégrader la QoS des autres mobiles. Pour cela, le RMM gère une base de données (Figure IV-2) sauvegardant toutes les informations de ressources du domaine. Cette base de données est mise à jour à chaque installation admise. A ce titre, le RMM a l'autorité d'accepter ou de rejeter (d'arrêter les procédures Roaming et Macro-mobilité) une requête avant d'envoyer le message *USR* au H-RMM pour la mise à jour de la localisation. Cette décision est faite selon la disponibilité des ressources ou/et la priorité de l'utilisateur du mobile. Ainsi, selon la gestion du domaine, le RMM peut réserver certaines ressources pour les mobiles en Handover sur chaque routeur d'accès.

Le RMM peut faire requête de deux types de configuration. La première dite passive et la deuxième dite active (Diagramme de séquence IV-4). Une configuration passive peut, dans le cas du déroulement correct des opérations (authentification, mobilité, Handover) devenir configuration active. Tant qu'une configuration est passive, le routeur du bord saura que le flot du mobile faisant demande ne va pas utiliser ces ressources avant que la configuration ne soit activée. Autrement dit, le mobile ne transmettra pas de paquets mais la réservation de ressources est en cours. De ce fait, le routeur ne tient pas compte de cette configuration dans la gestion momentanée de la transmission des paquets de l'utilisateur concerné. Le routeur de bord efface cette configuration après l'écoulement d'un Timer si elle n'est pas activée. Ce principe est exploité dans cette architecture dans le but de ne pas monopoliser des ressources avant la mise à jour de la localisation du mobile. La configuration de la QoS est rapidement installée en l'activant dès que la mise à jour de la localisation est confirmée par le H-RMM.

L'installation passive de la configuration est déclenchée dès l'authentification du mobile et avant la mise à jour de la localisation soit dans la procédure Roaming soit dans la procédure Macro-mobilité. A la réception par le V-RMM du message *USR* qui est envoyé par le MaMM et qui transporte l'AVP **Registration-Request** afin d'exécuter la mise à jour de la localisation, le V-RMM entame le contrôle d'admission. Le V-RMM exécute ensuite la procédure de la configuration passive des routeurs de bord et du routeur d'accès concernés (Diagramme de

séquence IV-4 : Installation passive de la QoS). Il envoie au client QoS Diameter un message *PIR* transportant l'AVP **Session-ID**, l'AVP **Session-QoS-ID**, l'AVP **Policy**, l'AVP **QoS-Install** et l'AVP **Timer**. La valeur de l'AVP **Session-ID** est la même utilisée dans la session Diameter concernant le mobile authentifié et qui change à chaque Handover Inter-AR. L'AVP **Session-QoS-ID** a la valeur qui identifie la session de la QoS du mobile et le Handle de l'état de configuration de la QoS. Cet identificateur est unique pour chaque utilisateur connecté au domaine et reste le même pendant tout le "séjour" du mobile dans le domaine. L'AVP **Policy** transporte les règles à installer représentées sous forme PIB (Policy Information Base[40]). Ces règles sont extraites du SLS du mobile ; ce SLS est envoyé par le H-RMM lors de l'authentification. L'AVP **QoS-Install** contient la valeur **PASSIVE** pour indiquer une installation passive et l'AVP **Timer** mentionne la durée de cette installation. Quand le client QoS Diameter reçoit le message *PIR*, il crée un état identifié par la valeur de l'AVP **Session-QoS-ID** et installe les règles transmises dans l'AVP **Policy**. Après cela, le résultat de l'installation est signalé par l'AVP **Result-Code** transporté par le message *PIA* envoyé par le client QoS Diameter au RMM. Ce RMM met à jour sa base d'informations de disponibilité de ressources afin de prendre en considération cette admission du flot du mobile surtout pour l'admission des Handover vers les routeurs d'accès.

Quand le V-RMM reçoit le message *USA* transportant l'AVP **Registration-Reply** et l'AVP **Result-Code** signalant la réussite de la mise à jour de la localisation, il construit un message *PIR* encapsulant l'AVP **Session-ID**, l'AVP **Session-QoS-ID**, l'AVP **Timer** et l'AVP **QoS-Install** avec la constante **ACTIVE** afin d'activer l'état de configuration correspondant (Diagramme de séquence IV-4 : *Activation de l'installation de la QoS*). Le client QoS Diameter active la configuration et saura qu'il va recevoir des paquets du mobile. Il construit alors un message *PIA* transportant l'AVP **Result-Code** signalant le résultat de l'activation et l'envoie au V-RMM. L'installation de l'état de la configuration, dans les routeurs de bord, est maintenue pendant un temps mentionné dans l'AVP **Timer**. Elle doit être rafraîchie par le RMM avec un message *PIR* transportant les mêmes AVP et référencé avec la valeur **Session-QoS-ID**. Le client répondra avec un message *PIA* contenant toujours un AVP **Result-Code** (Diagramme de séquence IV-4 : *Rafraîchissement de la configuration*).

A tout moment, un SLS peut être changé et dans ce cas, le V-RMM reçoit un nouveau SLS de la part du H-RMM ou par une configuration manuelle. Le V-RMM envoie un message *PIR* avec un AVP **Policy** encapsulant de nouvelles règles et un AVP **Session-QoS-ID** référençant l'état précédemment installé (Diagramme de séquence IV-4 : *Mise à jour de la configuration*). Le client réinstalle les nouvelles règles et renvoie un message *PIA* avec l'AVP **Result-Code** signalant le résultat. Le RMM met ensuite à jour sa base d'informations de disponibilité de ressources. Ce scénario pourrait être exécuté aussi dans le cas où la gestion par politique du domaine changerait et de nouvelles règles seraient définies au sein du domaine.

A la fin d'une session QoS ou parce qu'il y a un changement de la route issue de la procédure Macro-mobilité, le V-RMM désinstalle les états de configuration de tout ou de certains routeurs de bord en envoyant un message *PIR* à chacun d'eux (Diagramme de séquence IV-4 : *Désinstallation de la configuration*). Ce message *PIR* transporte l'AVP **Session-QoS-ID** pour référencer l'état à désinstaller et transporte aussi l'AVP **QoS-Install** encapsulant la constante **UNINSTALL** pour ordonner l'effacement de l'état. Le client QoS Diameter, à la réception du message, efface l'état et construit un message *PIA* contenant l'AVP **Session-QoS-ID** et l'AV **Result-Code**. Le RMM met ensuite à jour sa base d'informations de disponibilité de ressources pour les routeurs correspondants. Ce scénario est aussi utilisé pour libérer les ressources au niveau de l'ancien routeur d'accès dès que l'authentification du mobile est accomplie. Dans tous les cas, si le RMM n'a pas envoyé un message *PIR* pour effacer les états ou pour les rafraîchir, ces états seront effacés par les clients QoS Diameter après l'écoulement du Timer.

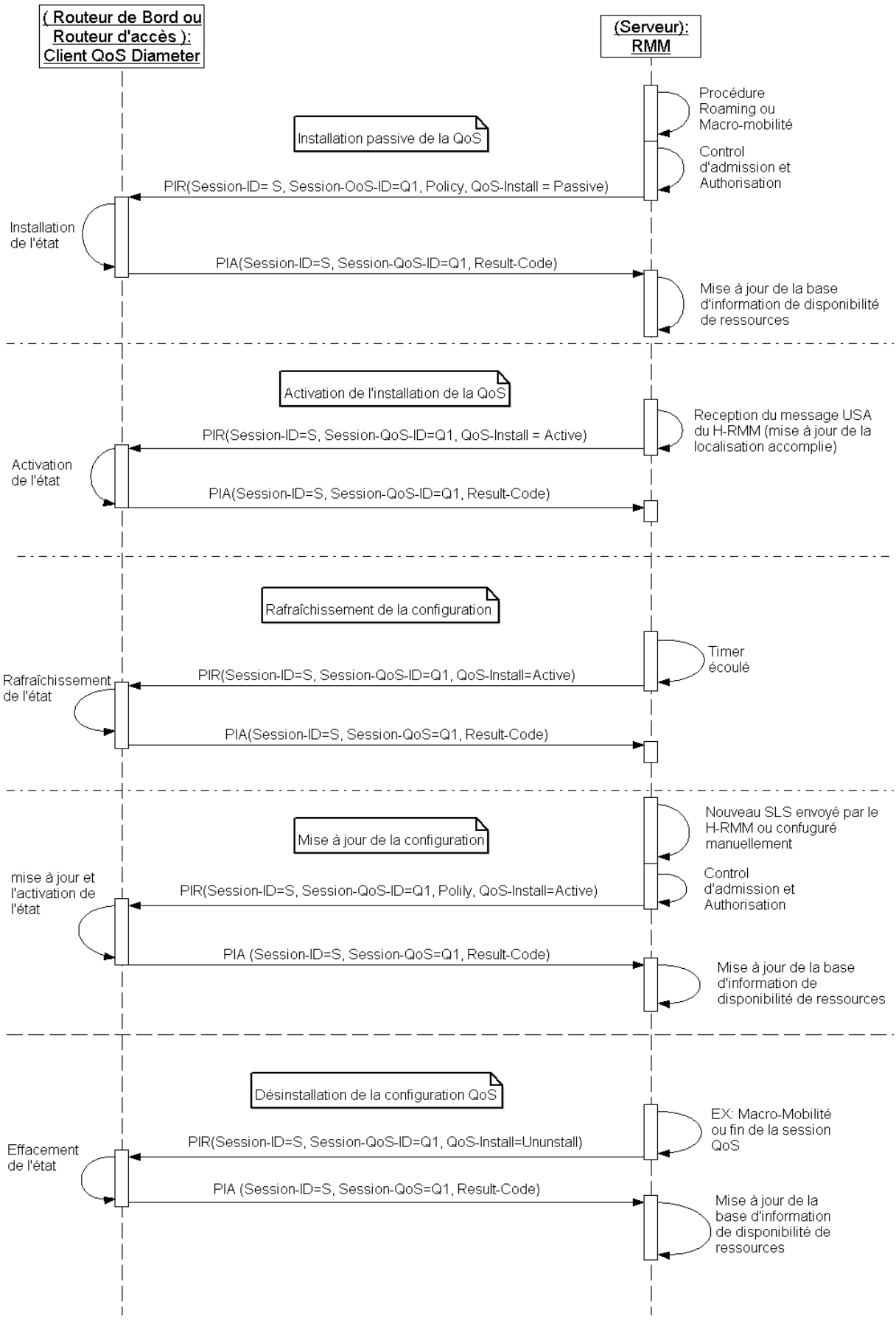


Diagramme de séquence IV-4 : gestion de la QoS dans l'architecture UNISIG.

4. Conclusion :

Ce chapitre décrit l'architecture unifiée qui est proposé dans notre travail. Elle est constituée de trois parties : la première est l'infrastructure matérielle qui met en place les réseaux, la deuxième est l'ensemble des entités fonctionnelles qui communiquent entre elles pour accomplir la signalisation et la troisième est le protocole de signalisation qui unifie les signalisations de la AAA, la gestion de la QoS et la gestion de la mobilité.

Les entités fonctionnelles, tout comme l'infrastructure sur laquelle elles s'installent, suivent une architecture hiérarchique. Le premier niveau de la hiérarchie se compose des MC installés sur les mobiles. Le deuxième niveau de la hiérarchie se compose des MiMM installés sur les AP. Le troisième niveau se compose des MaMM et des clients QoS Diameter installés sur les routeurs. Le quatrième et dernier niveau se compose des RMM installés sur les serveurs.

Le protocole UNISIG Diameter qui permet la signalisation entre les entités fonctionnelles s'exécute sous forme de trois procédures pour accomplir les taches assurées par les protocoles de la AAA, la gestion de la QoS et la gestion de la mobilité. La première de ces procédures est celle du Roaming qui se déclenche au moment d'un Handover Inter-Domaine autrement dit au moment d'une nouvelle entrée du mobile dans un domaine. Cette procédure accomplit une authentification Inter-Domaine dans laquelle les données d'authentification et les données de la QoS sont rapatriées au RMM visité pour la première authentification et installation de la QoS et pour celles qui se suivent pendant le séjour du mobile dans le domaine. La contribution principale de cette procédure dans la réduction de l'interruption due au Handover consiste en la réduction de la signalisation. La réduction de la signalisation est menée à bien grâce au processus suivant : au moment de la réponse de l'authentification par le mobile, ce mobile envoie en même temps la requête de mise à jour de la localisation et, quand ce mobile est correctement authentifié, l'installation de la QoS est faite en parallèle avec celle de la mise à jour de la localisation. De ce fait quand le mobile reçoit la réponse de l'authentification, il lui est possible de commencer directement la communication. La deuxième procédure est la procédure Micro-mobilité qui est exécutée dans le cas d'un Handover Intra-AR. Cette procédure accomplit seulement une authentification Intra-AR au niveau du sous réseau sans fil et une mise à jour de l'association de la MiM. Cette authentification est faite très rapidement puisque le routeur d'accès qui gère le réseau sans fil dispose à son niveau de toutes les informations de l'authentification indispensables et aucune mise à jour de l'installation de la QoS ni de localisation est nécessaire. La dernière procédure est celle de la Macro-mobilité qui est exécutée dans le cas d'un Handover Inter-AR quand le mobile change de routeur d'accès. Dans cette procédure, une mise à jour de l'installation de la QoS et de la localisation est nécessaire. L'authentification effectuée est toutefois une authentification Intra-Domaine seulement au niveau du serveur du domaine sans la sollicitation du serveur d'origine ; cela réduit le temps de l'authentification. De plus, tout comme dans la procédure Roaming, la mise à jour de la localisation est déclenchée dès l'authentification du mobile et en parallèle la mise à jour de l'installation de la QoS est exécutée.

Le but principal du travail présenté dans ce chapitre est de réduire la taille et le nombre de messages de signalisation ainsi que le temps d'exécution du Handover afin d'améliorer la QoS dans un réseau mobile. Vu sous un autre aspect, on peut constater que le nombre et l'emplacement des entités fonctionnelles de cette architecture influence son efficacité et l'amélioration du service. Pour cette architecture, ce qui reste à faire pour son exploitation, est la valider afin de l'optimiser.

Chapitre V

Simulation de l'architecture unifiée UNISIG

1. Introduction :

Après avoir proposé un protocole de signalisation unifiée et son architecture appelée UNISIG, nous avons essayé d'analyser le comportement du protocole UNISIG par rapport aux protocoles existants : protocoles de signalisations MIPv4 et protocole d'authentification avec gestion de la QoS exécutés indépendamment. Le premier point concerne la découverte des limites de la nouvelle architecture et du nouveau protocole ainsi que les avantages et les inconvénients des deux types de protocoles à savoir UNISIG et les protocoles non unifiés existants ; il s'agit d'une analyse de l'architecture en fonction de différentes surcharges des abonnés. Le second point est l'analyse du comportement des deux catégories de protocoles sur la même architecture et savoir quel protocole permet de mieux préparer les ressources de Qualité de Service en mobilité.

Dans ce chapitre, on commencera par exposer le simulateur de réseaux NS utilisé pour simuler les protocoles. On décrira ensuite l'architecture de test sur laquelle le protocole est simulé. Après cela, on définira les paramètres utilisés et on exposera enfin les différentes simulations effectuées et les interprétations de ses résultats.

2. Simulateur de réseaux (Network Simulator NS) :

NS [70] est un outil logiciel de simulation de réseaux informatiques. Il est principalement conçu avec les idées de la conception par objets, de la réutilisabilité du code et de la modularité. Il est devenu aujourd'hui un outil de référence pour les chercheurs du domaine du réseau. Ils peuvent ainsi partager leurs efforts et échanger leurs résultats de simulations. NS est un logiciel dans le domaine public disponible sur Internet et son utilisation est gratuite. Ce logiciel est exécutable tant sous Unix (Linux) que sous Windows.

2.1. Description de NS :

Le simulateur NS actuel est particulièrement bien adapté aux réseaux à commutation de paquets et à la réalisation de simulations de petites et moyennes tailles. Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage, des protocoles de transport, des protocoles de session, des protocoles de réservation et des protocoles d'application. De plus, le simulateur possède déjà une palette de systèmes de transmission, d'ordonnanceurs et

de politiques de gestion de files d'attentes pour effectuer des études de contrôle de congestion. Prises ensembles, ces capacités ouvrent le champ à l'étude de nouveaux mécanismes au niveau des différentes couches de l'architecture réseau TCP/IP.

NS est bâti autour d'un langage de programmation appelé Tcl dont il est une extension. Du point de vue de l'utilisateur, la mise en œuvre de ce simulateur se fait via une étape de programmation qui décrit la topologie du réseau et le comportement de ses composants ; vient ensuite l'étape de simulation proprement dite et enfin l'interprétation des résultats. Cette dernière étape peut être prise en charge par un outil annexe intégré à NS, appelé NAM lequel permet une visualisation et une analyse des éléments simulés.

NS est écrit en C++ avec une interface textuelle (ou shell) qui utilise le langage OTcl (Object Tool Cammand Langage). L'OTcl est une extension objet au langage de commande Tcl. Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation. Il sert à définir les classes pour reprendre la terminologie objet. Quant au langage OTcl, il fournit un moyen flexible et puissant de contrôle de la simulation comme le déclenchement d'événements, la configuration du réseau, la collecte de statistiques, etc. L'application NS se compose de deux éléments fonctionnels : un interpréteur et un moteur de simulation. Au moyen de l'interpréteur, l'utilisateur est capable de créer le modèle de simulation : ce qui revient à assembler les différents composants nécessaires à l'étude. Les composants du modèle de simulation sont appelés objets et le moteur de simulation, quant à lui, effectue les calculs applicables au modèle préalablement construit par l'utilisateur via l'interpréteur.

NS permet de simuler aussi bien des réseaux filaires que des réseaux mobiles sans fil de type infrastructure ou de type ad hoc. Les principaux composants de réseau qui permettent de définir l'architecture et la topologie d'un modèle sont les nœuds et les liens. Un protocole est implémenté sous forme d'agent qui peut être installé sur un nœud. Une entité appelée paquet est transportée d'un nœud à un autre à travers les liens. Cette entité véhicule différentes informations qui seront utiles pour générer les résultats de la simulation. Le résultat de la simulation est généré sous forme d'un fichier texte structuré en ligne et colonne. Chaque ligne porte les informations d'un paquet (taille, type…) et le type d'événement déclenché par ce paquet (tropped, received, …).

2.2. Nouveaux modules implémentés :

Le protocole UNISIG ainsi que les protocoles d'authentification, de la gestion de la mobilité et de la gestion de la QoS sont implémentés dans NS sous forme d'agents. Le Tableau 1 et le Tableau 2 décrivent les agents implémentés dans le simulateur NS. Les détails de l'implémentation du protocole UNISIG et les échanges des messages entres les agents sont exposés en annexe Annexe E :.

Agent dans NS	Comportement implémenté	Installé sur le composant
MCAgent	Mobility-Client	Mobile Node (MN)
MiMMAgent	Micro-Mobilité Manager	Access Point (AP)
MaMMAgent	Macro-Mobilité Manager	Access Router (AR)
RMMAgent	Roaming Manager	Server du domaine
QoSAgent	QoS client	DiffServ edge router

Tableau 1 : Agents implémentés pour simuler UNISIG.

Agent dans NS	Comportement implémenté	Installé sur le composant
SUPPAgent	l'authentifiant	Mobile Node (MN)
AUTHAgent	l'authentificateur (Client Diameter)	Access Point /Access Router
DIAMSRVAgent	serveur d'authentification Diameter	Server du domaine
QoSAgent	QoS client	DiffServ edge router

Tableau 2 : Agents implémentés pour simuler les protocoles indépendants.

En plus des fichiers générés habituellement par NS, un autre fichier y a été introduit. Ce fichier définit pour chaque procédure liée au mouvement des mobiles une ligne qui décrit le numéro de séquence de la procédure, l'identifiant du MN qui l'a exécutée, son moment début, son moment de fin et son type. Ces fichiers servent à l'analyse des durées des procédures des deux protocoles.

Une particularité a été ajoutée à cette implémentation. Quand le mobile envoie le message de mise à jour de la localisation *Registration-Request* au HA, celui-ci authentifie le message chez le serveur d'authentification avant de faire la mise à jour de la localisation et d'envoyer le message *Registration-Response* au mobile demandeur. Dans notre implémentation, cela n'est pas exécuté car le point d'accès qui utilise le port PAE ne laisse passer que des messages EAP et pas les messages des autres protocoles. Ce qui signifie que même les messages du protocole son filtrés ; ainsi, le mobile ne peut envoyer *Registration-Request* qu'après son authentification au niveau du point d'accès. Cela rend l'authentification du message au niveau du HA facultative. Ce schéma réduit les durées d'exécution des procédures de MIPv4 et de UNISIG mais du moment que l'étude des protocoles se fait dans les mêmes conditions cela ne va pas fausser l'analyse. Ce point est considéré comme une nouveauté introduite dans la gestion de la mobilité car l'opportunité d'utiliser le protocole IEEE 802.1X permet d'exploiter ce principe.

3. Architecture de test :

Cette architecture de test est une représentation de l'ensemble des objets d'un réseau (mobile, point d'accès, routeur d'accès), comme étant une réalisation d'une famille de processus de poisson ponctuels dans le plan. Le point de départ de la construction de l'architecture est une répartition des mobiles dans le plan selon un processus ponctuel de Poisson qui traduit de manière simple la répartition irrégulière de ces mobiles. En effet, le processus d'implantation des points d'accès est déterminé par celui de la répartition des abonnés à desservir dans une optique d'optimisation et d'adaptation de l'architecture du réseau.

Les mobiles, les points d'accès et les routeurs d'accès sont représentés par leurs coordonnées dans le plan R^2. On considère ensuite trois processus ponctuels de Poisson homogènes et stochastiquement indépendants *P0*, *P1* et *P2* dans le plan. Chaque réalisation d'un tel processus consiste en un ensemble aléatoire de points représentant les mobiles (pour le processus *P0*), les points d'accès (pour le processus *P1*) et les routeurs d'accès (pour le processus *P2*). A chaque routeurs d'accès, on associe sa zone de raccordement qui est constituée des points d'accès qui sont les plus proches de ce routeurs d'accès que de tout autres routeurs d'accès. Les paramètres de localisation des mobiles, des points d'accès et des routeurs d'accès sont donc réduits aux mesures λ_0, λ_1 et λ_2 d'intensité de ces processus. On suppose que λ_0 et λ_1 ainsi que λ_1 et λ_2 sont proportionnelles, c'est-à-dire $\lambda_0 = \alpha_1 * \lambda_1$ et

$\lambda_1 = \alpha_2 * \lambda_2$ avec $\alpha_1 > 1$ et $\alpha_2 > 1$; ce qui traduit le souci d'implanter les des points d'accès et les routeurs d'accès la où ils sont nécessaires.

Pour simuler le protocole UNISIG parallèlement à la signalisation indépendante, les objets du réseau d'accès sont associés comme suit : le niveau 0 correspond aux mobiles (MN), le niveau 1 correspond aux MiMM pour UNISIG et aux AP pour la signalisation indépendante et le niveau 2 correspond aux MaMM pour UNISIG et aux AR pour la signalisation indépendante. On associe donc aux MN un processus de poisson d'intensités λ_0, et aux MiMM (respectivement aux AP) un processus de poisson d'intensités $\lambda_1 = \alpha_1 * \lambda_0$ et aux MaMM (respectivement aux AR) un processus de poisson d'intensités $\lambda_2 = \alpha_2 * \lambda_1$. Chaque MiMM (respectivement les AP) est ensuite relié au MaMM (respectivement les AR) le plus proche. Les MN se connectent automatiquement au point d'accès le plus proche lors du mouvement.

L'architecture de test utilisée dans notre simulation consiste en une modélisation aléatoire dans la limite des aptitudes offertes par NS. On décrit dans la sous-section l'architecture de test avec des paramètres génériques. Par la suite, en fonction ce qu'on veut étudier, on précisera les paramètres à modifier ou à faire varier pour exécuter la simulation. Le modèle aléatoire exposé dans ce chapitre se base sur une répartition des objets de réseaux sur un plan ; chaque objet a ses coordonnées dans R^2. Comme dans les réseaux sans fil, la distance qui sépare les point d'accès et les mobiles est importante pour déterminer si les nœuds peuvent recevoir les paquets des autres nœuds ; la modélisation doit se faire à trois (3) dimensions. Puisque NS permet de faire la modélisation des réseaux mobiles seulement sur deux (2) dimensions, on doit se limiter à la distribution des objets de réseaux sur une dimension. Les processus ponctuels utilisés dans la simulation sont donc des distributions sur une ligne.

3.1. Infrastructure :

L'infrastructure de l'architecture de test se compose d'un backbone qui modélise l'Internet et de deux domaines (D1 et D2). Chaque domaine est constitué d'un réseau DiffServ et d'un ensemble d'AP (Figure V-1). On lie aussi un ensemble de Nœuds Correspondants (CN) au backbone.

a. Le backbone est un ensemble de nœuds reliés entre eux par des liaisons duplexes. Il est relié aux deux domaines et à un ensemble de nœuds correspondants CN.

b. Un domaine est constitué d'un ensemble de routeurs de bords DiffServ, d'un ensemble de MaMM (ou AP) qui sont aussi des routeurs de bords DiffServ, d'un ensemble de routeurs de cœurs de type DiffServ, d'un serveur RMM unique et d'un ensemble de MiMM (ou AP). Les routeurs de cœurs sont connectés entre eux pour constituer un réseau de cœur dans le domaine. Les routeurs de bords sont reliés au réseau backbone avec des liaisons duplexes symétriques. Le serveur RMM, les MaMM et les routeurs de bords sont reliés au réseau de routeurs de cœurs avec des liaisons simplexes. Le réseau de routeurs de cœurs est relié aux routeurs de bord avec des liaisons simplexes. Chaque MaMM est relié à un ensemble de MiMM avec des liaisons duplexes.

c. Les MN sont en mouvement sous la zone de couverture des MiMM. Chaque MN a son Home domain (D1 ou D2), son Home RMM et son Home MaMM. Quand il se déplace vers l'autre domaine, il devient sous Visited domain.

d. Les CN sont des nœuds connectés au backbone avec des liaisons duplexes. Leur nombre est variable selon le nombre de MaMM des deux domaines et ils sont répartis en groupes ayant approximativement un nombre identique de CN. Chaque groupe CN est relié à un nœud du backbone.

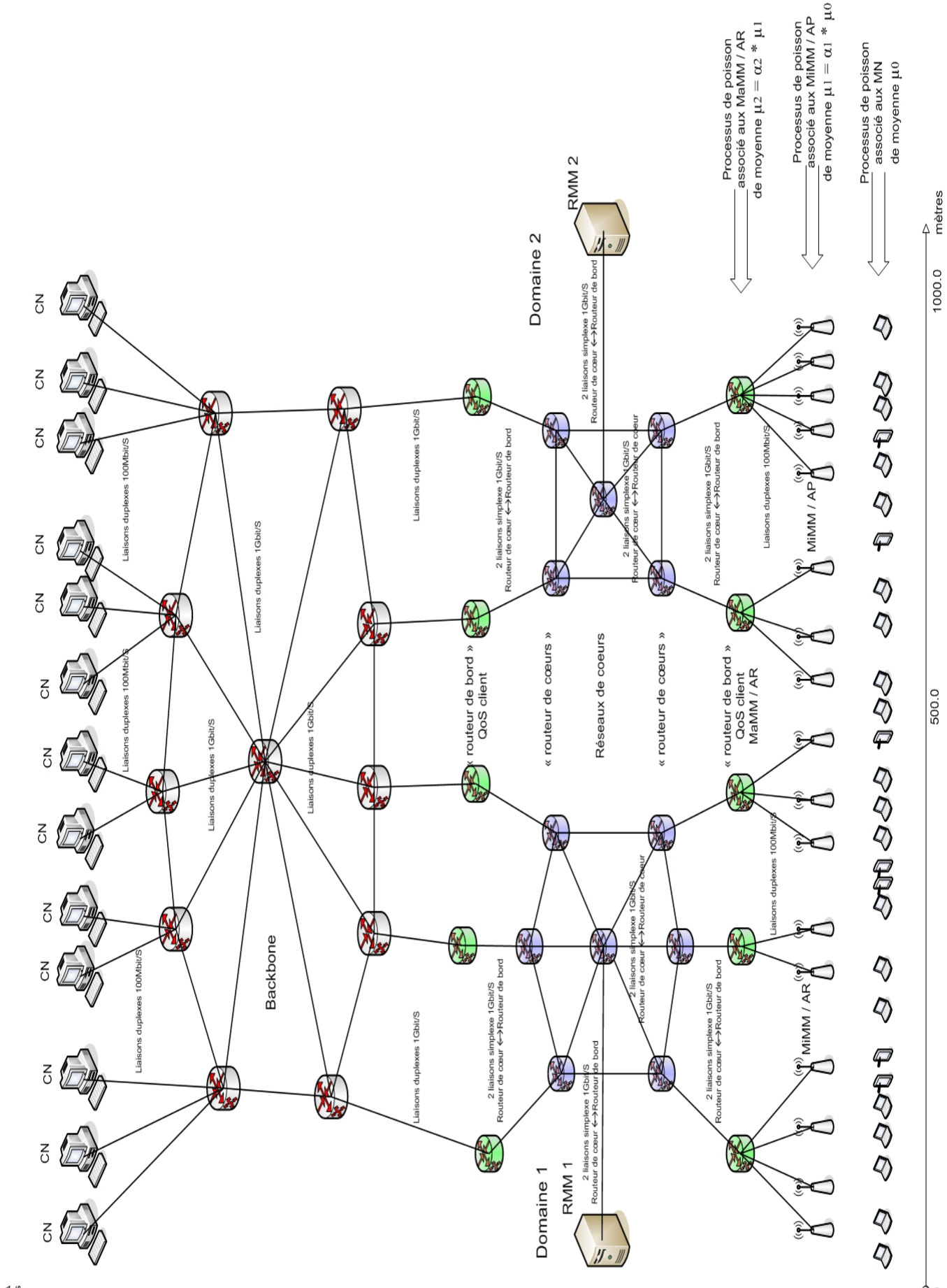


Figure V-1 :
e V-1 : Architecture

3.2. Configuration :

Liaisons duplexes entre CN et le backbone :

- débit 100 Mbits/s,
- temps de propagation 2ms,
- la file d'attente est de taille 50 paquets et de type DropTail (FIFO).

Liaisons duplexes entre les routeurs du backbone :

- débit 1Gbits/s,
- temps de propagation 2ms,
- la file d'attente est de taille 50 paquets et de type DropTail (FIFO).

Liaisons duplexes entre les routeurs de bords et le backbone :

- débit 1Gbits/s,
- temps de propagation 2ms,
- la file d'attente est de taille 50 paquets et de type DropTail (FIFO).

Routeurs de bords et leurs liaisons simplexes sortantes :

- débit 1Gbits/s,
- temps de propagation 1ms,
- file d'attente physique de taille 50 paquets et de type RED de mean paquet =100 bytes.
- la file d'attente physique se constitue de deux files d'attente virtuelles dont chacune correspond à un PHB (00 ou 10),
- les paramètres des files d'attente virtuelles ont les valeurs suivantes : min queue size threshold = 15, max queue size threshold = 45 et max dropping probability = 0.75,
- un Policer de type Token Bucket qui fait marquer les paquets out-profile avec le PHB=00 et les paquets in-profile avec PHB=10,
- les paquets de signalisation entre les MaMM et le RMM et entre les RMM sont des flots sur lesquels une Policy Token Bucket de CIR (Committed Information Rate)=100K bits/s et un CBS (Committed Burst Size)=10K bytes est appliquée.

Routeurs de cœurs et leurs liaisons simplexes sortantes :

- débit 1Gbits/s,
- temps de propagation 1ms,
- file d'attente physique de taille 50 paquets, de type RED et de mean paquet =100 bytes.
- la file d'attente physique se constitue de deux files d'attente virtuelles dont chacune correspond à un PHB (00 ou 10).
- les paramètres des files d'attentes virtuelles ont les valeurs suivantes : min queue size threshold = 15, max queue size threshold = 45 et maxi dropping probability = 0.75.

Liaison duplexe entre les MaMM et MiMM :

- débit 100Mbit/s,
- temps de propagation 2ms,
- la file d'attente est de taille 50 paquets et de type DropTail (FIFO).

Liaison entre les MiMM et les MN :

Les AP (MiMM) et les MN ont une portée de 20 mètres et ils utilisent le protocole 802.11 en mode infrastructure. Un MN se connecte à un AP pour communiquer à travers le domaine de cet AP. Pour assurer le chevauchement (overlapping), une contrainte est ajoutée lors de la génération des processus de Poisson des AP : si la distance entre deux AP est supérieure à 34 mètres, des AP, sont alors insérés pour avoir la distance inférieure ou égale à 34 mètres. Les AP de chaque domaine couvrent une distance de 500 mètres. La somme des deux distances est de 1000 mètres sur laquelle les MN se déplacent avec vitesse de 10m.

Chaque MN a, à sa création, son Home MaMM, son Home RMM, sa propre identité et son propre mot de passe. Il est ensuite inscrit sur son Homme RMM avec son identité, son mot de passe, son SLS qui contient la règle (Policy) à appliquer avec ses paramètres, le source-id de ses flots de données et le destination-id de ses flots de données.

3.3. Flow :

Chaque MN a une connexion UDP avec un CN. Les connexions sont établies d'une manière uniforme sur les CN. Sur ces connexions UDP, un générateur de trafic ON/OFF est utilisé pour modéliser une application VoIP qui, durant la période ON, génère des paquets de taille constante à un débit constant et reste silencieuse durant la période OFF. Les périodes ON et OFF sont calculées selon les distributions exponentielles de moyennes burst_time et idle_time : c'est l'application Traffic/Exponential avec les paramètres « packetSize_ = 100 bytes », « burst_time = 1000ms », « idle_time = 1000ms » et « rate_ =100K ».

Chaque communication entre un MN et un CN détermine un flots avec l'id du MN et l'id du NC. Chaque RMM contient tous les flots de ses abonnés avec la politique à appliquer sur ces flots. Le type de règle à appliquer est le même pour les flots : c'est le type Token Bucket avec CIR (débit) = 100K bit/s et CBS (burst)=10K bytes.

3.4. Scénario :

Les MN sont distribués sur la ligne de *y = 10 mètres* sous les MiMM qui sont distribués sur la ligne *y = 20 mètres*. Les coordonnées *x* des MN, des MiMM (AP) et des MaMM (AR) sont calculées selon les processus ponctuels de Poisson associés à chacun de ces types d'objets. A temps *t = 0* seconde, chaque MN se déplace à une vitesse égale à *10m/s* vers un point *x* qui est calculé avec la loi Uniforme de min = 0.0 mètre et de max = 100 mètres. Les mobiles commencent la communication avec les CN à temps *t = 0 seconde*. Le temps simulé est de *50 secondes*.

4. Comportement des protocoles :

La première étape de l'étude de l'architecture UNISIG et ses protocoles a pour finalité l'analyse du comportement des protocoles en fonction des paramètres de l'architecture. Les paramètres pertinents de l'architecture présentée précédemment sont les intensités des processus associés aux composants de réseaux. Les paramètres qui déterminent la structure du réseau sont donc : l'intensité λ_0 *MN/mètre* du processus de Poisson des MN, l'intensité λ_1 *MiMM/mètre* (ou *AP/mètre*) du processus de Poisson des MiMM (ou AP) et l'intensité λ_2 *MaMM/mètre* (ou *AP/mètre*) du processus de Poisson de MaMM (ou AR). Notons, à cet effet, que la moyenne μ d'un processus de Poisson est égale à $1/\lambda$ tel que λ est l'intensité de ce processus. Dans notre simulation, c'est la moyenne qui est utilisée et non l'intensité parce que le générateur de la loi exponentielle de NS prend comme paramètre seulement la moyenne.

Cette simulation est en réalité un ensemble de simulations avec les différents paramètres des processus de Poisson. Chaque simulation effectuée est destinée au protocole UNISIG ainsi qu'au protocole MiPv4 exécuté indépendamment du protocole d'authentification avec gestion de la QoS. Afin d'abréger le nom de cet signalisation indépendante, le protocole est appelé INDSIG. Nous opérons donc plusieurs simulations en faisant varier la moyenne μ_0 du processus des MN, le facteur de multiplication α_1 qui permet de calculer la moyenne μ_1 du processus de MiMM (AP) tel que $\mu_1 = \alpha_1 * \mu_0$ et le facteur de multiplication α_2 qui permet de calculer la moyenne μ_2 du processus de MaMM (AR) tel que $\mu_2 = \alpha_2 * \mu_1$. On prend le facteur de multiplication α supérieur ou égal à 1. Chaque simulation est donc distinguée par le triplet $(\mu_{0i}, \alpha_{1j}, \alpha_{2k})$ tel que i, j et k entiers. Les résultats sont issus d'un ensemble de simulations tel que les valeurs de μ_{0i} appartiennent à {12, 13, …30}, les valeurs α_{1j} appartiennent à {2, 3, …20} et les valeurs de α_{2k} appartiennent à {1, 2, …15}. Pour les triplets de valeurs $(\mu_{0i}, \alpha_{1j}, \alpha_{2k})$ qui induisent la valeur de μ_2 supérieure à 500 *mètres/MaMM (mètre/AR)*, la simulation n'est pas exécutée car cette moyenne du processus des MaMM (ou AR) doit être inférieure à 500. Cela est dû au fait que dans notre architecture un domaine couvre 500 mètres et il faut générer au moins un MaMM (AR) pour construire un domaine.

Rappelons que le protocole UNISIG s'organise principalement autour de trois procédures dont l'exécution dépend du type du mouvement du MN. Ces trois procédures sont : Roaming (RM) qui est exécutée lorsque le MN change de domaine ou s'allume, Macro-Mobility (MaM) qui est exécutée lorsque le MN change de MaMM sans changer de domaine et Micro-Mobility (MiM) qui est exécutée lorsque le MN change de MiMM sans changer de MaMM. Pour l'exécution du protocole MIPv4 indépendamment du protocole d'authentification et de gestion de la QoS (INDSIG), seulement deux procédures sont exécutées : la procédure Roaming (RM) qui est exécutée lorsque le MN change de domaine ou d'AP sous un domaine visité ou lorqu'il s'allume et la procédure Macro-Mobility (MaM) qui est exécutée lorsque le MN change d'AP sous son domaine.

Les premiers résultats de cette simulation expriment les durées moyennes des procédures exécutées dans chacune des simulations pour tous les MN. La relation « ***RM>MaM>MiM*** », dans le cas de UNISIG, signifie que « la durée moyenne de l'exécution de la procédure de Roaming est supérieure à la durée moyenne de l'exécution de la procédure Macro-Mobility qui, elle, a une durée moyenne d'exécution supérieure à la durée moyenne de l'exécution de la procédure Micro-Mobility ». Dans le cas de INDSIG, la relation *« **RM>MaM*** » signifie que « la durée moyenne de l'exécution de la procédure de Roaming est supérieure à la durée moyenne de l'exécution de la procédure Macro-Mobility ». D'après les quantités de signalisations générées par chacune des procédures et d'après les éléments du réseau, ainsi que les distances qui séparent ces éléments qui interviennent dans chacune de ces procédures si le réseau est bien stable et cohérent, la relation « ***RM>MaM>MiM*** » et la relation «***RM>MaM***» doivent être satisfaites.

4.1. Résultats :

Nous présentons ci-dessous deux nuages de points : celui de gauche (Figure V-2) est produit dans le cas des simulations du protocole INDSIG ; celui de droite (Figure V-3) est produit dans le cas des simulations du protocole UNISIG. La présence d'un point dans l'espace ayant comme coordonnées un triplet de valeurs $(\mu_{0i}, \alpha_{1j}, \alpha_{2k})$ signifie que pour la simulation des protocoles dans une architecture ayant comme paramètres les trois valeurs de ce triplet donne des résultats qui vérifient la relation «***RM>MaM>MiM***» pour UNISIG et la relation «***RM>MaM***» pour INDSIG.

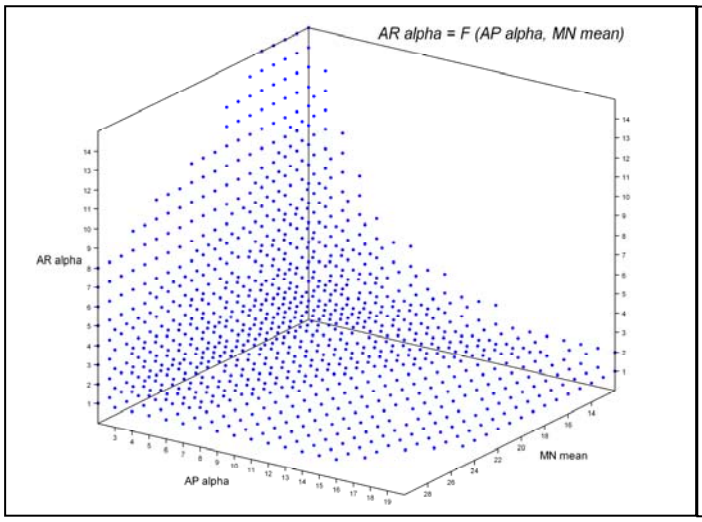


Figure V-2 : nuage de points représentant les paramètres pour lesquels la relation «RM >MaM» est satisfaite pour INDSIG.

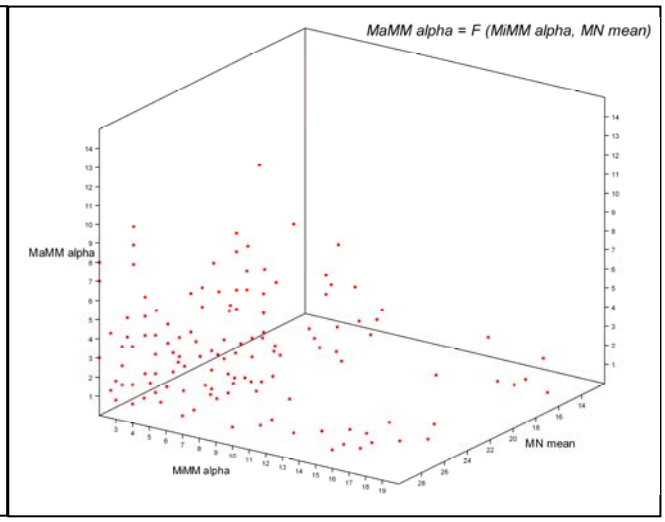


Figure V-3 : nuage de points représentant les paramètres pour lesquels la relation «RM>MaM>MiM » est satisfaite pour UNISIG

Nous remarquons que le nuage de points de INDSIG (Figure V-2) est régulier. Presque tous les points qui manquent pour remplir le cube, correspondent aux simulations non exécutées quand μ_2 est supérieure à 500. Nous pouvons toutefois distinguer certains points tel que (22, 14, 1) et (20, 16, 1) qui ne satisfont pas la relation «***RM>MaM***». Par contre, en observant le nuage de points de UNISIG (Figure V-3), nous remarquons que le nombre de points qui satisfont la relation «***RM>MaM>MiM***» est modeste.

Le nuage de points de UNISIG est concentré dans la région où μ_0 est supérieur à 16, α_1 inférieur à 9 et α_2 inférieur à 6. Cela explique que le protocole se comporte convenablement pour une moyenne du processus de MN supérieur à 16 (une intensité inférieure à 1/16 *MN/m*) sous une architecture avec des MiMM dont chacun gère en moyenne 8 MN ou moins et des MaMM dont chacun gère en moyenne 5 MiMM ou moins. Dans d'autres régions où les points existent, ces points sont éparpillés d'une façon hasardeuse mais il n'y a aucun point pour des valeurs de α_2 supérieures à 9. Cela veut dire que si chaque MaMM gère en moyenne 9 MiMM ou plus le protocole s'exécute improprement. Pour des valeurs de α_1 supérieures à 9 les points sont très rares et n'existent seulement pour la valeur α_2 égale 1. Nous expliquons cela par le fait que si chaque MiM gère en moyenne plus de 9 MN il faudrait alors, pour une bonne exécution de UNISIG, prévoir une configuration du réseau d'accès qui permet à chaque MaMM de gérer un seul MiMM. Pour des valeurs de α_2 entre 5 et 8, les points existent uniquement pour la valeur $\alpha_1 = 2$ qui est la plus petite valeur de α_1. Donc, si pour chaque MaMM, qu'on destine à gérer en moyenne de 5 à 8 MiMM, ce qui est considérable, alors les MiMM de cette architecture doivent chacun gérer le minimum de MN ; autrement dit, dans notre cas, un MiMM doit gérer au maximum 2 MN.

Pour tenter de comprendre quelle est la procédure qui influence le plus sur la relation « ***RM>MaM>MiM*** » dans l'architecture UNISIG, nous commencerons par l'examen du tableau suivant (Tableau 3) qui donne le nombre et le pourcentage des simulations qui ne satisfont pas la relation « ***RM>MaM>MiM*** » selon des relations entre deux procédures qui la rend insatisfaite. La troisième, quatrième et la cinquième colonne représentent le nombre et le pourcentage des simulations qui rendent la relation « ***RM>MaM>MiM*** » fausse selon la relation indiquée («***RM<MaM***»,«***RM<MiM*** » et «***MaM<MiM***») par rapport au nombre total des simulations qui rendent la relation « ***RM>MaM>MiM*** » fausse et qui est représenté dans la deuxième colonne.

Relation rendant la relation « ***RM>MaM>MiM*** » fausse.	$\overline{RM>MaM>MiM}$	RM<MaM	RM<MiM	MaM<MiM
nombre de simulations ne satisfaisant pas la relation « ***RM>MaM >MiM*** »	1044	481	685	791
pourcentage du nombre de simulation ne satisfaisant pas la relation « ***RM>MaM>MiM*** »	100.00 %	46.07 %	65.61 %	75.77 %

Tableau 3 : Influence de RM, de MaM et de MiM sur la relation RM>MaM>MiM

Nous remarquons que les pourcentages de «***MaM<MiM***» et de «***RM<MiM***» ont les plus grandes valeurs et dépassent amplement le pourcentage de «***RM<MaM***». De plus, le pourcentage de la relation «***MaM<MiM***» est la plus grande valeur. Il découle de ce constat que la durée de la procédure qui influence négativement le plus la relation «***RM>MaM>MiM***» est la durée de la procédure MiM. Donc, dans l'exécution du protocole UNISIG, la durée moyenne de la procédure de MiM croît anormalement devant la durée moyenne de la procédure MaM et devant la durée moyenne de la procédure RM. Si la durée moyenne de la procédure de MiM était inférieure à la durée moyenne des procédures RM et MaM, le nuage de points de UNISIG serait plus dense ; ceci permettrait un choix plus large dans la configuration l'architecture. Mais pourquoi est-ce la durée de la procédure MiM qui croît proportionnellement plus que les durées moyennes des autres procédures alors que, théoriquement, c'est la plus petite procédure qui devrait être la plus courte puisqu'elle correspond à la micro mobilité ?

Pour savoir pourquoi c'est la durée moyenne de la procédure MiM qui augmente anormalement par rapport aux autres procédures et qui influence le plus négativement la relation «***RM>MaM>MiM***» observons le graphe ci-dessous à droite (Figure V-5) qui montre le pourcentage du nombre de procédures de MiM, de MaM et de RM par rapport au nombre global des procédures exécutées dans chaque simulation.

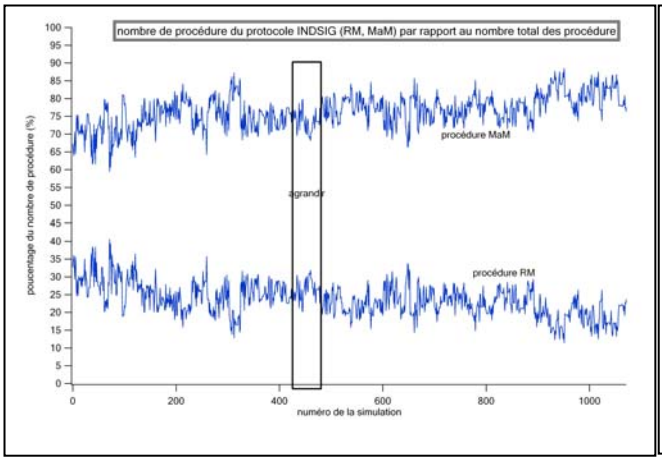


Figure V-4 : pourcentage du nombre de procédures (INDSIG)

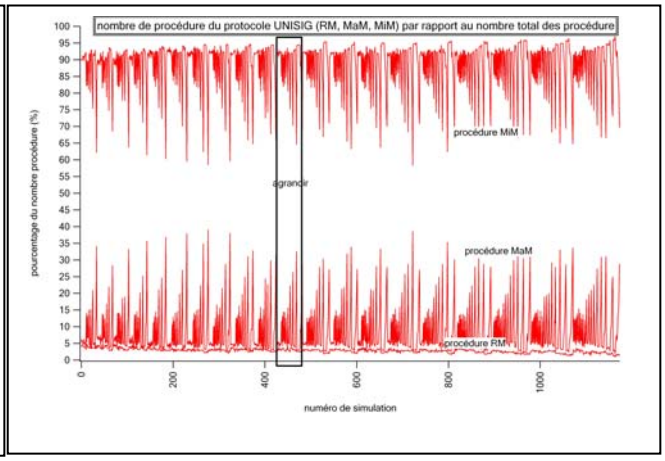


Figure V-5 : pourcentage du nombre de procédures (UNISIG)

On remarque que le graphe UNISIG (Figure V-5) des pourcentages des procédures est périodique ; son comportement se répète à chaque moyenne du processus des MN avec l'ensemble des valeurs (α_1, α_2) associées. Prenons une période ; celle de μ_0=20 par exemple. On voit ci-dessous l'agrandissement (Figure V-6 et Figure V-7) de cette partie des deux graphes précédents.

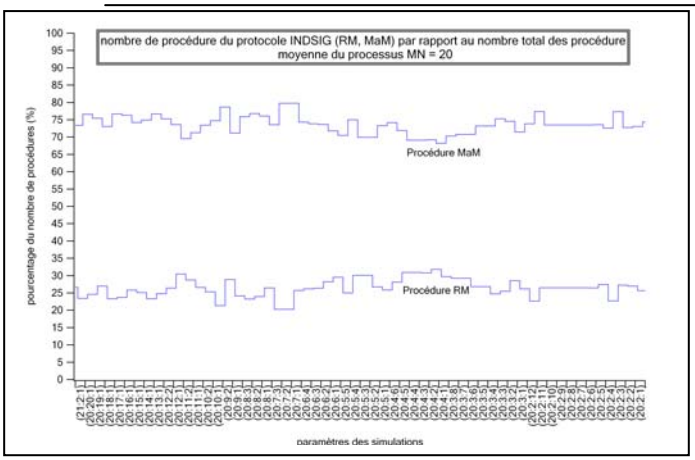


Figure V-6 : pourcentage du nombre de procédures (INDSIG), µ0 = 20

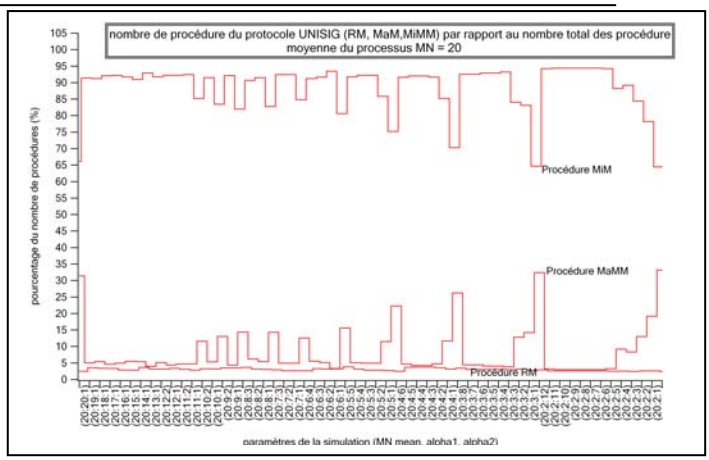


Figure V-7 : pourcentage du nombre de procédures (UNISIG), µ0 = 20

Sur le graphe Figure V-5, on constate que le pourcentage du nombre de procédures de MiM est trop élevé par rapport au deux autres procédures (RM et MaM). Il varie entre 70% et 95%, mais il est plus concentré à 90% pendant que celui de MaM varie entre 5% et 30% et il est plus concentré à 7%; tandis que le pourcentage du nombre de procédures de RM est quasi-stable avec un faible taux de 4%. Le nombre de procédures MiM représente à peu près 13 fois le nombre de procédures de MaM. C'est pour cela que la durée de MiM influence fortement la relation «***RM>MaM>MiM***». La croissance majeure de la procédure MiM dans la simulation congestionne le réseau d'accès au niveau des MiMM et des MaMM. La signalisation de la procédure MiM fait intervenir seulement les MN, les MiMM et les MaMM sans que le trafic de la signalisation circule dans le domaine et, avec un fort pourcentage, augmente la bande passante occupée par la signalisation. Ceci allonge la durée de la procédure MiM et, avec une grande fréquence de celle-ci, sa durée d'exécution influence nettement et négativement la relation «***RM>MaM>MiM***». La quantité de la bande passante occupée par la signalisation entre le les MiMM et les MaMM est proportionnelle au nombre de MiM. Si on doit vraiment compter sur la procédure MiM pour optimiser le temps du Handover, il faut trouver un compromis entre le nombre de MiM et la distribution des MiMM et des MaMM ; ce qui se traduirait par une recherche d'équilibre entre une architecture économique et un service plus au moins satisfaisant.

Dans le cas de la simulation du protocole INDSIG, le graphe Figure V-4 montre que la répartition du pourcentage du nombre de procédures RM et celui du nombre de procédures MaM sont bien réparties. Cela crée un équilibre dans le réseau d'accès puisque tous les trafics de cette signalisation traversent le domaine et, dans le cas de RM, ils traversent aussi le backbone qui simule l'Internet. Le trafic se retrouve équitablement réparti sur tous les nœuds des domaines. Une comparaison bijective entre UNISIG et INDSIG n'est pas possible puisque le nombre de procédures n'est pas le même. Néanmoins, cela met en évidence l'inconvénient d'avoir une procédure -procédure MiM dans UNISIG- qui fait concentrer la signalisation dans une seule partie du réseau même si cette procédure est très courte.

5. Durée d'exécution des procédures :

Dans cette section, nous étudierons en parallèle l'efficacité des deux protocoles sur les sujets qui peuvent influencer la QoS. Il s'agit d'analyser la comparaison des deux protocoles UNISIG et INDSIG sur la durée moyenne indispensable pour que le MN réacquiert la connexion à un nouveau point d'accès après avoir perdu sa connexion à l'ancien point d'accès à cause de son déplacement. Ce paramètre influence d'une manière directe la durée de l'exécution du Handover et, par conséquent, la QoS offerte au mobile. La duré moyenne de l'exécution des procédures est calculée sous la base de tous les types de procédures (RM, MaM, MiM) pour tous les MN en même temps.

Pour commencer, on doit distinguer seulement quelques paramètres pour avoir des architectures sur lesquelles les protocoles s'exécutent selon les moyennes des processus MN. Pour cela, on doit relever les paramètres qui permettent d'avoir le maximum de moyennes du processus des MN qui satisfont la relation «***RM>MaM>MiM***». En d'autres termes, nous cherchons α_1 et α_2 qui rendent l'architecture plus stable pour le maximum du nombre de paramètres μ_0.

5.1. Résultat :

Le graphe ci-dessous représente un nuage de points dont les hauteurs par rapport au plan (α_1, α_2) illustrent le nombre de moyennes μ_0 qui satisfont la relation « ***RM>MaM>MiM*** » (Figure V-8) dans le cas de UNISIG.

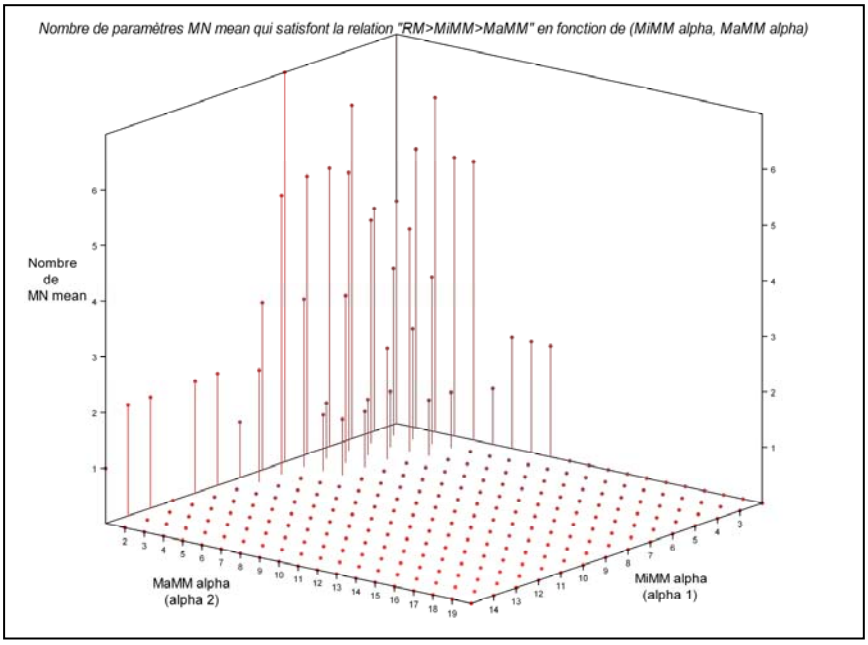


Figure V-8 : nuage de points représentant le nombre de moyennes du processus MN qui satisfont la relation RM>MaM>MiM pour UNISIG.

On cherche à repérer les valeurs de (α_1, α_2) qui donnent les plus grandes valeurs du nombre de μ_0. Le graphe montre que les valeurs de μ_0 supérieures à 0 sont concentrées au-dessus du semi plan (α_1, α_2) tel que α_1 est entre 3 et 7 et α_2 entre 2 et 5. Pour α_2 égal à 1, les points sont plus hauts et plus fréquents. Pour α_1=2, quand α_2 est inférieur à 9, les points sont avec une hauteur assez grande par rapport aux autres points. Plus α_1 et α_2 se rapprochent parallèlement à la leurs valeurs minimales respectives qui sont 2 pour α_1 et 1 pour α_2 plus les hauteurs des points grandissent. Plus α_1 se rapproche de la valeur de $\alpha_1 = 7$ parallèlement à α_2 qui se rapproche à la valeur $\alpha_2 = 5$, la hauteur des points diminue. D'après ce graphe, l'idéal est de choisir $\alpha_2 = 1$ mais cela revient à construire une architecture dont chaque MaMM gère un seul MiMM. De ce fait, le protocole UNISIG perd tout son sens pour le comparer puisqu'il n'y aura plus de procédure de MiM. En effet, la procédure de macro mobilité sera confondue avec celle de micro mobilité puisqu'on aura un MaMM pour un MiMM. En choisissant de prendre la valeur de $\alpha_1 = 2$, on aura une architecture qui gère seulement 2 MN par MiMM et avec en importante densité de MiMM. Ce qui peut augmenter dans certains cas le nombre de procédures de MiM. Les paramètres qu'on doit choisir doivent être cohérents et refléter la réalité. On prendra donc des valeurs de α_1 entre 3 et 5 et des valeurs de α_2 entre 3 et 5 qui correspondent aux plus hauts points sur le graphe. L'échantillon

des paramètres pris pour l'étude est celui qui permet la construction des architectures stables. On prendra les valeurs du vecteur (α_1, α_2) appartenant à l'ensemble {(3,4), (3,5), (4,4), (4,5), (5,3), (5,4)}

Pour chaque vecteur (α_1,α_2) on construit le graphe qui illustre la durée moyenne des procédures exécutées en mouvement de MN pour les deux protocoles ; la durée moyenne des procédures en fonction de la densité des MN. Chaque figure graphique ci-dessous illustre deux graphes dont chacun est associé à l'un des protocoles. Le bleu est le graphe du protocole INDSIG et le rouge est le graphe du protocole UNISIG. L'ensemble des points des résultats des simulations, est rapproché sur chaque graphe par une interpolation linéaire.

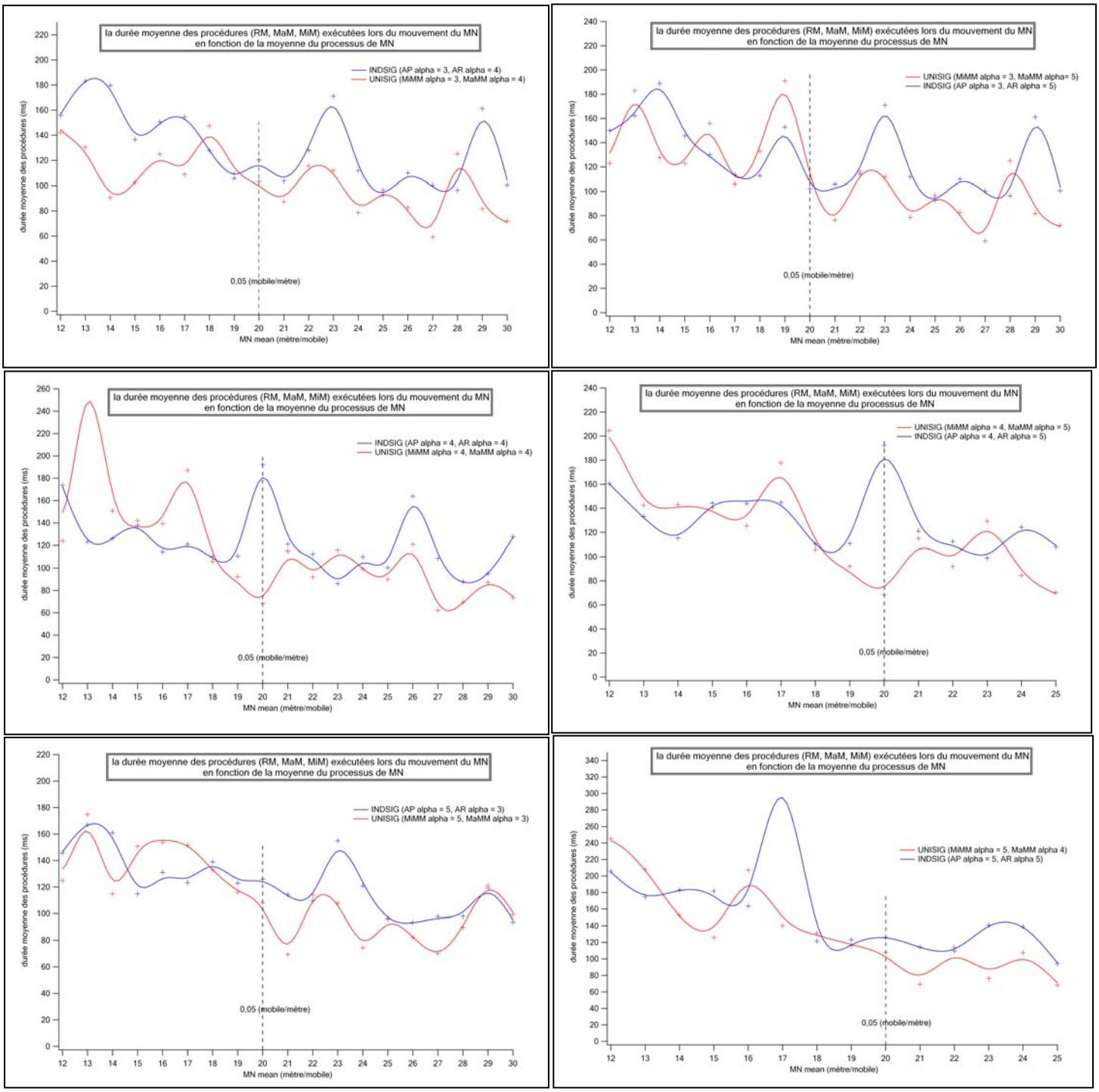


Figure V-9 : graphes de la durée moyenne des procédures exécutées lors du mouvement du MN en fonction de la moyenne du processus des MN.

Nous constatons sur ces graphes que la durée moyenne d'exécution des procédures décroît lentement avec l'augmentation de la moyenne des processus. Nous notons qu'après une certaine valeur de la moyenne μ_0 -19 pour le graphe (3,4), 20 pour le graphe (3,5), 16 pour le graphe (5,4) et 18 pour les autres graphes- la durée moyenne des procédures de UNISIG est régulièrement inférieure à celle des procédures de INDSIG. Pour μ_0 inférieur à ces valeurs, la comparaison des deux durées moyennes est incertaine ; dans certains cas, la durée des procédures UNISIG est inférieure à celles des procédures INDSIG et dans d'autres cas c'est l'inverse.

Donc, lorsque la densité des mobiles diminue, la durée moyenne des procédures pour les deux types de protocoles diminue, mais la durée moyenne des procédures dans le cas de UNISIG diminue plus rapidement que dans le cas de INDSIG. Cela s'explique par le fait que l'accroissement de l'intensité des mobiles augmente la quantité de la signalisation traitée par les nœuds du réseau d'accès (MiMM/AP, MaMM/AR). Les MaMM (AR) sont moins encombrés et gèrent plus rapidement la signalisation. Dans le cas de UNISIG, quand la densité des mobiles est modeste, le réseau d'accès n'a donc plus d'obstacle pour mieux préparer la QoS dans la mobilité.

Nous déduisons que pour une population de mobiles suivant un processus de Poisson avec intensité inférieure à 0.05 *MN/m* (montrée dans la Figure V-9), le protocole UNISIG permet une exécution de procédures plus optimale que dans le cas du protocole INDSIG. En généralisant ce résultat, on peut dire que dans une région à faible population de mobiles, le protocole UNISIG est largement favorisé et la QoS est certainement mieux préparée dans le cas de UNISIG. Dans une région à une forte population de mobiles, les deux protocoles sont similaires sur le plan du temps d'exécution du Handover.

6. Conclusion :

Le but principal d'une architecture unifiée est l'amélioration de la préparation de la QoS lors de la mobilité des nœuds sans pénaliser la durée d'exécution du Handover. Pour arriver à cette fin, le protocole UNISIG transporte les données des différentes signalisations en même temps, rapatrie les données de l'authentification et de la gestion de la QoS à l'endroit du réseau le plus proche du mobile et exploite le principe de MiPv4 hiérarchique pour des mises à jour locales au niveau du réseau d'accès et du domaine visité. Ces concepts exploités par UNISIG semblent être des facteurs incontestablement favorisant une signalisation optimisée pour la prise en charge de la QoS lors de la mobilité. Dans le cas de la mobilité sans besoin de la QoS, l'architecture UNISIG n'est pas adaptée ; une signalisation indépendante (INDSIG) est la plus appropriée.

En revanche avec le protocole UNISIG, il est nécessaire d'apporter quelques aménagements dans le réseau comparé à la signalisation indépendante. Celle-ci devrait favoriser la mise en place de la QoS sans augmenter la durée d'exécution des procédures de mobilité. Un paramètre très important dans la planification de cette architecture est le seuil acceptable de nombre de nœuds mobiles. En effet, dépassant ce seuil, il y aurait une fréquence importante d'exécution de la procédure Micro-Mobilité ; augmentant la bande passante occupée par la signalisation UNISIG au niveau du réseau d'accès. En utilisant une simulation de notre architecture, nous avons pu détecter pour notre scénario les paramètres qui permettront au protocole UNISIG de s'exécuter de manière stable. Bien que la signalisation indépendante ne nécessite pas de modification de l'architecture de communication TCP/IP, le protocole UNISIG reste cependant plus intéressant pour considérer des communications avec QoS en mobilité. La durée d'exécution du Handover, dans le cas de UNISIG, n'est pas plus longue que celle de l'exécution du Handover dans le cas de la signalisation indépendante alors

qu'on plus UNISIG met en place la QoS. Cela reste encore plus sûr quand la densité des mobiles est modeste.

Dans ce travail, nous avons utilisé le simulateur NS qui est conçu initialement pour simuler des réseaux IP filaires. Pour permettre de simuler avec NS le protocole UNISIG, nous avons intégré progressivement des modules assurant l'authentification Diameter, la gestion dynamique de la QoS et la gestion de la mobilité hiérarchique. Le simulateur reste stable après cette intégration et donne des résultats satisfaisants. NS s'avère être un outil de simulation intéressant et reste ouvert à d'autres extensions qui permettront la simulation d'autres protocoles.

Ce travail permet en outre une ouverture sur d'autres champs d'étude pour la signalisation unifiée. Dans le cadre du protocole UNISIG, plusieurs paramètres restent non étudiés et d'autres simulations peuvent être effectuées. On peut ainsi proposer l'étude des pertes de paquets quand le mobile effectue un Handover, l'analyse du comportement du protocole en fonction de la vitesse des mobiles et le calcul de la bande passante exacte occupée par le protocole UNISIG sur tous les nœuds du réseau. La connectivité des mobiles est aussi un sujet d'étude intéressant car si le mobile n'entame pas la mise à jour de la localisation et l'authentification dès sa déconnexion de l'ancien point d'accès, la durée d'exécution du Handover se voit prolongée.

Liste des Figures :

Liste des Diagrammes de séquence :

Liste des Tableaux :

Liste des Organigrammes :

Références :

[1] C. Perkins, “IP Mobility Support”, RFC2002, October 1996

[2] Ian F. Akyildiz, Janise Mcnair, Joseph S. M. Ho, Huseyin Uzunalio Glu, Wenye Wang, “Mobility Management in Next-Generation Wireless Systems” PROCEEDINGS OF THE IEEE VOL.87, NO.8, August 1999

[3] C. Perkins, “IP Encapsulation within IP”, RFC2003, October 1996

[4] Chaouchi H, « Chapitre 8 : La Gestion de la micro mobilité »

[5] Georgios Karagiannis, Geert Heijenk “Mobile IP, State of the Art Report”, No 3/0362-FCP NB 102 88 Uen, 1999-07-13

[6] C.Perkins, David B. Johnson, “Route Optimization in Mobile IP”, draft-ietf-mobilip-optim-08.txt, 25 Feburary 1999

[7] D. Johnson, C. Perkins, J. Arkko, “Mobility Support in IPv6”, RFC3775, June 2004.

[8] T. Narten, E. Nordmark, W. Simpson,“Neighbor Discovery for IP Version 6(IPv6)”,RFC1970, August 1996

[9] S. Thomson, T. Narten, “IPv6 Stateless Address Autoconfiguration”, RFC1971,August 1996

[10] S. Deering, R. Hinden, “Internet Protocol Version 6 (IPv6) Specification”, RFC2460, December 1998.

[11] J. Manner, A. López, A. Mihailovic, “Evaluation of Mobility and QoS Interaction”, Computer Networks, 38(2):137-163, Feb 2002.

[12] E. Gustafsson, A. Jonsson, C. Perkins “Mobile IPv4 Regional Registration” (draft-ietf-mip4-reg-tunnel-01) November 29, 2005

[13] Chaouchi H, “ Chapitre 1 : “ Mobilité et Qualité de Service dans les réseaux IP ”

[14] R. Braden, D. Clark, S. Shenker “ Integrated Services in the Internet Architecture: an Overview ”, RFC1633, June 1994.

[15] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin “ Resource ReSerVation Protocol (RSVP)— Version 1 Functional Spécification ” RFC2205, September 1997.

[16] Lixia Zhang, Stephen Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala “ RSVP: A New Resource ReSerVation Protocol” IEEE Network Magazine – Volume 7, Number 5 September 1993.

[17] J. Wroclawski “ The Use of RSVP with IETF Integrated Services ” RFC2210, September 1997.

[18] S. Shenker, C. Partridge, R. Guerin “ Specification of Guaranteed Quality of Service ” RFC2212, September 1997.

[19] J. Wroclawski “ Specification of the Controlled-Load Network Element Service ” RFC2211, September 1997.

[20] M. Christophe Deleuze “Qualité de service dans l’Internet : problèmes liés au haut débit et au facteur d’échelle” Thèse de Doctorat de l’université Paris VI Pierre et Marie Curie, 13 janvier 2000.

[21] K. Nichols, S. Blake, F. Baker, D. Black “Definition of the Differentiatiated Services Field (DS Field) in the IPv4 and IPv6 Headers” RFC2474, December 1998.

[22] S. Blake, D. Blake, M. Carlson, E. Davies, Z. Weiss “An Architecture for Differentiated Service” RFC2475, December 1998.

[23] J. Heinanen, F. Baker, W. Weiss, J. Wrolawski “Assured Forwarding PHB Group” RFC2597, June 1999.

[24] V. Jacobson, K. Nichols, K. Poduri “An Expedited Forwarding PHB” RFC2598, June 1999.

[25] D. Grossman “New Terminologie and Clarifications for DiffServ” RFC3260, April 2002.

[26] B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V.Firoiu, D. Stiliadis “An Expedited Forwarding PHB (Per-Hop Behavior)” RFC3246 March 2002.

[27] Y. Bernet, S. Blake, D. Grossman, A. Smith "An Informal Management Model for DiffServ Routers" RFC3290, May2002.

[28] Y. Bernet, P.Ford, R. Yavatkar, F.Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, E. Felstaine " A Framework for Integrated Services Operation over DiffServ Networks" RFC2998, November 2000.

[29] C. de Laat, G. Gross, L.Gommans, J.Vollbrecht, D. Spence " Generic AAA Architecture" RFC2903, August 2000.

[30] J. Vollbrecht, P Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence " AAA authorization Framework" RFC2904 August 2000.

[31] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence "AAA Authorization application Examples" RFC2905, August 2000.

[32] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence "AAA Authorization requirements" RFC2906, August 2000.

[33] C. Rigney, A. Rubens, W. Simpson, S. Willens, " Remote Authentication Dial In User Service (RADIUS)" RFC2865, June 2000.

[34] C. Rigney "RADIUS Accounting" RFC2866 , June 2000.

[35] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, "Diameter Base Protocol", RFC3588, September 2003.

[36] P. Calhoun, G. Zorn, D. Spence, D. Mitton, "Diameter Network Access Server Application", RFC4005 August 2005.

[37] P. Calhoun, T. Johansson, C. Perkins, T. Hiller, P. McCann, " Diameter Mobile IPv4 Application", RFC4004 August 2005.

[38] D. Durham, J. Boyle, R. Cohen, S. Herzog, R.Rajan, A. Sastry, "The COPS Protocol" RFC2748 January 2000.

[39] S. Hrzog, J. Boyle, R. Cohen, D. Durham, R. Rajan, A. Sastry, "COPS usage for RSVP" RFC2749 January 2000.

[40] K. Chan, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, A. Smith, "COPS Usage for Policy Provisioning (COPS-PR)" RFC3084 March 2001.

[41] S. Herzon, " RSVP Extentions for Policy Control", RFC2750, January 2000.

[42] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz "Extensible Authentification (EAP)" RFC3748 June 2004.

[43] R. Rivest, "The MD5 Message-Digest Algorithm", RFC1321, April 1992.

[44] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[45] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.

[46] G. Zorn, S. Gobb, "Microsoft PPP CHAP Extentions", RFC2433, October 1998.

[47] G. Zorn, "Mircosoft PPP CHAP Extentions, Version 2", RFC2759, January 2000.

[48] Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Port-Based Network Access Control", IEEE Standard 802.1x, September 2001.

[49] Institute of Electrical and Electronics Engineers, "Wireless LAN Medium Access Control and Physical Layer (PHY) Specification", IEEE Standard 802.11, 1999.

[50] Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Overview and Architecture" IEEE Standard 802, 1990.

[51] N. Haller, C. Metz, P. Nesser, M. Straw, "A One-Time Password System", RFC 2289, May 1998.

[52] T. Dierks, C. Allen,"The TLS Protocol : Transport Layer Security Version 1.0", RFC 2246, January 1999.

[53] B. Aboba, D. Simon, " PPP EAP TLS Authentication Protocol" RFC2716, October 1999.

[54] Aurélien Géron "WiFi Déploiement et sécurité" Dunod 2004.

[55] B. Aboba, P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentification Protocol (EAP)" RFC3579, September 2003.

[56] P. Eronen, T. Hiller, G. Zorn, " Diameter Extensible Authentication Protocol (EAP) Application" RFC4072, August 2005.

[57] D. Mitton, M. St.Johns, S. Barkley, D. Nelson, B. Patil, M. Stevens, M. Stevens, B. Wolff, "Authentication, Authorization, and Accounting Protocol Evaluation" RFC3127, June 2001.

[58] Chaouchi H "Global Mobility Control in Wireless and Mobile networks" PhD Thesis, Ocotober 2004.

[59] A. O'Neill, G. Tsirtsis "Edge Mobility Architecture" : draft-oneill-ema-02, Internet Draft, July 2000.

[60] N. BADACHE, T.LEMLOUMA " le routage dans les réseaux mobiles Ad hoc" (http://opera.inrialpes.fr:80/people/Tayeb.Lemlouma/Papers/AdHoc_Presentation.pdf)

[61] J. Loughney, M. Nakhjiri, C. Perkins, R. Koodli "Context Transfer Protocol (CXTP)", RFC 4067, July 2005.

[62] Jyh-Cheng Chen, Armando Coro, Anthony McAuley, Shinichi Baba, Yoshihiro Ohba, Parameswaran Ramanathan, "A QoS Architecture for Future Wireless IP Networks".

[63] Anup K. Talukdar, B. R. Badrinah, Arup Acharya "MRSVP: A Reservation Protocol for an Integrated Services Packet Network with Mobile Hosts".

[64] Hahnsang Kim, Hossam Afifi "Improving Mobile Authentication with New AAA Protocols"

[65] R. Moskowitz, P. Nikander, P. Jokela, T. Henderson "Host Identity Protocol" (draft-ietf-hip-base-06), June 15 2006.

[66] T. Henderson "End-Host Mobility and Multihoming with the Host Identity Protocol" (draft-ietf-hip-mm-04), June 2006.

[67] R. Moskowitz, P. Nikander "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.

[68] W. Simpson "PPP Challenge Handshake Authentication Protocol (CHAP)" , RFC 1994, August 1996.

[69] B. Aboba, M. Beadles, J. Arkko, P. Eronen "Network Access Identifier", RFC 4282, December 2005.

[70] The VINT Project; collaboration between researchers at UC Berkeley, LBL, USC/ISI and Xerox PARC. http://www.isi.edu/nsnam/.

Annexe A : RSVP

La communication entre les nœuds avec le protocole RSVP est accomplie en utilisant un ensemble de messages. Un message RSVP est constitué d'un en-tête commun suivi du corps qui regroupe l'ensemble variable d'objets de taille variable comme suit :

***Message RSVP* = [en-tête commun] [objet1]…[objet N]**

L'en-tête commun contient un champ qui spécifie le type du message RSVP. Pour chaque type de message un ensemble de règles détermine les objets que ce message peut contenir et dans quel ordre.

Un objet RSVP est un ensemble de mots de 32 bits dont le premier est l'en-tête de l'objet lui-même. Les champs de l'en-tête spécifient la longueur de l'objet, la classe d'objet et le type de la classe. Les classes d'objet son résumées dans le (**Erreur ! Source du renvoi introuvable.**). Le type de la classe prend généralement deux valeurs pour spécifier une utilisation avec IPv4 ou avec IPv6. Il sera utile pour une éventuelle extension du protocole.

Classe d'objet	Description
NULL	Un objet nul. Il est ignoré par les traitements dans les nœuds.
SESSION	Contient adresse IP du récepteur, ID protocole IP et le port du protocole UDP ou TCP. Il définit une session RSVP pour les objets qui le suivent. Il est exigé dans tous les messages RSVP.
RSVP_HOP	Contient l'adresse IP du nœud qui envoie le message qui contient cet objet.
TIME_VALUE	Contient la valeur en milliseconde de la période de rafraîchissement utilisée par le créateur du message. Il est exigé dans le message *Path* et le message *Resv*.
STYLE	Définit le style de réservation. Il est exigé dans le message *Resv*.
FLOWSPEC	Définit la QoS souhaité. Il est exigé dans le message *Resv*.
FILTER_SPEC	Définit l'ensemble de paquets qui recevront la QoS définie dans FLOWSPEC. Contient l'adresse IP de l'émetteur et le port de l'application ou le champ « Flow Label » d'IPv6. Il est exigé dans le message *Resv*.
SENDER_TEMPLATE	Contient l'adresse IP de l'émetteur et probablement des informations pour identifier l'émetteur. C'est le FILTER_SPEC qui est envoyé par l'émetteur. Il est exigé dans le message *Path*.
SENDER_TSPEC	Définit les caractéristiques du flot de données de l'émetteur. Il est exigé dans le message *Path*.
ADSPEC	Contient les données de OPWA. Il apparaît dans le message *Path*.
ERROR_SPEC	Il spécifie une erreur dans les messages *PathErr* et *ResvErr* ou une confirmation dans le message *ResvConf*. Il contient l'adresse IP du nœud qui a détecté l'erreur et le code de l'erreur.
POLICY_DATA	Contient des informations qui permettent au module Policy Control de décider si la réservation associée est administrativement permise. Peut apparaître dans les messages *Path*, *Resv*, *PathErr* ou *ResvErr* [41].
INTEGRITY	Transporte des informations pour assurer l'intégrité du contenu du message RSVP.
SCOPE	Contient la liste des émetteurs qui recevront les informations du message qui transporte cet objet. Il peut apparaître dans les messages *Resv*, *ResvErr* ou *ResvTear*.
RESV_CONFIRM	Contient l'adresse IP du récepteur qui demande une confirmation de réservation. Il peut apparaître dans les messages *Resv* ou *ResvConf*.

Tableau 4 : Les objets de RSVP.

RSVP utilise sept messages. Les plus importants pour l'établissement d'une session RSVP sont les messages *Path* et *Resv*. Les autres sont introduits soit pour signaler une erreur (*PathErr*, *ResvErr*) soit pour annoncer une confirmation de réservation (*ResvConf*) soit pour

terminer explicitement la session RSVP (*PathTear*, *ResvTear*). Le Tableau 5 résume la fonction de chacun des ces messages et la leurs directions.

Message	Description
Path	Ce message est envoyé par l'émetteur qui initie la session RSVP vers les récepteurs qui se trouvent aux feuilles de l'arbre de la route Multicast/Unicast fournie par le protocole de routage et laquelle les paquets de données vont emprunter. Il crée un état de route « Path State » qui contient l'adresse du nœud précédent sur chacun des routeurs qu'il traverse. L'ensemble de ces « Path State » constitue la route inverse du récepteur vers l'émetteur.
Resv	Ce message est envoyé par le récepteur vers les émetteurs en suivant la route inverse qui a été tracée par le message *Path*. Il crée et maintient un état de réservation « Reservation State » sur chacun des routeurs qu'il traverse. A sa réception par l'émetteur, il est utilisé pour mettre en place les paramètres du flot.
PathErr	Ce message signale une erreur lors de l'établissement du chemin. Il est envoyé à l'émetteur comme réponse à son message *Path* qui a envoyé et qui a causé l'erreur.
ResvErr	Ce message signale une erreur lors de la réservation de ressources. Il est envoyé au récepteur comme réponse à son message *Resv* qui a envoyé et qui a causé l'erreur.
PathTear	Ce message efface explicitement le « Path State » qui se trouve sur chacun des routeurs qu'il traverse.
ResvTear	Ce message efface explicitement le « Reservation State » qui se trouve sur chacun des routeurs qu'il traverse.
ResvConf	Ce message est envoyé au récepteur initiant la réservation de ressources pour indiquer que la réservation a été effectuée. *ResvConf* est envoyé seulement à la demande du récepteur et n'est pas traité par les nœuds qu'il traverse.

Tableau 5 : Les messages de RSVP

Périodiquement, chaque émetteur[10] envoie un message *Path* pour chaque flot qu'il génère (Diagramme de séquence V-1). Le message *Path* contient un objet **SENDER_TSPEC** qui est la description du format des paquets de données que l'émetteur souhaite envoyer et un objet **SENDER_TSPEC** qui définit les caractéristiques du flot que l'émetteur va générer. **SENDER_TSPEC** est utilisé par l'organe Contrôle de Trafic pour empêcher la surréservation et éviter des échecs de réservations inutiles. Le message *Path* peut aussi transporter un objet **ADSPEC** qui est mis à jour par le Contrôle de Trafic sur chacun des routeurs traversés par le message *Path*. Cet objet est utilisé pour réaliser le mécanisme OPWA.

En réponse au message *Path*, chaque récepteur fait une demande de réservation en envoyant le message *Resv* vers les émetteurs de la sélection de l'objet **SCOPE**. Ce message va suivre exactement le (ou les) chemin inverse spécifié par les « Path State » crées par le message *Path*. Il transporte l'objet **STYLE** qui définit le type de réservation et un descripteur de flot. Selon le type de réservation, le descripteur de flot contient un ou plusieurs objets **FLOWSPEC** qui contiennent eux-mêmes les paramètres de la QoS demandée ; le descripteur de flot contient aussi un ou plusieurs objets **FILTER_SPEC** qui décrivent les paquets de données qui doivent recevoir la QoS et il peut aussi ne contenir aucun **FILTER_SPEC**.

Les messages *PathTear* et *ResvTear* peuvent être envoyés soit par le terminal (application ou système) soit par le routeur dans le cas où le temps de « cleanup timeout » est expiré. *PathTear* traverse le réseau de son point d'initiation vers tous les récepteurs en effaçant tous les « Path State » correspondants à l'objet **SENDER_TEMPLATE** et en ajustant les « Reservation State » tout au long du chemin. *ResvTear* traverse le réseau de son point d'initiation vers tous les émetteurs en effaçant tous les « Reservation State » correspondants à l'objet **STYLE** et au descripteur de flot qu'il transporte.

[10] émetteur c'est celui qui initie la session RSVP et le récepteur est celui qui initie la réservation en réponse à l'émetteur.

Les messages *PathErr* et *ResvErr* contiennent l'objet **ERROR_SPEC** qui définit l'erreur. Le *PathErr* est simplement envoyé à l'émetteur qui a causé l'erreur et ne change aucun état dans les routeurs qu'il traverse. Par contre la gestion du message *ResvErr* est assez complexe puisque la demande de réservation qui a échoué peut être le résultat de plusieurs demandes qui sont fusionnées « merged »; dans ce cas le *ResvErr* doit être envoyé vers tous les récepteurs concernés.

Quand un récepteur demande la confirmation d'une demande de réservation, il inclut l'objet **RESV_CONFIRM** qui contient son adresse IP dans le message *Resv*. Dans une réservation Multicast, aux points fusionnés, seulement le plus large **FLOWSPEC** et son objet de confirmation sont envoyés à l'émetteur. Si une demande de réservation d'un récepteur donné parmi le groupe de récepteurs est inférieure ou égale à la réservation installée au point fusionné, son message *Resv* n'est pas transmis au nœud suivant de ce point et si ce message *Resv* inclut une demande de confirmation, le message *ResvConf* est renvoyé à ce récepteur par le nœud du point fusionné (Diagramme de séquence V-1). La demande de confirmation est reçue par l'émetteur dans seulement le cas de la plus large réservation et c'est l'émetteur qui renvoie le message *ResvConf*. La réception du message *ResvConf* signifie uniquement la forte chance d'une réservation accomplie tout au long du chemin puisque le nœud du point fusionné pourrait avoir envoyé le message *ResvConf* avant que la plus large réservation ne soit complètement installée entre le point fusionné est l'émetteur. Dans ce dernier cas le récepteur recevra un message *ResvErr* après avoir reçu un message *ResvConf*. Le message *ResvConf* contient aussi un objet **ERROR_SPEC** contenant l'adresse IP du nœud qui confirme la réservation et un descripteur de flot qui décrit la réservation confirmée.

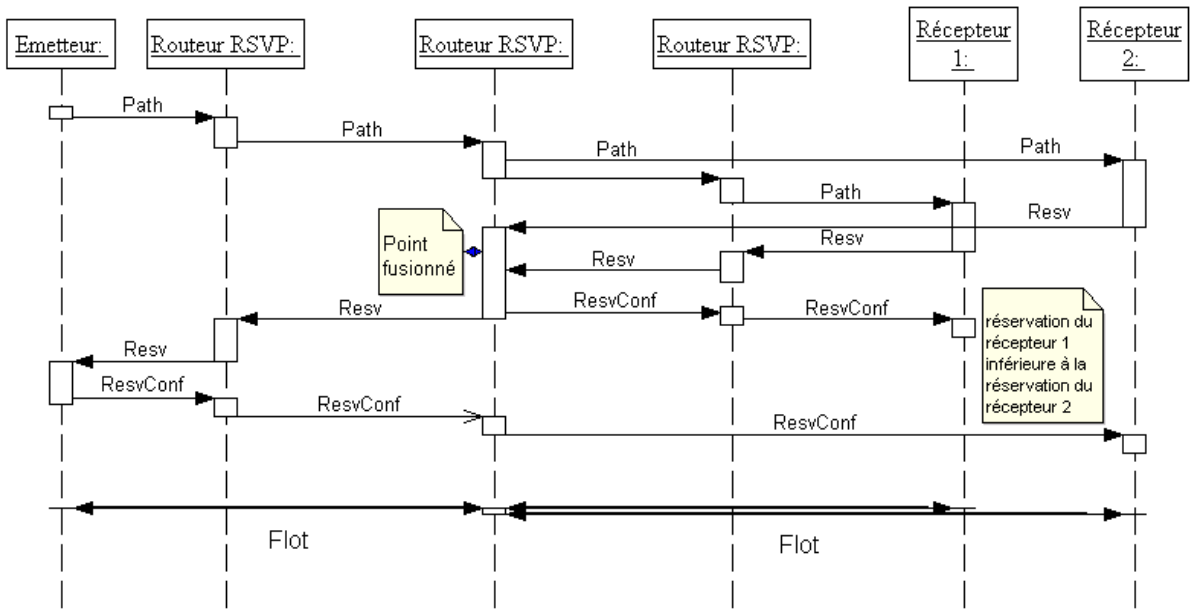


Diagramme de séquence V-1 : communication RSVP.

Annexe B : COPS

Un message COPS se compose d'un en-tête commun et un ensemble d'objets typés. Il est de la forme suivante :

***Message COPS* = [en-tête commun] [objet1]… [objet N]**

L'en-tête commun :

En plus d'un champ pour la version du protocole et d'un champ pour la taille globale du message, l'en-tête commun contient quatre flags dont seulement un est utilisé pour indiquer que le message est sollicité par un autre message COPS et deux autres champs qui sont le code opération qui indique le type du message et le champ type de client qui précise la nature du client. Le type du client permet d'indiquer le mode d'application de COPS et induit des définitions complémentaires par rapport au fonctionnement de base de COPS. Il détermine l'interprétation des objets encapsulés dans les messages. Le protocole COPS définit dix messages qui sont listés et expliqués dans le tableau suivant :

Message	Abr.	**Description**
Request	***REQ***	Il est envoyé par le PEP. Celui-ci établit un état de requête pour le Handle d'un type de client pour lequel le PDP va maintenir un état. Toutes les modifications des états se font avec ce message.
Decision	***DEC***	Il est envoyé par le PDP et contient le Handle associé et un ou plusieurs objets **Decision** selon l'objet **Context** et l'objet **Decision Flags**. S'il y a une erreur protocole, un objet **Error** est envoyé à la place. Un message d'une nouvelle décision pour une nouvelle requête ou une requête de mise à jour a le flag de sollicitation positionné.
Report State	***RPT***	Il est envoyé par PEP pour informer le PDP du succès ou de l'échec de la réalisation de la décision de ce PDP. Il contient un objet **Report-Type** et optionnellement un objet **ClientSI** pour apporter plus de précision pour un type de client. Il est obligatoirement renvoyé après un message *DEC* et le flag de sollicitation positionné. Il est aussi utilisé pour envoyer des rapports des mises à jour périodiques.
Delete Request State	***DRQ***	Il est envoyé par le PEP pour indiquer au PDP que l'état référencé par le Handle n'est plus valide. Il transporte l'objet **Reason code** pour préciser la cause de l'effacement de l'état.
Synchronize State Req	***SSQ***	Il est envoyé par PDP pour qu'il demande la retransmission de ou des états d'un type de client. Si c'est spécifié, c'est seulement l'état relatif à un Handle qui est retransmis.
Client-Open	***OPN***	Il est envoyé par le PEP pour informer le PDP sur les types de clients supportés par le PEP, sur le dernier PDP sur lequel PEP est correctement connecté et/ou sur des négociations spécifiques aux types de clients. Il est envoyé au début et à n'importe quel moment de la connexion même si c'est pour le même type de client.
Client-Accept	***CAT***	Il est envoyé par le PDP pour une réponse positive au message *OPN* envoyé par le PEP. Il est utilisé aussi pour que le PDP informe le PEP sur le Timer qu'ils doivent tous les deux respecter pour envoyer des messages *Keep-Alive*. Il peut aussi transporter un objet **ACCT timer** pour la période des messages de comptabilité**.**
Client-Close	***CC***	Il pourrait être envoyé par le PEP ou par le PDP pour indiquer qu'un type de client particulier n'est plus supporté. Il transporte l'objet **Error** pour informer sur la raison de la fermeture. S'il est envoyé par le PDP, ce dernier peut indiquer une adresse d'un autre PDP.
Keep-Alive	***KA***	Il est envoyé périodiquement par le PEP à un intervalle défini dans l'objet **KA Timer** reçu dans le message *CAT* pour une connexion donnée. Il est généré au hasard entre le premier quart et le troisième

		quart de cet intervalle de temps. Quand un PDP le reçoit, il répond au PEP avec un message du même type. Ce message sert à valider la connexion de la part des deux cotés même s'il n'y a aucun autre message qui est envoyé.
Synchronize Complete	***SSC***	Il est envoyé par le PEP en réponse à un message *SSQ* pour informer le PDP que les anciens états ont bien été synchronisés entre eux pour un type de client donné.

Tableau 6 : Les messages COPS.

Les objets COPS :

Tous les objets encapsulés sont constitués d'un en-tête suivi du contenu de l'objet formé par un ou plusieurs mots de 32 bits. L'en-tête contient un champ qui donne la taille de l'objet, un champ numéro de classe qui définit la classe d'information contenue dans l'objet et un champ type de classe qui détermine la sous-classe ou version de l'information. Les seize classes d'objet sont listées et expliquées dans le tableau suivant :

Classe de l'objet	Description
Handle	Il transporte l'unique valeur qui identifie un état installé et qui est choisie par le PEP. Il est utilisé dans la plupart des messages COPS et pour référencer un état installé dans le PEP et dans le PDP pour un type de client donné. Sa valeur est opaque au PDP. Ce dernier ne fait qu'une comparaison de bit par bit.
Context	Il spécifie le type de l'événement qui a déclenché la requête dans laquelle il est inclus obligatoirement. Il est aussi inclus dans les décisions. Le type de l'évènement pourrait être une requête d'admission de contrôle, d'entrée de message, d'allocation de ressource, de sortie de message ou de configuration.
In-Interface	Il permet d'identifier l'interface d'entrée du PEP sur laquelle la requête est appliquée et l'adresse IP de la source du message de signalisation reçu.
Out-Interface	Il permet d'identifier l'interface de sortie du PEP sur laquelle la requête est appliquée et l'adresse IP du destinataire du message de signalisation à envoyer.
Reason code	Il contient le code qui détermine la raison pour laquelle l'état est effacé. Il figure dans le message *DRQ*.
Decision	Il transporte la décision faite par le PDP et pourrait être spécifique à un type de client particulier. Il y a cinq types de cet objet. Le premier est le « **Decision Flags** » qui a un flag d'erreur et un champ qui indique si la décision est Nulle (pas de configuration disponible), si la requête est admise (ou configuration Install) ou si la requête va être effacée (ou la configuration va être effacée). Le deuxième est un ensemble de données « **Stateless Data** » qui ne sont pas à installer. Ce type d'objet de décision transporte des données additionnelles qui peuvent être utilisées localement par le PEP. Le troisième type est le type de données de remplacement « **Replacement Data**» qui sont utilisées pour remplacer des données dans les messages de signalisation par exemple. Le quatrième est « **Client Specific Decision Data** » qui est le type de données de décision spécifiques au type client. Le dernier est le type « **Named Decision Data**» qui transporte l'objet de décision (données politiques) dans les réponses à la requête de configuration.
LPDP Decision	Il transporte la décision faite dans le PDP local. Il a le même format que l'objet **Decision**.
Error	Il est utilisé pour identifier une erreur particulière du protocole COPS tel un mauvais Handle reçu, format du message incorrect, échec de la communication…
Client Specific Info (ClientSI)	Il contient des informations spécifiques à un type de client. Il peut être de deux types : « **Signaled ClientSI** » comme dans le cas du client COPS-RSVP, et de type « **Named ClientSI** » comme dans le cas du client COPS-PR.
Keep-Alive Timer	Il est utilisé pour spécifier le nombre maximum de secondes qu'une connexion peut rester sans échanges de message COPS avant qu'elle soit considérée invalide.
(PEPID) PEP	Il transporte une chaîne de caractères qui identifie le client PEP d'une manière

Identification	unique dans un domaine de politique.
Report-Type	Il détermine le type de rapport du résultat de l'installation de l'état. Il pourrait indiquer un succès, un échec ou un enregistrement de comptabilité (Accounting).
PDP Redirect Address	Il transporte l'adresse IP et le port TCP d'un PDP pour re-directionner un type de client particulier par le PDP qui envoie cet objet.
Last PDP Address	Il transporte l'adresse IP du dernier PDP sur lequel le PEP s'est correctement connecté pour un type de client.
Accounting Timer (ACCT)	Il contient le nombre de secondes qui détermine l'intervalle de l'émission des enregistrements de comptabilité.
Message Integrity	Il contient la clé secrète et la séquence de numéro utilisée pour assurer l'intégrité des messages COPS. Il est utile aussi pour l'authentification du message.

Tableau 7 : Les objets COPS.

Annexe C : Diameter

Un paquet Diameter est encapsulé dans un paquet TCP ou SCTP et est de la forme :

***Message Diameter* = [en-tête commun] [AVP 1]… [AVP N]**

Un message Diameter consiste en un en-tête commun suivi d'un ou plusieurs AVP (Attribute Value Pairs). Un AVP inclut un en-tête et des données. Il est utilisé pour encapsuler aussi bien les données spécifiques au protocole telles les informations du routage que les informations d'authentification, d'autorisation et de comptabilité. Un message Diameter peut contenir plusieurs AVP.

L'en-tête commun :

En plus du champ version du protocole et du champ taille du message, l'en-tête commun contient des bits flags. Un bit « R » détermine si le message est une requête ou une réponse ; un bit « P » détermine si le message peut passer par un agent proxy, être relayé ou être redirigé ou être traité localement ; un bit « E » détermine si c'est un message d'erreur et un bit « T » qui permet de « re-synchroniser » le client Diameter et le serveur Diameter en cas de coupure de communication et il est positionné dans un message retransmis.

L'en-tête commun contient quatre autres champs qui sont le code de commande (comand-code), l'identificateur de l'application (Application-ID), l'identificateur de saut (Hop-by-Hop Identifier) et l'identificateur de bout-en-bout (End-to-End Identifier). Le champ d'identificateur d'application permet d'identifier l'application qui utilise le protocole Diameter. Ce champ prend la valeur « 0 » pour les messages du protocole Diameter de base, « 1 » pour l'application NAS, « 2 » pour l'application Mobile IPv4 et « 3 » pour les messages de comptabilité. D'autres codes pourront être définis dans le futur mais à condition de faire une demande auprès de l'organisme IANA[11].

Le champ identificateur de saut permet de numéroter les messages à chaque relais ou proxy. Un numéro est choisi au hasard par l'entité qui fait transmettre ou transiter une requête. Le numéro laissé par l'entité précédente dans le message est sauvegardé par l'agent et remplacé par son propre ID. Les sauvegardes des identificateurs de saut constituent un état de transaction (Transaction state). Ce processus se répète tout le long du chemin. Le serveur qui répond à la requête reprend le même numéro pour la réponse. Ainsi, les relais voient transiter les réponses avec les numéros qu'ils attendaient. Pour poursuivre la chaîne, ils remettent l'identificateur de leur prédécesseur qu'ils avaient mémorisé.

Le champ d'identificateur de bout-en-bout contient le numéro globalement unique pour chaque message envoyé par l'émetteur (client ou serveur) et qui est choisi aléatoirement par ce même émetteur. En retour, le correspondant va reprendre le même numéro pour renvoyer la réponse. Ainsi, l'émetteur sait à quelle requête correspond telle ou telle réponse reçue. En cas de même identificateur dans des messages de réponse, les doublons devront être écartés sans traitement.

Chaque message Diameter porte un code (valeur) dans le champ code commande qui détermine le traitement du message. Chaque code de commande est combiné avec le bit flag « R » pour déterminer si c'est une requête ou une réponse. Déduire ainsi les noms des message. Les messages de Diameter sont listés et décrits dans le tableau suivant :

[11] **IANA** : *Internet Assigned Numbers Authority* - L'IANA délègue son autorité à l'InterNIC et à d'autres organisations pour l'allocation d'espaces d'adresses IP et l'assignation de noms de domaines. L'IANA maintient aussi une base de données d'identifiants de protocoles assignés utilisés dans la pile TCP/IP, y compris les numéros de systèmes autonomes.

Nom de la Cde	Abr.	Description
Abort-Session-Request	***ASR***	Il est envoyé par un serveur Diameter à un client Diameter qui fournit le service à l'utilisateur pour l'informer que la session identifiée par **Session-Id** devrait être stoppée.
Abort-Session-Answer	***ASA***	Il est envoyé en réponse à *ASR* et transporte l'AVP **Result-Code** pour indiquer les dispositions prises à la réception du message *ASR*.
Accounting-Request	***ACR***	Il est envoyé par le client Diameter afin d'échanger des informations (enregistrements) de comptabilité avec le serveur Diameter.
Accounting-Answer	***ACA***	Il est utilisé pour accuser réception d'un message *ACR* et transporte le même AVP **Session-Id** reçu dans ce message *ACR*. Seul le serveur Diameter cible, connu comme serveur Diameter mère, répond avec un *ACA*.
Capabilities-Exchange-Request	***CER***	Il est envoyé pour fournir les caractéristiques de la communication et les capacités de l'émetteur (ce qu'il peut supporter comme applications).
Capabilities-Exchange-Answer	***CEA***	Il est envoyé en réponse au message *CEA* et transporte les mêmes informations que celles de *CEA* et le rapport d'une éventuelle erreur.
Device-Watchdog-Request	***DWR***	Il est envoyé quand il n'y a pas de trafic échangé entre deux peer afin d'anticiper la découverte d'un éventuel échec de transport.
Devic-Watchdog-Answer	***DWA***	Il est envoyé en réponse au message *DWR* et accuse la réception de celui-ci.
Disconnect-Peer-Request	***DPR***	Il est envoyé pour informer le destinateur du message de l'intention de fermer la connexion transport.
Disconnect-Peer-Answer	***DPA***	Il est envoyé en réponse au message *DPR* et, à la réception, déclenche la fermeture de la connexion.
Re-Auth-Request	***RAR***	Il est envoyé par le serveur Diameter au client Diameter qui fournit le service ou l'accès pour demander à l'utilisateur de faire une requête de ré-authentification et/ou de ré-autorisation.
Re-Auth-Answer	***RAA***	Il est envoyé en réponse au message *RAR*. Si c'est un succès, la requête de *RAA* est suivie du message de l'application d'authentification et/ou d'autorisation concernée.
Session-Termination-Request	***STR***	Il est envoyé par le client Diameter qui fournit le service ou l'accès pour informer le serveur Diameter que la session d'authentification et/ou d'autorisation est terminée.
Session-Termination-Answer	***STA***	Il est envoyé par le serveur Diameter qui a reçu le message *STR* pour notifier que le la session d'authentification et/ou d'autorisation est terminée.

Tableau 8 : Les messages Diameter

Les AVP :

Toutes les données délivrées par le protocole sont sous forme d'AVP (*Attribute Value Pairs*). Les AVP peuvent être des fonctions d'authentification, d'autorisation, de comptabilité, de sécurité ou de routage. Ils sont utilisés pour transporter les données associées à une application particulière qui utilise Diameter. Quelques AVP sont cependant réservés seulement aux besoins du protocole de base. Le protocole de base est extensible dans le sens où d'autres utilisations dans des anciennes ou dans de nouvelles applications sont possibles en définissant de nouveaux AVP.

Chaque AVP contient un en-tête et des données en formats dérivés (Time, DiameterIdentity, DiameterURI…) des formats de base (OctetString, Float32, Unsigned64…). L'en-tête de l'AVP est composé d'un champ code de l'AVP, d'un champ taille de l'AVP, d'un champ optionnel d'identificateur du vendeur et de bits flags. Les données sont interprétées selon le code défini dans les spécifications.

Le champ d'identificateur du vendeur porte un numéro qui est fourni par l'IANA à la demande. Cela permet, à des constructeurs qui le désirent, de créer leur propre AVP et donc de numéroter de façon unique chaque AVP défini.

Les trois bits flags de l'AVP sont le bit « V » qui indique si le champ Identificateur du vendeur est utilisé, le bit « P » qui indique la nécessité de crypter le message de bout en bout et le bit « M » qui indique si le message doit être rejeté par n'importe quelle entité Diameter.

Le champ code combiné avec le champ Identificateur du vendeur, s'il existe, identifie de façon unique un AVP qui a une fonction précise dans le mécanisme Diameter. Les AVP du numéro 1 jusqu'à 255 sont réservés pour la compatibilité avec RADIUS sans mettre l'identificateur du vendeur. Les AVP numéro 256 et plus sont utilisés pour Diameter et ces numéros sont alloués par l'IANA. Le tableau suivant liste et décrit les AVP du protocole Diameter de base.

AVP	Description
Acct-Interim-Interval	Les informations contenues dans cet AVP sont reçues par le client Diameter de la part du serveur Diameter mère pour dicter comment et quand produire des enregistrements de comptabilité. Si la valeur contenue dans cet AVP est différente de zéro, elle signifie l'intervalle en seconde. Entre l'ouverture et la fermeture de la session de comptabilité et à chaque fois que cet intervalle de temps s'écoule, le client génère un enregistrement de comptabilité de type INTERIM_RECORD pour l'envoyer au serveur.
Accounting-Realtime-Required	Il est inclus dans un message *ACR* envoyé par le serveur Diameter mère vers le client Diameter. Les informations qu'il contient indiquent au client comment réagir dans le cas où ces requêtes de comptabilité n'atteindraient pas le serveur Diameter mère à cause des problèmes temporaires du réseau.
Acct-Multi-Session-Id	Il peut être inclus par le serveur Diameter dans une réponse d'autorisation et doit être inclus dans tous les messages de comptabilité. Il est utilisé pour lier plusieurs sessions de comptabilité qui ont chacune un **Session-Id** unique mais le même **Acc-Multi-Session-Id**.
Accounting-Record-Number	Il contient un numéro pour identifier l'enregistrement de comptabilité au sein d'une session. La combinaison de la valeur de cet AVP avec celle de l'AVP **Session-Id** offre un identificateur d'enregistrement de comptabilité globalement unique.
Accounting-Record-Type	Il contient le code associé au type d'enregistrement de comptabilité (Accounting record = EVENT_RECORD, SRART_RECORD, STOP_RECORD, INTERIM_RECORD)
Accounting-Session-Id	Il est utilisé seulement pour la translation RADIUS/Diameter. Il contient le contenu de l'attribut **Acct-session-Id** de RADIUS.
Accounting-Sub-Session-Id	Il contient l'identificateur de la sub-session de comptabilité. Sa valeur est incrémenté à chaque nouvelle sub-session. La combinaison de sa valeur avec celle de **Session-Id** est un identificateur unique pour chaque sub-session. L'absence de cet AVP signifie qu'il n'y a pas de sub-session mais son absence, dans le message *ACR* avec le type STOP_RECORD, indique la terminaison de toutes les sub-sessions d'un **Session-Id** donné. La sub-session est utilisée pour distinguer les différents services (QoS, transfert de données…) fournis dans la même session.
Acct-Application-Id	Il contient les identificateurs des applications Diameter de comptabilité supportées par l'entité Diameter qui envoie le message. Il est utilisé pour annoncer ces applications.
Auth-Application-Id	Il contient les identificateurs des applications Diameter d'authentification et d'autorisation supportées par l'entité Diameter qui envoie le message. Il est utilisé pour annoncer ces applications dans la phase de négociation.
Auth-Request-Type	Il est inclus dans une requête d'application spécifique pour informer les peer si l'utilisateur sera seulement authentifié, seulement autorisé ou les deux.
Authorization-Lifetime	Il contient le nombre maximum de secondes de l'utilisation du service par l'utilisateur avant qu'il soit ré-authentifié et/ou ré-autorisé. Quand l'AVP **Session-**

	Timeout et l'AVP **Authorization-Lifetime** sont tous deux présents, le temps du **Authorization-Lifetime** doit être supérieur ou égal au temps de **Session-Timeout**.
Auth-Grace-Period	Il contient le nombre maximum de secondes que, après l'expiration du temps **Authorization-Lifetime,** le serveur Diameter va attendre avant de libérer toutes les ressources occupées pour la session.
Auth-Session-State	Il est envoyé dans un message pour indiquer si l'état de la session est ou non maintenu pour une session particulière.
Re-Auth-Request-Type	Il est inclus dans une réponse d'une application Diameter spécifique pour informer le client Diameter qui fournit le service, de l'action attendue par le serveur Diameter après l'expiration du temps **Authorization-Lifetime**. L'action est soit l'authentification soit l'autorisation soit les deux.
Class	Il est utilisé par le serveur Diameter pour envoyer au client Diameter des informations d'état dans les messages réponse d'autorisation d'une application spécifique. Quand un ou plusieurs de cet AVP sont présents dans le message de réponse d'autorisation, ils doivent être présents dans tous les messages subséquents tels les messages de re-autorisation, de terminaison de session et de comptabilité.
Destination-Host	C'est le nom de la machine destinataire. Il peut être présent dans un message requête mais n'est jamais présent dans un message réponse. Son absence dans un message signifie que le message peut être envoyé à n'importe quel serveur qui supporte l'application et qui est dans le domaine spécifié par l'AVP **Destination-Realm.**
Destination-Realm	Il contient le nom du domaine vers lequel le message sera routé et est utilisé pour les décisions de routage. Il n'est jamais présent dans un message réponse.
Disconnect-Cause	Il est inclus dans le message *DPR* pour transporter la raison pour laquelle la connexion sera fermée.
E2E-Sequence AVP	Il contient une valeur générée au hasard et un compteur pour la protection contre la réplication de message de bout-en-bout.
Error-Message	Il accompagne l'AVP **Result-Code** et transporte un message d'erreur qu'une personne peut lire et qui pourrait être affiché à l'utilisateur.
Error-Reporting-Host	En cas d'erreur, il contient l'identité de l'entité Diameter qui a envoyé l'AVP **Result-Code**.
Event-Timestamp	Il est envoyé dans *ACR* ou *ACA* pour transmettre l'heure du déclenchement du rapport.
Experimental-Result	Il contient un AVP **Vendor-ID** et un AVP **Experimental-Result-Code** pour indiquer si une requête d'un constructeur particulier a causé une erreur ou a été traitée avec succès.
Experimental-Result-Code	Il contient le code de l'erreur assigné par un constructeur particulier.
Failed-AVP	Il contient des informations de débogage. Le code de l'AVP **Result-Code** fournit la raison de la présence de l'AVP **Failed-AVP.**
Firmware-Revision	Transporte les informations relatives à la révision du programme Diameter implémenté dans la machine de l'émetteur du message.
Host-IP-Address	Transporte une seule adresse IP de l'émetteur. Il figure seulement dans *CEA* et dans *CER*.
Inband-Security-Id	Il est utilisé pour annoncer les types de sécurité supportés par une entité Diameter.
Multi-Round-Time-Out	Il est envoyé dans une réponse d'autorisation d'une application spécifique. Il est utilisé quand le mécanisme d'authentification utilise plusieurs requêtes (plusieurs phases). Cet AVP contient le nombre maximum de secondes que le client Diameter doit laisser à l'utilisateur pour répondre à une requête d'authentification.
Origin-Host	Il identifie l'entité Diameter de l'extrémité qui a envoyé le message. Il n'est pas modifié par les agents Diameter.
Origin-Realm	Il contient le nom du domaine de celui qui a envoyé le message.
Origin-State-Id	Il contient une valeur qui est incrémentée à chaque fois que l'entité Diameter concernée ré-initialise une session qui a un état perdu. Cela se passe quand

	l'équipement d'accès est redémarré. L'entité qui reçoit cet AVP dans un *CER* déduira que l'état de session qui a une valeur de **Origin-State-Id** inférieur celle reçue est expiré et que l'équipement a fermé involontairement la session.
Product-Name	Transporte le nom assigné au constructeur qui a fournit le produit (l'application).
Proxy-Host	Contient l'identité de l'agent de proxy qui a ajouté un AVP dans **Proxy-Info**.
Proxy-Info	Regroupe **Proxy-Host** et **Proxy-State** ou plusieurs de ces AVP.
Proxy-State	Il contient des informations d'état locales au proxy concerné. Il est traité comme données opaques au protocole.
Redirect-Host	Il identifie le peer vers lequel la requête sera re-directionnée par l'agent de re-direction.
Redirect-Host-Usage	Il dicte la façon dont l'entrée de la table de routage créée par **Redirect-Host** sera utilisée (pour toutes les sessions, pour toutes les applications…).
Redirect-Max-Cache-Time	Il contient le nombre maximum de secondes que l'entrée de la table de routage créée avec l'AVP **Redirect-Host** est sauvegardé.
Result-Code	Il indique si une requête est traitée avec succès ou une erreur a été détectée. Le code qu'il transporte décrit l'erreur qu'une entité Diameter a rencontrée au cours du traitement de la requête. Il est présent que dans les messages réponse.
Route-Record	Contient la valeur de l'identité reçue dans l'AVP **Origin-Host** des messages *CER* ou *CEA*. Cet AVP est utilisé pour détecter les boucles dans le routage.
Session-Id	Il contient l'identifiant de la session. Il n'est présent qu'une fois et dans tous les messages. Il reste le même pendant toute la durée de la session.
Session-Timeout	Il contient le nombre maximum de secondes de service fourni pour l'utilisateur avant que la session ne soit terminée. Quand l'AVP **Session-Timeout** et l'AVP **Authorization-Lifetime** sont tous deux présents, le temps du **Authorization-Lifetime** doit être supérieur ou égal au temps de **Session-Timeout**.
Session-Binding	Il est envoyé dans un message réponse d'authentification et/ou d'autorisation. S'il est présent, il informe le client Diameter que toutes les requêtes de re-authentification et de ré-autorisation d'une application spécifique pour cette session vont être envoyées vers le même serveur.
Session-Server-Failover	Il est envoyé dans un message réponse d'autorisation d'une application particulière et ne peut pas être présent si l'AVP **Session-Binding** est présent. Cet AVP informe le client Diameter que, si le message de demande de re-authentification et/ou de ré-autorisation ou le message *STR* échoue, ce client Diameter doit envoyer les messages subséquents sans l'AVP **Destination-Host**. Le code qu'il contient détermine l'action à entreprendre dans ce cas.
Supported-Vendor-Id	Il transporte le numéro assigné à un constructeur différent de celui qui a fourni l'implémentation qui supporte les AVP de ce constructeur. Il est utilisé dans les messages *CER* et *CEA*.
Termination-Cause	Il contient un code pour indiquer la raison pour laquelle la session a été terminée.
User-Name	Il contient le nom de l'utilisateur.
Vendor-Id	Il transporte le numéro assigné à un constructeur d'une application Diameter. En combinant cet AVP avec **Supported-Vendor-Id**, cela permet de connaître quels AVP de constructeur peuvent être envoyés au peer. Sa combinaison avec **Product-Name** et **Firmware-Revision** fournit des informations pour le débogage.
Vendor-Specific-Application-Id	Il regroupe un AVP **Vendor-Id**, un AVP **Auth-Application-Id** au plus et un AVP **Acct-Application-Id** au plus ou plusieurs de ces AVP.

Tableau 9 : Les AVP du protocole Diameter de base

Annexe D : EAP

Un message EAP se constitue d'un en-tête qui contient le champ Code, le champ Identifiant et le champ Taille. Le champ Taille est utilisé pour déduire la taille du message et le champ Identifiant pour faire correspondre les messages qui appartiennent à la même session d'authentification. Au champ Code, quatre valeurs lui sont associées: la valeur « 1 » pour indiquer un message *Requête*, la valeur « 2 » pour indiquer un message *Réponse*, la valeur « 3 » pour indiquer un message *Succès* et enfin la valeur « 4 » pour indiquer un message *Echec*. Pour les messages *Requête* et *Réponse*, sauf indication, l'en-tête est suivi d'un bloc de données mais les messages *Succès* et *Echec* ne contiennent que l'en-tête EAP.

***Requête* :**

Elle est envoyée par le serveur d'authentification et demande à l'utilisateur de fournir une information précise comme son identité ou bien une preuve de cette identité (adresse e-mail, login etc.) ou d'autres informations selon la méthode d'authentification choisie par le serveur (mot de passe, certificat électronique, clé etc.). Son bloc de données a un champ Type qui détermine ce qui est envoyé dans la *Requête*. Des *Requêtes* additionnelles sont envoyées jusqu'à la réception d'une *Réponse* valide, l'expiration d'un compteur décrémenté ou l'indication d'échec de communication signalé par le protocole inférieur. Les *Requêtes* retransmises ont le même Identificateur que celui de la *Requête* originale afin de les distinguer des autres nouvelles *Requêtes*. Tout nouveau message de *Requête* modifie le champ Identifiant ; ce qui signifie une nouvelle session d'authentification.

***Réponse* :**

Les messages *Réponses* sont toujours envoyés par le suppliant en réponse au message *Requête* valide et ne sont jamais retransmis après l'écoulement d'un Timer. L'identificateur d'une *Réponse* correspond à l'identifiant de la *Requête* à laquelle elle réplique. Le contenu de ce message dépend de la méthode d'authentification requise par le serveur. Si le suppliant reçoit une *Requête* valide à laquelle il a déjà répondu avec une *Réponse*, il retransmet la même *Réponse* sans aucun traitement du message *Requête*. Les *Requêtes* sont traitées dans l'ordre de leurs arrivées et le traitement d'une *Requête* n'est commencé qu'à la fin du traitement de la *Requête* précédente.

***Succès* :**

Le message *Succès* est envoyé par le serveur AAA après l'accomplissement de l'exécution d'une méthode EAP pour indiquer au suppliant qu'il a été correctement authentifié. Au passage de ce message, l'authentificateur ouvre l'accès au réseau. Le suppliant qui reçoit un message *Succès* incohérent doit le refuser afin d'éviter toute communication avec un authentificateur douteux qui pourrait envoyer un message *Succès* pour faire croire que c'est une authentification mutuelle et entamer une communication avec le suppliant.

***Echec* :**

Ce message est envoyé par le serveur AAA qui ne peut pas authentifier le suppliant à cause d'une *Réponse* inacceptable à une ou au plusieurs *Requête* ou après l'accomplissement d'une méthode d'authentification sans succès. Un serveur pourrait envoyer plusieurs *Requête*s avant d'envoyer un message *Echec* afin de tolérer les erreurs de frappe d'une personne par exemple.

***Les Types* :**

Le bloc de données du message EAP ne contient aucun ou contient plusieurs octets. Son format dépend de la valeur du champ Code. Le premier octet du bloc de données, s'il existe, est le champ Type qui détermine la structure du reste du bloc dans le message. L'EAP définit sept Type. Les trois premiers Type (*1,2 et 3*) sont des cas particuliers. Le reste des types (*4,5,6 et 254*) définit les échanges d'authentifications entre les différentes méthodes EAP. A

chaque fois qu'on introduit une nouvelle méthode d'authentification EAP, cette méthode prend une nouvelle valeur pour un nouveau Type.

Les types définis dans la spécification de EAP [42] sont :
-1- Identity
-2- Notification
-3- Nak
-4- MD5-Challenge
-5- One-Time Password (OTP)
-6- Generic Token Card (GTC)
-254- Expanded Types

Tous les Types sont valides pour les *Requêtes* et les *Réponses* sauf le Type Nak et le Type Expanded Nak qui ne sont valides que pour les *Réponses*. Une implémentation EAP supporte au moins les quatre premiers types.

Le Type Identité est utilisé par la *Requête* pour demander l'identité du suppliant. Généralement, l'authentificateur envoie la *Requête-Identité* comme *Requête* initiale. Dans le bloc de données, l'authentificateur peut inclure un message qui pourrait être affiché pour qu'il soit lu par l'utilisateur. Une *Réponse-Identité* est toujours renvoyée en réponse à une *Requête-Identité*. Souvent, le suppliant acquiert l'identité par la saisie faite par l'utilisateur. Si l'identité est inconnue le bloc de données dans le message de *Réponse-Identité* ne contiendra aucun octet. Les méthodes d'authentification peuvent parfois souhaiter accéder à l'identité fournie par le suppliant ; l'implémentation EAP peut permettre dans ce cas aux méthodes l'accessibilité aux messages *Requête-Identité* et *Réponse-Identité*. Toutefois, une méthode aura toujours son propre mécanise et ne doit pas compter sur *Réponse-Identité* qui transporte l'identité en texte clair (sans cryptage).

Le Type Notification est optionnel et est utilisé pour transporter de l'authentificateur au suppliant un message à afficher. A moins que la méthode ne le permette pas, une *Requête-Notification* pourrait être envoyée à n'importe quel moment où il n'y a pas de *Requête* et avant l'achèvement de la méthode EAP. Le suppliant répond toujours à une *Requête-Notification* avec une *Réponse-Notification* dont le bloc de données ne contient aucun octet. La *Réponse-Notification* est utilisée seulement pour confirmer que le suppliant a bien reçu la *Requête-Notification* et sa réception ne signifie pas que le suppliant a traité la *Requête-Notification*. Le suppliant doit afficher ce message à l'utilisateur et s'il ne peut l'afficher, il doit au moins l'enregistrer dans les logs. La *Requête-Notification* est prévue pour fournir une notification sur un point précis et impératif ; à titre d'exemple l'information porte sur un mot de passe qui est sur le point d'expirer, une mise en garde sur l'échec possible de l'authentification etc. Dans la plupart des cas, la notification n'est pas vraiment nécessaire. Le contenu de la réponse de notification et de la requête de notification ne sont pas disponibles aux autres méthodes.

Le Type Nak est utilisé seulement dans une *Réponse*. Une *Réponse-Nak* est envoyée par le suppliant pour répondre à une *Requête* dont le type de méthode EAP n'est pas supporté par ce suppliant. Le bloc de données pourrait contenir une ou plusieurs valeurs de Types de méthode EAP supportées par le suppliant comme des propositions. S'il ne contient aucun type, cela indique que le suppliant n'a aucune alternative et de ce fait, le serveur d'authentification n'aura pas besoin d'envoyer d'autre *Requête* après la réception d'une *Réponse-Nak*. Si le suppliant supporte Expanded Type, il pourra inclure la valeur de ce Type dans la *Réponse-Nak*. Le Type Expanded Nak n'est aussi valide que dans la *Réponse* mais il est envoyé seulement en réponse à la *Requête-Expanded-Types* dont le Type de l'authentification n'est pas accepté par le suppliant. Ce Type utilise le même format que celui du Type Expanded Types et contient un ou plusieurs Types de méthodes supportés par le suppliant selon le

format utilisé dans la méthode. Nak ou Expanded Nak sont utilisés pour la négociation d'une méthode. Le contenu de ces deux derniers messages n'est pas disponible aux autres méthodes.

Puisque plusieurs des utilisations de EAP sont spécifiques aux constructeurs, les méthodes de Type Expanded Types sont introduites pour permettre à ces constructeurs de supporter leurs propres types non compatibles à l'utilisation générale. Ce Type est aussi utilisé pour étendre les types de méthodes appartenant aux types de méthodes initiales (entre 4 et 254).

Les messages *Succès*, *Echec*, *Réponse-Nak*, *Réponse-Expanded-Nak*, *Requête-Notification* et *Réponse-Notification* ne sont pas utilisés pour transporter des données destinées aux méthodes d'authentifications et leurs données ne sont pas utilisées par les méthodes EAP.

***Les méthodes EAP* :**

Les méthodes EAP sont très variées. Les *Requêtes* et les *Réponses* sont typées selon la méthode utilisée dans une conversation EAP pour faire transporter les données de cette méthode afin d'aboutir à un succès ou à un échec d'authentification. Chaque méthode EAP utilise son mécanisme et chacune a ses avantages et ses inconvénients. Nous décrirons brièvement les méthodes les plus connues. Cette description n'a pas pour objectif d'analyser en détail toutes les méthodes mais seulement de permettre d'avoir une idée claire sur les données transportées avec le protocole EAP.

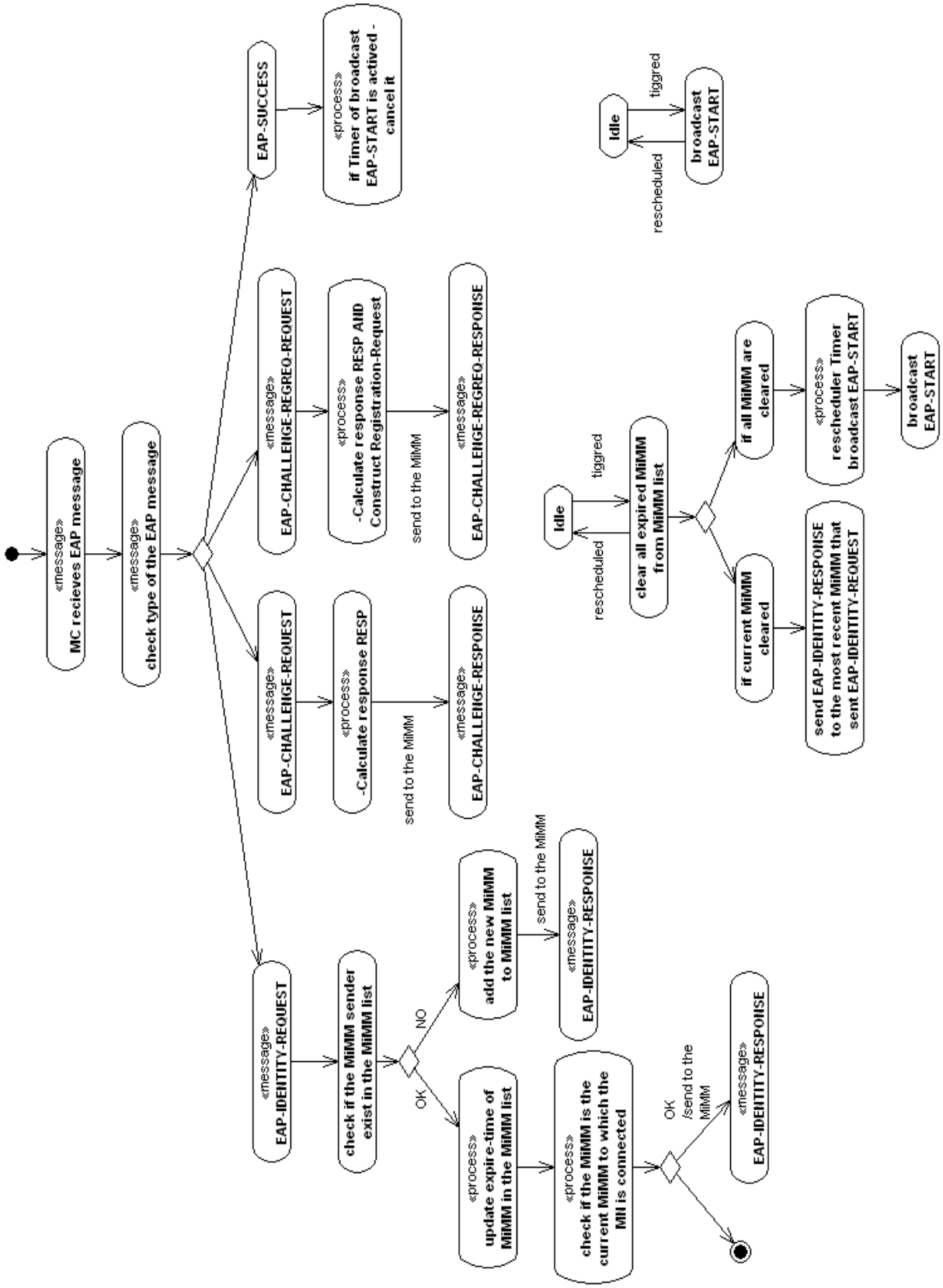
La méthode EAP/MD5-Challenge a le même principe que celui de la méthode d'authentification CHAP ([68]) utilisée dans le protocole PPP. Le serveur d'authentification commence par envoyer au suppliant dans une *Requête* une valeur aléatoire (16 octets) nommée défi (Challenge) ainsi qu'un compteur incrémenté à chaque envoi d'un défi. Le suppliant va faire passer le compteur, son mot de passe et le défi au travers de l'algorithme de hachage (fonction Hache) MD5 [43]. Le résultat de ce calcul est une séquence de bits pseudo-aléatoire ; il est appelé le Hash (16 octet). Le suppliant envoie ensuite ce Hash dans une *Réponse* au serveur d'authentification qui peut faire le même calcul pour vérifier si le résultat concorde avec celui du suppliant. Dans ce genre d'authentification, on trouve aussi la méthode EAP/MS-CHAP-v2 qui, en plus du mécanisme EAP/MD5-Challenge, permet de stocker et d'utiliser le mot de passe après l'avoir fait passer par une autre fonction Hache (pas en texte en clair).

La méthode EAP/OTP utilise le système One Time Password (OTP, [51]). Un OTP est un mot de passe conçu pour n'être utilisé qu'une seule fois afin éviter une autre utilisation si jamais ce mot de passe a été piraté. Le serveur d'authentification commence par envoyer au suppliant un défi dans une *Requêt*e. Ce défi contient quelques octets aléatoires et un index qui change à chaque nouveau défi. Le suppliant utilise un générateur (logiciel) pour produire un OTP. Pour cela le suppliant fournit le défi et son mot de passe (nommé phrase secrète ou Passphrase) au générateur et, en fonction de l'index, ce générateur fait passer plusieurs fois ces informations dans une fonction Hash. Le résultat est un OTP que le suppliant envoie au serveur dans une *Réponse*. Pour vérifier l'authentification, le serveur refait le même calcul et compare son résultat avec celui qui est reçu.

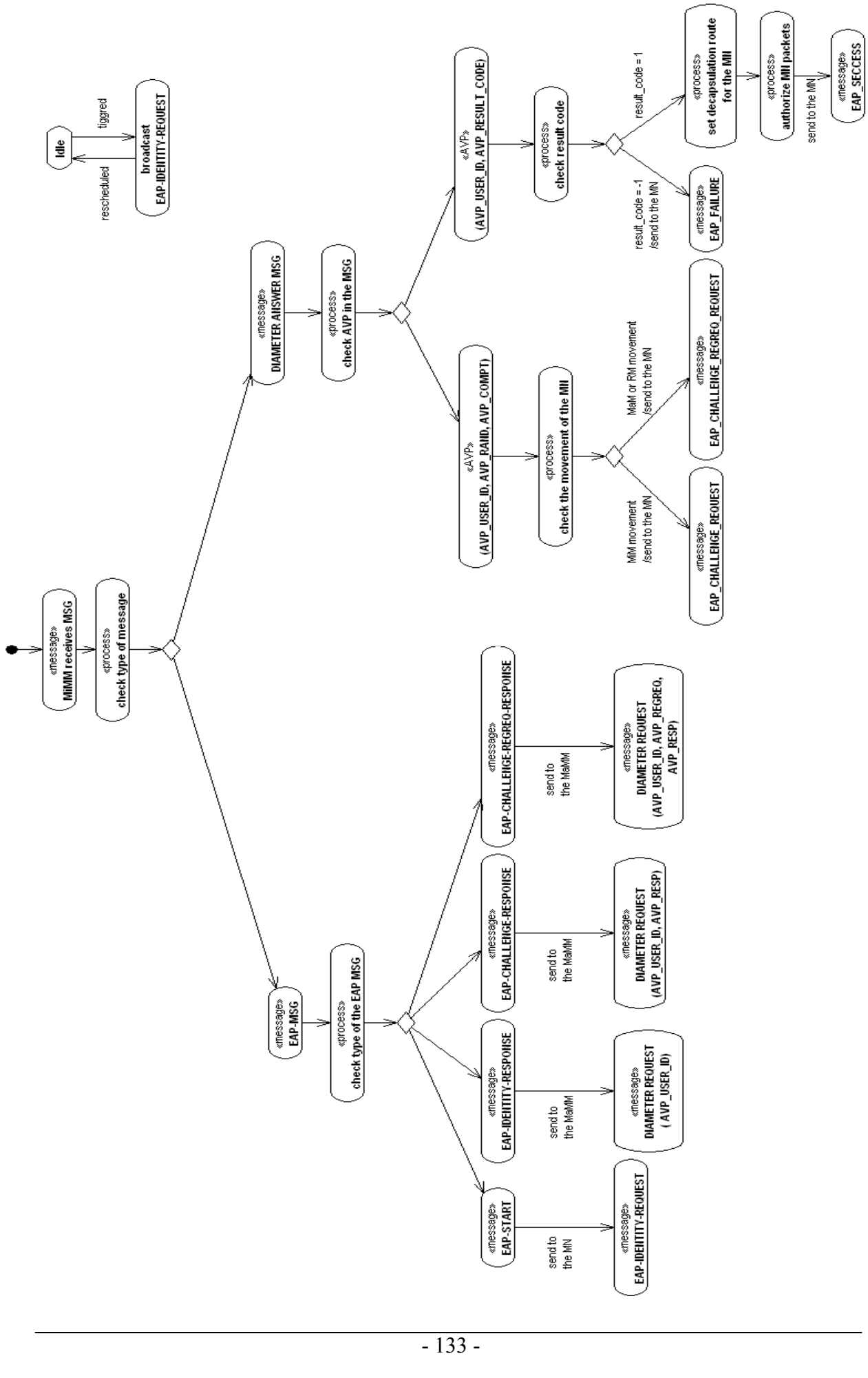
La méthode EAP/GTC (carte à jeton générique) utilise un jeton. Un jeton est une clé assez longue qui n'est connue que par le serveur d'authentification. Cette clé est conservée dans une carte à puce par exemple. Pour ce connecter, l'utilisateur doit avoir sa carte avec lui. Généralement, avec cette méthode, l'utilisateur doit avoir aussi un mot de passe ou un code. Le serveur envoie le défi dans une *Requête* au suppliant. Ce dernier fait passer le jeton, le mot de passe de l'utilisateur et le défi dans une fonction Hash. Le Hash est envoyé dans une *Réponse* au serveur d'authentification qui fait le même calcul pour vérifier l'authenticité de l'utilisateur.

D'autres méthodes EAP conçues et proposées se basent sur l'utilisation d'un tunnel sécurisé. L'une des ces méthode qui est EAP/TLS [53] utilise la phase de négociation de l'établissement d'un tunnel sécurisé avec le protocole TLS (Transport Layer Security, [52]). Cette méthode repose sur cette phase qui est l'identification par certificat. Pour utiliser EAP/TLS, il faut créer et installer un certificat électronique et sa clé privée correspondante sur le serveur d'authentification. Lors du dialogue EAP, le client et le serveur s'échangent ces informations en utilisant les *Requêtes* et *Réponses*. Une autre méthode nommée EAP/PEAP (Protected EAP, [54]) qui signifie l'EAP protégé utilise le protocole TLS. Cette méthode commence par mettre en place un tunnel TLS entre le client et le serveur puis le suppliant et l'authentificateur entament une nouvelle négociation d'une méthode EAP (EAP/MD5-Challenge, EAP/OTP…) au sein de ce tunnel qui protège les données d'authentification. Une autre méthode nommée EAP/TTLS (Tunneled TLS [54]) est la même que celle d'EAP/PEAP. La différence entre ces méthodes réside dans le fait que dans la méthode EAP/TTLS, ce n'est pas seulement les méthodes EAP qui sont utilisées dans le tunnel sécurisé mais n'importe quelle méthode d'authentification. Avec cette dernière méthode on peut même transporter un mot de passe en texte en clair, ce qui n'est pas dans les méthodes EAP.

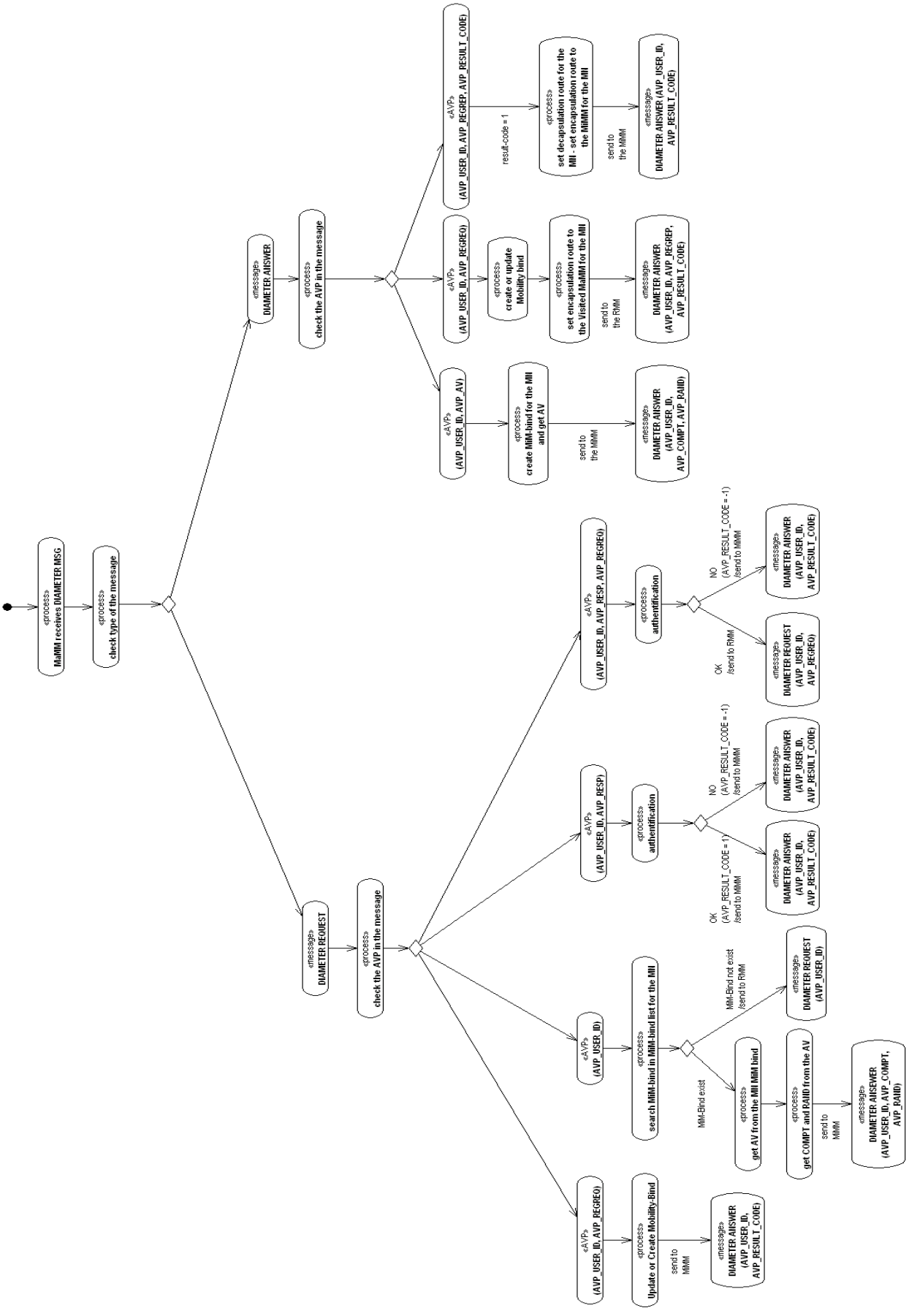
Annexe E : Organigrammes de l'implémentation de UNISIG



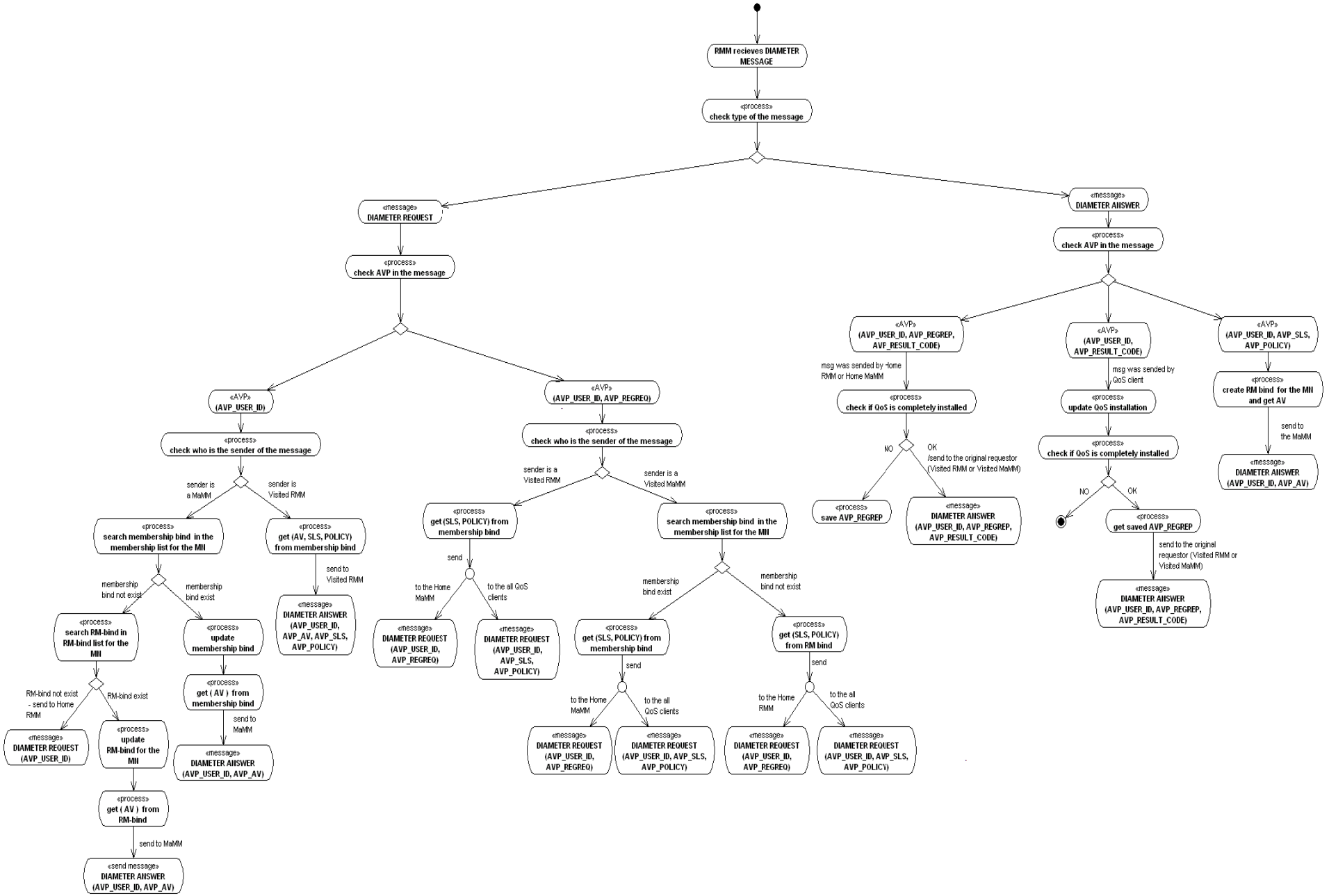
Organigramme 1: fonctionnement du



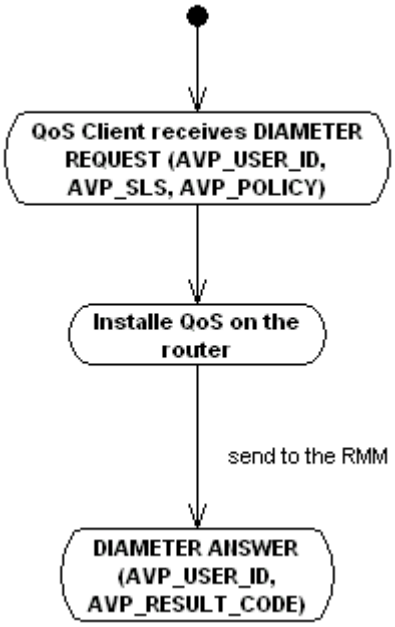
Organigramme 2 : fonctionnement de MiMM
nigramme 2 : fonctionnement de MiMM module



Organigramme 3: fonctionnement du MaMM
nigramme 3: fonctionnement du MaMM dans



Organigramme 4: fonctionnement du RMM



Organigramme 5 : fonctionnement du QoS client sur un routeur de bord DiffServ