Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE ABOU BEKR BELKAID TLEMCEN
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE GENIE ELECTRIQUE ET ELECTRONIQUE

Laboratoire STIC

Ecole doctorale des
Sciences et Technologies de l'Information et Télécommunication

Mémoire de Magister en
Systèmes des Réseaux de Télécommunication

THEME

Évaluation des Performances de Mobile SCTP dans l'Equilibrage de Charge basée sur le Type de Flux dans les réseaux NEMO

Présenté par

Mohamed Rabbie Naimi

Soutenu en Février 2012 devant le jury composé de :

Président : Mr. CHIKH Mohamed Amine, MCA, UABB Tlemcen

Examinateur : Mme DIDI Fedoua, MCA, UABB Tlemcen

Examinateur : Mr KADRI Benamar, MCB, UABB Tlemcen

Encadreur : Mr. FEHAM Mohammed, Pr., UABB Tlemcen

Co-encadreur : Mr ABDELMALEK Abdelhafid, MAA, UABB Tlemcen

Dédicace

Je dédie ce travail à mes parents et à tous ceux qui se sacrifient pour le bien de leurs enfants, j’espère être digne de leur effort.

A tous les membres de ma famille.

A tous mes amis.

Résumé

L’adresse IP est devenue une fonctionnalité clé dans l'architecture actuelle d'Internet. Cependant, la conception de base de la pile d'Internet Protocol est basée sur une architecture formée il y a des années, quand l'Internet était un réseau statique. Les adresses IP fixes n'ont posé aucun problème jusqu'à ce que les appareils de communication sont devenus mobiles. Quand un nœud mobile (MN) communique avec un autre nœud via une association active, il peut avoir accès à plusieurs point d'attachement de différentes technologies, son adresse IP actuel change et par conséquent sa session active se termine. Différentes techniques sont mises en application pour soutenir la mobilité utilisant la pile d'Internet Protocol.

L'IP mobile soutient la mobilité à la couche réseau, le SCTP soutient la mobilité à la couche transport. Quelle couche convient à la mobilité est toujours une question ouverte. Le protocole SCTP présente une caractéristique multihoming avec la reconfiguration dynamique d'adresse (DAR), pour fournir une solution plus sécurisée avec des pertes presque négligeables de paquet. Et d’un autre côté, il permet d’activer plusieurs sessions pour n’utiliser qu’une seule.

Dans ce projet, après avoir présenté les caractéristiques du protocole SCTP, nous avons proposé une approche d’équilibrage de charge dont le but est de réduire les délais de transmission et d’optimiser l’utilisation des ressources disponibles (Débit) en exploitant les différentes sessions ouvertes en même temps. Nous avons ensuite présenté les étapes de modélisation et l’environnement de simulation. Enfin, nous avons évalué les performances du protocole SCTP vis à vis de l’équilibrage de charge.

Mots clés :

Mobilité, SCTP, multihomming, équilibrage de charge, évaluation, simulation.

Abstract

IP address has become a key feature in the current internet architecture. However, the basic design of Internet Protocol stack is based on architecture shaped years ago, when internet was a static network. Fixed IP addresses did not cause any problems until communication devices became mobile. When a Mobile Node (MN) in an active association with another node, It can have access to many attachment points of different technologies, here current ip address changes, consequently its active session terminates. Different techniques are implemented to support mobility using Internet Protocol Stack.

Mobile IP supports mobility at network layer, SCTP supports mobility at transport layer. Which layer is suitable for mobility is still an open issue. Stream Control Transmission Protocols (SCTP) exploits its multihoming feature along with Dynamic Address Reconfiguration (DAR), to provide a solution with higher security and almost negligible packet losses. In addition it permits to activate more sessions in order to use only one.

After presenting sctp protocol characteristics in this project, we have suggested an approach of load balancing in order to reduce transmition delays and improve the use of available resources () in exploiting different open sessions in the same time. Then we have presented modeling stapes and simulation environment. Finally we have evaluated protocol SCTP performances for load balancing.

Keywords :

Mobility, SCTP, multihoming, load balancing, evaluation, simulation.

:

الملخص

سمح الانترنت ظهور مفاهيم جديدة, نميز منها إدارة الأجهزة المتنقلة و هي تتمثل في الحفاظ على صلة الجهاز المتنقل بالانترنت و هدا رخم تحركه المستمر. العديد من التكنولوجيات هده في مجال الدراسات.
قد حاولنا من خلال هدا المشروع اقتراح بروتوكول "أساسي ت ب" الذي لم يقدم فحسب إدارة الأجهزة المتنقلة بل أضاف إمكانية موازنة الضغط الذي قمنا بتقييمه.

الكلمات المفتاحية

الأجهزة المتنقلة, أساسي ت ب, موازنة الضغط, محاكاة.

Remerciements

A.Mr FEHAM Mohammed, Pr., UABB Tlemcen

Vous avez contribué énormément à la réalisation de ce travail. Votre compétence et l'étendu de vos connaissances m'ont toujours inspiré.

A.Mr ABDELMALEK Abdelhafid, MAA, UABB Tlemcen

Il m'est agréable de vous exprimer ma reconnaissance et mes remerciements pour votre aide précieuse, vos conseils, vos suggestions et votre disponibilité.

J'adresse mes vifs remerciements à tous les enseignants du laboratoire STIC qui ont contribué à ma formation.

Je tiens aussi à exprimer l'honneur qui m'est fait par les membres du jury en acceptant d'évaluer mon travail.

Table des matières

CHAPITRE III

Extension de Mobile SCTP avec le Support d'Equilibrage de Charge

CHAPITRE IV

Implémentation sous NS2 de mSCTP avec Equilibrage de charge Résultats de Simulation

Liste des figures

Liste des tableaux

Tableau **Page**

Acronymes

A	
AAA	Authentication, Authorization, and Accounting
ACK	Acknowledgement
AMM	Address Management Module
ASCONF	Address Configuration Change Chunk
ASCONF-Ack	Address Configuration Acknowledgement
API	Application Programming Interface
AR	Access Router
ARP	Address Resolution Protocol
Awk	Alfred Aho, Peter Weinberger et Brian Kernighan
B	
BA	Binding Acknowledgement
BU	Binding Update
C	
CBR	Constant Bit Rate
CN	Correspondant Node
CoA	Care-of Address
CoT	Care-of Test
CoTI	Care-of Test Initiate
CSCTP	Cellular SCTP
CWND	Congestion Window
D	
DAD	Duplicate Address Detection
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol for IPv6
DNS	Domain Name Server
E	
EC	Error Counter
E-TCP	Extended TCP

F

FMIPv6	Fast handover Mobile IPv6
FTP	File Transfert Protocol

H

HA	Home Agent
HA	Host Agent
HB	HeartBeat
HMIPv6	Hiérarchique Mobile IPv6
HoT	Home Test
HoTI	Home Test Initiate
HTML	Hypertext Markup Language

I

ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	Internet Protocol Security
IPv4	Internet Protocol, version 4
IPv6	Internet Protocol, version 6
ISO	International Standard Organisation
ITU	International Telecommunication Union

L

L2HC	Layer 2 Handover Completion
L2SS	Layer 2 Signal Strength

M

MAC	Message Authentication Code
mIP	Mobile IP
MN	Mobile Node
mIPv4	Mobile Internet Protocol version 4
mIPv6	Mobile Internet Protocol version 6
MPEG	Moving Picture Experts Group
mSCTP	Mobile SCTP
MTU	Maximum Transmission Unit

N

NAT	Network Address Translation
NIC	Network Interface Card
NS2	Network Simulator 2

O

OSI	Open System Interconnection

P

PBA	Partial Bytes Ack
PEL	Protocol Engineering Lab
PMM	Path Management Mechanism
PMTU	Path MaximumTransmission Unit

Q

QoE	Quality of Exeprience
QoS	Quality of Service

R

RA	Router Advertisement
RR	Round Robin
RS	Router Solicitation
RTP	Real-Time Transport Protocol
RTSP	Real Time Streaming Protocol
RTO	Retransmission Time Out
RWND	Receiver Window

S

SACK	Selective Acknowledgement
SCTP	Stream Control Transmission Protocol
SHA-1	Secure Hash Algorithm
SIP	Session Initiation Protocol
SN	Sequence Number
SSN	Stream Sequence Number
SSTHRESH	Slow Start Threshold

T

TCB	Transmission Control Block
TCP	Transmission Control Protocol
TLV	Type Length Value
TSN	Transmit Sequence Number

U

UDP	User Datagram Protocol
ULP	Upper Layer Protocol
UMTS	Universal Mobile Telecommunications System

V

VBR	Variable bit rate
VoIP	Voice over IP

W

WiFi	Wireless Fidelity
WiMax	Worldwide interoperability for Microwave Access
WLAN	Wireless Local Area Network

Introduction générale

Le réseau Internet connaît aujourd'hui un succès extraordinaire. Ses principes ont remarquablement résisté aux changements des technologies et de son utilisation dans le temps. En effet, pendant cette période, le monde de technologies informatiques et de télécommunications n'a cessé de changer. Nous dressons un bilan du contexte actuel :

• On observe une présence accrue, en nombre et pourcentage, d'ordinateurs portables. Dotés d'une richesse multimédia et d'une inter-connectivité étendue, ils sont suffisamment performants pour concourir les machines fixes. En plus des ordinateurs portables classiques, un nouveau genre d'équipements est apparu : les dispositifs intelligents. Équipés de microprocesseurs et mémoire, ils sont capables de produire, stocker, manipuler et échanger l'information. Aujourd'hui, la variété de ces dispositifs est impressionnante : tablettes PC, assistants personnels et téléphones portables jusqu'aux cartes à puce, capteurs ou actionneurs.

• Les techniques de communications sans fil ont évoluées. La largeur de la bande passante et le faible prix d'accès a permis le développement et le déploiement des nouvelles technologies sans fil. Ainsi, on a accès au GPRS, l'UMTS et au WIMAX dans les réseaux de télécommunications, aux réseaux locaux sans-fil IEEE 802.11. Toutes ces nouvelles technologies, complétées par les communications traditionnelles filaires, nous permettent d'être sous la couverture continue de plusieurs types de réseaux.

Dans l'environnement hétérogène des accès réseaux entourant l'utilisateur, cela demande que la connexion des terminaux au réseau, ainsi que le transfert vers d'autres réseaux doivent s'exécuter automatiquement, sans que l'utilisateur ne l'aperçoive.

La principale difficulté vient du fait que les protocoles de l'Internet ont été conçus sans prendre en considération que les utilisateurs et leurs machines peuvent changer leur point d'attachement dans des réseaux hétérogènes. En effet, à l'époque où les protocoles Internet se développaient, et jusqu'au milieu des années 90, la plupart des utilisateurs utilisaient l'Internet à partir d'ordinateurs fixes de leurs institutions. Aujourd'hui, la plupart des utilisateurs sont devenus nomades.

Cependant, dans la plupart des cas, les terminaux et les applications ne peuvent pas continuer à fonctionner sans des opérations supplémentaires comme la reconfiguration des adresses IP ou autres paramétrages réseau. Cette reconfiguration provoque l'interruption des connexions actives et nécessite le redémarrage de certaines applications. Sans être liés toujours à la mobilité, des interruptions et déconnexions de réseau peuvent intervenir aussi, à cause de l'indisponibilité temporaire de la connexion ou de l'arrêt volontaire du terminal.

Le point important dans la vision présentée plus haut est comment assurer un fonctionnement transparent des applications et services réseau, en dépit de la mobilité?

Plusieurs travaux de recherche ont été menés pour rendre l'Internet mobile possible. Entre autres les travaux de la gestion de la mobilité dans la couche réseau et plus sont étudiés et ont donné naissance à de nombreux protocoles (MIP, MIPv6 et mSCTP…).

Dans ce travail, nous nous sommes intéressé au protocole de transport SCTP (Stream Control Transmission Protocol) qui, comparé aux protocoles de transport usuels, présente une meilleure fiabilité grâce à ses principales caractéristiques: le Multihoming et le Multistreaming.

Dans le cadre de ce mémoire de magister, nous nous somme proposé d'étudier les Performances de l'extension Mobile SCTP dans l'Equilibrage de Charge basée sur le Type de Flux dans un Environnement Mobile Hétérogène (réseau NEMO : Network Mobility). Il apparaît nécessaire de prendre donc des décisions qui s'appuient sur des informations sur la QoS et qui peuvent apporter réponse à la question suivante: quelle technologie d'accès attribuer pour quelle application? Notre travail envisage précisément de répondre à cette problématique grâce à une étude sur la QoS pour différents modes d'accès avec différentes applications.

Chapitre 1

Analyse de la Mobilité dans les Réseaux IP

I Analyse de la Mobilité dans les Réseaux IP

I.1 Mobilité - Vue d'ensemble

On définit la mobilité comme le caractère, la capacité ou la facilité d'un objet ou d'une personne à être déplacé ou de se déplacer par rapport à un lieu, position ou ensemble d'objets de même nature. L'action de changer de position et le résultat de cette action sont appelés mouvement ou déplacement. Dans le domaine des réseaux, la mobilité se traduit par la possibilité qu'ont certaines entités à se déplacer entre des points d'attachement différents. Nous énumérons quelques exemples, illustrés dans la figure (Fig 1.1) :

• (1) Un terminal est physiquement déplacé à un autre endroit et reconnecté à l'Internet par le biais d'un nouveau réseau ;

• (2) Un utilisateur décide d'utiliser un nouveau terminal ;

• (3) Un terminal connecté simultanément à plusieurs réseaux change l'interface active ;

• (4) Parallèlement au déplacement de l'utilisateur, des données personnelles et applications portables sont migrées sur un autre terminal.

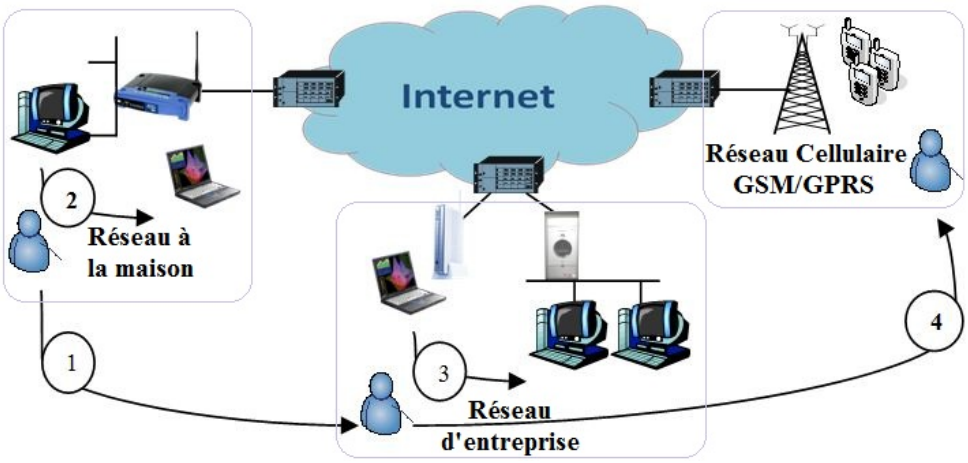

Fig. 1.1- Types de mobilité

Dans ces situations, l'identité de l'entité mobile ne change pas, mais son support d'attachement au réseau change. Les correspondants qui sont en train de communiquer avec l'entité mobile le font en envoyant des paquets adressés à un point d'attachement bien précis.

I.1.1 Types de mobilité

On distingue plusieurs types de mobilité en fonction des entités qui sont impliquées. Généralement, on définit une connexion réseau comme une liaison établie par deux entités qui se trouvent aux deux bouts de la connexion et qui s'envoient des données. En fonction du niveau d'abstraction, ces entités peuvent désigner les machines, les applications ou même les utilisateurs. Dans ce qui suit, nous présentons les caractéristiques de ces différents types de mobilité.

I.1.1.1 Mobilité des terminaux

La mobilité de terminaux, qui est aussi celui rencontré le plus fréquemment. En fonction de la portée et de la durée du déplacement, on distingue deux sous-catégories de la mobilité des terminaux : la portabilité et la mobilité continue (Fig 1.2).

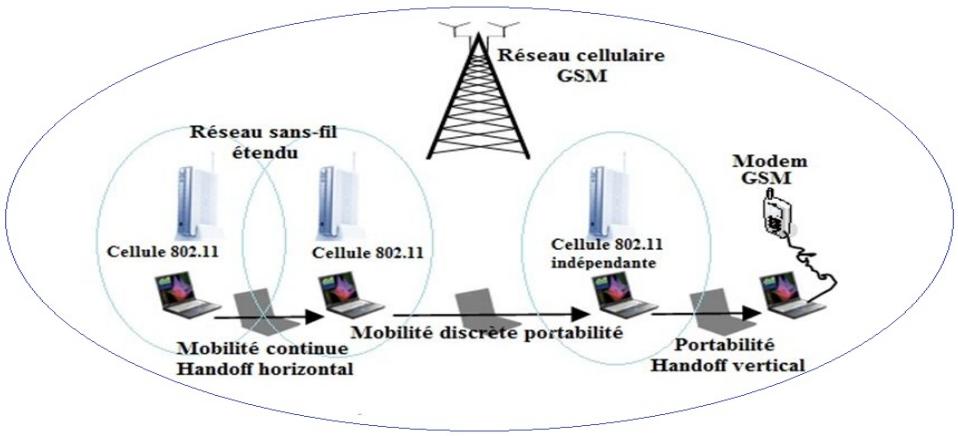

Fig. 1.2- Mobilité de terminaux

Dans le cas de la portabilité : un terminal est déplacé entre deux points d'attachement aux réseaux distants. En profitant de la présence accrue des terminaux portables, on a pris l'habitude de les emporter et de les connecter au réseau dans différents endroits, profitant des îlots de couverture sans-fil dans les campus universitaires, les hôtels ou les gares.

Cependant, ceci n'est pas un modèle de mobilité propre aux réseaux sans-fil, puisque c'est identique à la connexion de l'ordinateur portable par un réseau filaire. En même temps, être connecté par un sans fil ne signifie pas nécessairement être mobile, car des machines fixes peuvent être connectées par un sans-fil pour éviter le câblage.

La portabilité relève plutôt du nomadisme et sa principale caractéristique est que le terminal n'utilise pas le réseau pendant qu'il est déplacé.

Dans le second cas, celui de la mobilité continue, les utilisateurs se déplacent au long d'une couverture réseau sans-fil contiguë, fournie par un ou plusieurs points d'accès. Ces points d'accès peuvent être équipés d'une même technologie et former des cellules adjacentes. L'autre cas est quand plusieurs technologies sans-fil créent des zones de couverture superposées.

Quand un terminal est connecté à un réseau sans-fil, il peut être déplacé sans problème dans le rayon de son point d'accès courant, les seules conséquences notables apparaissant au niveau des erreurs de transmission si le terminal s'éloigne. Par contre, une opération plus complexe est le transfert d'un terminal d'un réseau à un autre, désigné par le terme anglais handoff. Le handoff entre deux cellules d'un même type de réseau est appelé handoff horizontal, tandis que le handoff vertical se fait entre des points d'accès d'une technologie sans-fil différente.

La continuité de la connexion au réseau malgré le changement de point d'accès est la caractéristique principale d'un handoff. Elle est exprimée par le niveau de pertes des paquets ou encore par le temps nécessaire au transfert de l'association. Ces pertes apparaissent à cause du délai pris par la réassociation physique au nouveau point d'accès, mais aussi parce que la nouvelle localisation doit être apprise par les équipements réseaux qui acheminent les paquets vers l'hôte mobile. Souvent dans le cas d'un handoff vertical mais parfois aussi dans les handoffs horizontaux, un changement d'adresse IP est nécessaire, ce qui induit d'autres délais supplémentaires et provoque même l'interruption des connexions réseaux. Dans le cas idéal où le handoff ne provoque pas d'interruption des services réseau, on parle d'un handoff transparent aux applications.

I.1.1.2 Mobilité des réseaux

Un cas particulier de la mobilité de terminaux est quand un sous-réseau entier se déplace, ses hôtes pouvant garder leur topologie inchangée à l'intérieur de ce réseau. Imaginons le cas de passagers d'un train ou d'un avion où les équipements embarqués dans une voiture.

Malgré l'immobilité des hôtes par rapport au lien local, le réseau lui-même change ses liens avec les réseaux voisins et donc change sa position et ses interconnexions dans l'Internet. Au cas où les adresses IP du réseau restent inchangées, ce déplacement pourrait rester invisible pour ses hôtes. Cependant, dans la plupart des cas, le réseau doit opérer une reconfiguration des adresses, ce qui a un impact sur les terminaux. Dans ce cas, une place privilégiée pour implanter des fonctionnalités liées à la mobilité est dans le routeur de bord mobile du réseau mobile.

I.1.1.3 Mobilité des utilisateurs

La plupart des utilisateurs se servent de plus d'un terminal pour communiquer à travers des applications et services réseaux. Il n'existe pas encore un seul terminal qui offre à la fois faible poids, petite taille, grande autonomie d'énergie, forte puissance de calcul et capacités multimédia étendues. Les utilisateurs font un compromis et utilisent différents terminaux en fonction de l'endroit et de la situation où ils se trouvent. Par conséquent à la pluralité des terminaux et services de communication, l'utilisateur possède plusieurs identifiants, en fonction de l'application utilisée : adresses e-mail professionnelles et personnelles, numéros de téléphone (téléphone portable, téléphones à la maison et au bureau), et d'autres noms d'utilisateur pour d'autres applications Internet comme la messagerie instantanée et la téléphonie sur Internet.

Une fonction importante à accomplir dans le cadre de la mobilité de personnes est qu'un utilisateur peut communiquer indépendamment de son terminal. Cela demande que tous les identificateurs énumérés plus haut soient regroupés pour que la personne puisse être localisée et les communications redirigées vers son terminal et application actifs. Les connexions réseaux doivent donc être établies à un niveau d'abstraction plus élevé, entre des personnes, et non plus entre des terminaux ou d'applications.

I.1.1.4 Mobilité de services et applications

La mobilité des utilisateurs décrite précédemment implique que les services et applications réseau sont disponibles et peuvent être utilisés d'une manière similaire, indépendamment du terminal courant de l'utilisateur. Pour satisfaire ceci, des applications entières ou des parties de code logiciel doivent être transférés dans certains cas d'une machine à une autre, même en cours d'exécution. On appelle ce transfert mobilité de services et applications.

Un exemple particulier de la mobilité de composants logiciels est le profil personnel de services utilisateur. La portabilité du profil de services signifie que les applications réseau fournissent les mêmes services, associés aux préférences de l'utilisateur, quelque soit son terminal actif. Le but est de créer un environnement personnel virtuel que l'utilisateur emporte partout pour accéder à ses données personnelles, à ses communications réseau et à ses applications favorites. Bien sur, l'environnement personnel doit aussi intégrer d'une manière transparente des services locaux – on veut avoir accès à des données personnelles, mais aussi accéder aux vidéoprojecteurs et imprimantes de l'endroit visité.

I.1.1.5 Mobilité de sessions

Une session réseau est une abstraction qui regroupe une ou plusieurs connexions réseau et qui fournit des services pour gérer globalement l'état de ces connexions. La mobilité au niveau d'une session doit permettre à ses connexions de rester actives et suivre les deux points finaux, en dépit de

leur mobilité. La mobilité au niveau session est plus générale que la mobilité des hôtes ou la mobilité des utilisateurs, puisque ces deux classes y en sont des instances particulières. Ainsi, comme on peut voir sur la figure (Fig 1.3), le déplacement d'un hôte peut être vu comme le déplacement d'une session regroupant toutes les connexions qui ont cette machine comme point final. De l'autre coté, la mobilité des personnes implique le déplacement des connexions ouvertes par une personne vers son nouveau terminal.

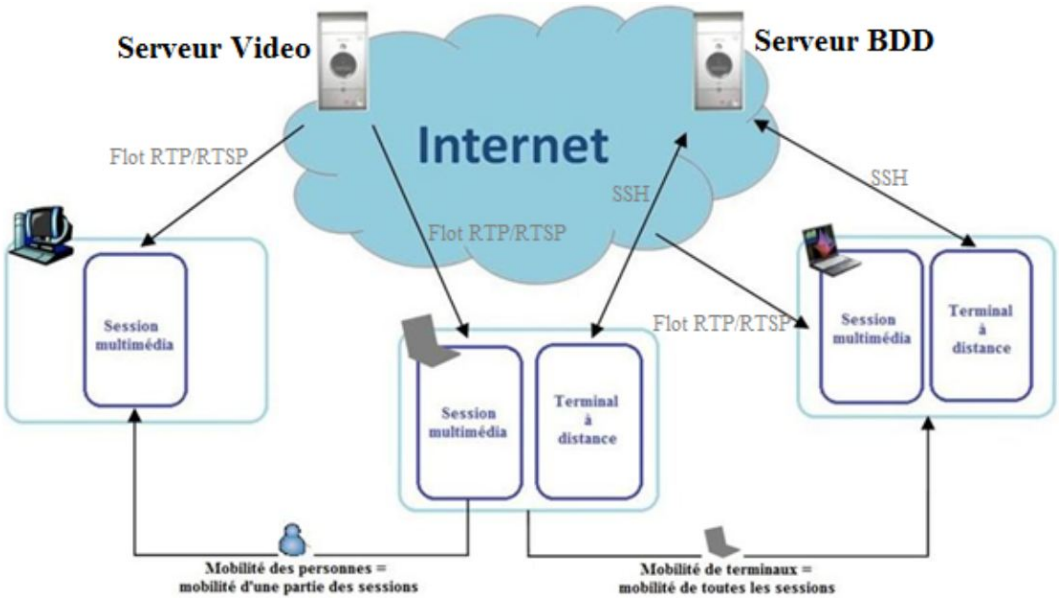

Fig. 1.3- Mobilité de sessions

Dans la pile de protocoles Internet il n'y a pas du support explicite pour regrouper plusieurs connexions dans une session, alors que ce niveau existait dans le modèle OSI. En plus de l'initialisation d'une session et de la négociation des divers paramètres pour l'identifier, le niveau session du modèle OSI permet la définition de points de synchronisation de la session. Ceci fournit le support pour l'interruption et le redémarrage de la session à partir de ces points. Vu l'absence du niveau session dans les protocoles Internet, certaines applications ont été conçues dès le départ pour s'établir elles-mêmes une association de longue durée qui peut comprendre plusieurs connexions simultanées ou enchaînées dans le temps. On cite comme exemple des sessions HTTP pour la navigation web.

I.1.2 Fonctionnalités requises par la mobilité

Nous présentons dans cette section les démarches nécessaires pour assurer le bon fonctionnement de la mobilité des machines dans les réseaux Internet. Nous avons illustré ces opérations dans la figure (Fig 1.4). Ainsi, l'association à un nouveau point d'attachement à l'Internet peut demander à l'hôte mobile de fournir des éléments d'authentification, ainsi que la reconfiguration de certains paramètres

liés au nouveau réseau. Ensuite, un nœud mobile qui remplit des fonctions de serveur devrait pouvoir être retrouvée par ses clients. De plus, on veut assurer le transfert de toutes les connexions actives de l'hôte mobile vers sa nouvelle localisation sur le réseau.

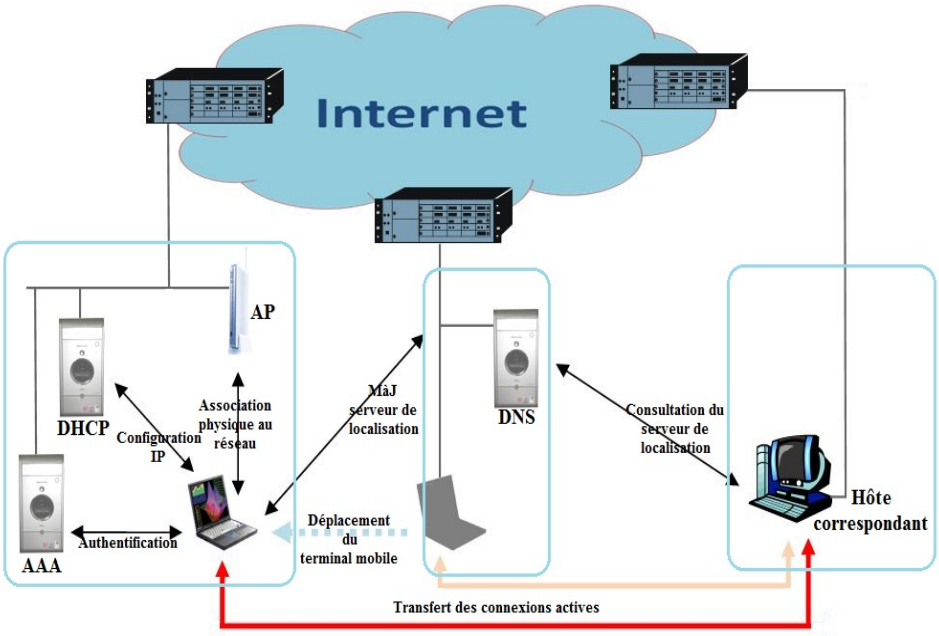

Fig. 1.4- Opérations au cours de la mobilité

Les démarches demandées par la mobilité d'un utilisateur sont en grandes lignes similaires à celles de la mobilité d'hôtes : authentification, mise à jour des informations qui servent à sa localisation, et le transfert du contexte utilisateur entre son ancien et son nouveau terminal. L'analyse de la suite d'opérations que nous allons présenter dans la suite est faite principalement du point de vue de la mobilité des hôtes, mais le même type de démarches est valable par exemple pour le transfert d'une session réseau suite au changement de terminal d'un utilisateur.

I.1.2.1 Association au réseau

La première étape est l'association physique du terminal au nouveau point d'attachement à l'Internet. Cette étape dépend fortement de la technologie d'accès à l'Internet : réseau local filaire Ethernet ou sans-fil WiFi, accès distant par modem classique ou en utilisant un téléphone 3G. L'association au nouveau réseau peut demander une authentification de la part du terminal mobile. L'association physique au réseau sans-fil peut être conditionné par l'authentification du terminal, qui doit fournir une clé sécrète ou un mot de passe. Après que l'association physique soit accomplie, d'autres interactions supplémentaire avec des serveurs AAA (Authentication, Authorization, and Accounting) [01] peut encore avoir lieu pour que l'hôte mobile ait de droit d'utiliser en partie ou en totalité les services réseau ou pour que son activité réseau puisse être comptabilisée.

I.1.2.2 Configuration au niveau IP

Si l'hôte mobile se déplace entre deux sous-réseaux différents, une démarche importante qu'un hôte mobile doit accomplir est la reconfiguration de son interface réseau au niveau L3. Pour compléter sa connexion au nouveau sous-réseau, l'hôte doit être reconfiguré avec une nouvelle adresse IP, masque du sous-réseau, passerelle, serveur de noms, etc. Les travaux de recherche dans ce domaine ont abouti par l'apparition de protocoles de configuration automatique de ces paramètres, sans aucune intervention nécessaire de la part de l'utilisateur. Les protocoles qui sont actuellement utilisés sont le DHCP (*Dynamic Host Configuration Protocol*) [02] pour l'IPv4 et son correspondant DHCPv6 [03] (pour la version 6 du protocole IP).

I.1.2.3 Mise à jour des informations de localisation

Correctement configuré avec une adresse IP propre au nouveau sous-réseau, l'hôte est maintenant en mesure d'initier des connexions avec d'autres machines, sur le lien local ou à travers l'Internet. Par contre, si l'hôte mobile remplit la fonction d'un serveur, il doit accomplir une démarche supplémentaire qui consiste à rendre sa localisation visible par les clients potentiels.

Pour qu'un hôte mobile puisse être localisé par ses correspondants, il doit être identifié par une autre chose que son adresse IP, si celle-ci change. Chaque fois qu'il se déplace et acquit une nouvelle adresse IP, il doit mettre à jour la correspondance entre son identifiant et son adresse courante dans un répertoire. Après que ce répertoire ait été mis à jour, les hôtes correspondants pourraient le consulter, apprendre la nouvelle adresse et initier des connexions vers l'hôte mobile. Le protocole Internet qui pourrait servir à ce but est le DNS [04], mais il présente néanmoins quelques inconvénients vis-à-vis de la mise à jour des correspondances nom de domaine < -- > adresse IP.

I.1.2.4 Transfert des sessions réseau

Si un nœud mobile se déplace pendant qu'il a des connexions réseau actives, un objectif additionnel apparaît : transférer les sessions réseaux vers la nouvelle adresse de l'hôte mobile. Notifier les correspondants de l'hôte mobile pour que ceux-ci envoient leurs paquets vers la nouvelle adresse de l'hôte mobile ne résout pas en totalité le problème. La cause vient des protocoles du niveau transport et de certaines applications, qui ne permettent pas le changement à la volée d'adresses IP cours d'une connexion. Ceci est en effet un des problèmes les plus difficiles à résoudre pour la mobilité d'hôtes, nous expliquerons ce point plus en détail dans le chapitre suivant.

I.1.2.5 Multi-domiciliation

Un autre cas de pseudo-mobilité est causé par la multi-domiciliation (multihoming), un terme qui désigne une machine qui possède plusieurs interfaces réseau et qui les utilise simultanément ou alternativement. Théoriquement, le multi-accès simultané peut être utile pour augmenter la robustesse et le débit des connexions. En utilisant plusieurs interfaces et chemins concurrents pour acheminer des paquets nous appelons cela le partage des charges, ce point sera discuté dans le deuxième chapitre.

I.2 Mobilité dans le Modèle TCP/IP

Les protocoles utilisés dans l'Internet sont communément désignée sous le nom TCP/IP. Ce nom vient de ses deux principaux composants, le Transmission Control Protocol (TCP) au niveau transport et Internet Protocol (IP) au niveau réseau.

Les protocoles Internet sont structurés en plusieurs couches. Le niveau liaison s'occupe de l'accès au lien local et de la communication effective entre les machines qui y sont connectées directement. Le niveau réseau est lui-même formé d'un ensemble de protocoles - IP, ICMP [05], ARP [06], etc. - qui collaborent à unifier les différents liens physiques et fournir le service réseau global offert par IP. Au niveau transport, les protocoles TCP et UDP sont utilisés. Enfin, le niveau application utilise les services de transfert de données de bout en bout fournies par les protocoles de transport TCP et UDP. La figure (Fig 1.5) montre le parallélisme du modèle TCP/IP avec la pile de protocoles OSI.

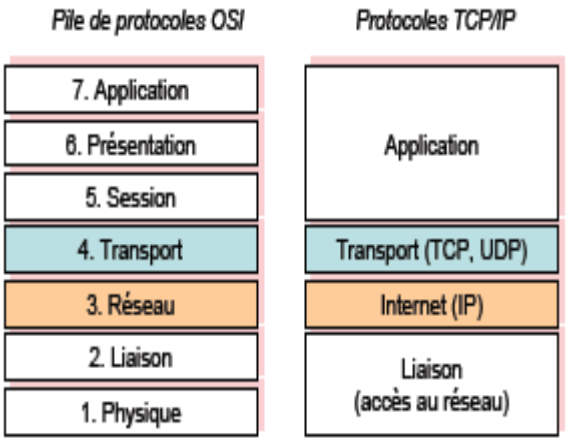

Fig. 1.5- Les piles de protocoles OSI et TCP/IP

I.2.1 Acheminement des paquets IP vers les nœuds mobiles

Le protocole IP est considéré comme étant le plus important d'Internet, car il unifie tous les réseaux physiques dans un seul réseau global. Il masque ainsi l'hétérogénéité des différentes technologies physiques et se charge d'acheminer les données entre n'importe quelle paire d'hôtes, indifféremment du sous-réseau ou celles-ci se trouvent. Chaque point d'attachement à l'Internet a une adresse IP unique, par laquelle l'hôte qui y est connecté est joignable par les autres machines. Chaque paquet IP envoyé via l'Internet contient l'adresse IP destination du paquet, qui sera utilisé dans le processus d'acheminement à travers l'Internet. L'acheminement est fait par les routeurs qui possèdent plusieurs interfaces réseau, chacune reliée à un sous-réseau différent. Les routeurs maintiennent une table de routage qui contient des correspondances entre des préfixes des adresses IP et le nœud suivant auquel le routeur doit délivrer le message. La séquence de consultation de tables de routage et de retransmission du paquet est répétée sur plusieurs routeurs, jusqu'au moment où le paquet arrive à sa destination.

Dans l'IPv4, les adresses IP sont codées sur 32 bits; la nouvelle version IPv6 utilise des adresses IP d'une longueur de 128 bits. On a donc un nombre immense d'adresses IP possibles. En conséquence, les tables de routage de routeurs ne peuvent pas contenir la totalité des adresses des hôtes connectés à l'Internet. Pour que le processus de routage soit fiable à l'échelle actuelle et future de l'Internet, les adresses IP ont été structurées dès le début d'une façon hiérarchique : une partie pour l'adresse du sous-réseau et le reste pour identifier les hôtes du sous-réseau respectif.

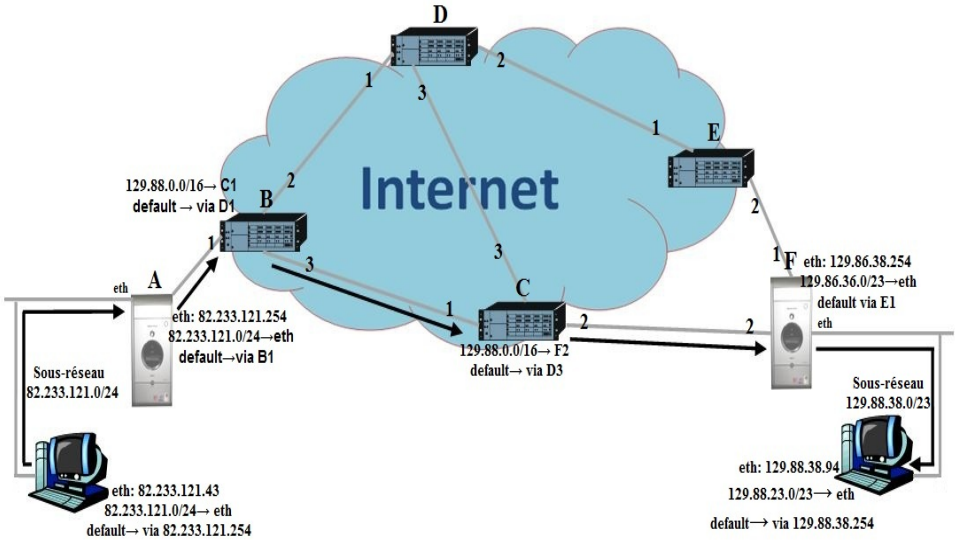

Fig. 1.6- Adressage IP et l'acheminement de paquets IP

Comme on peut voir sur la figure (Fig 1.6), seulement la partie sous-réseau des adresses IP est présente dans les tables de routage. Si la structuration des adresses permet la simplification des tables

de routage, elle introduit néanmoins une restriction très importante du point de vue de la mobilité d'hôtes. En effet, une adresse IP ne peut être utilisée que dans le sous-réseau dont elle fait partie, car les paquets qu'elle reçoit sont acheminés en utilisant l'adresse du sous-réseau. Chaque fois qu'un hôte se déplace, elle doit utiliser une nouvelle adresse conforme au nouveau point d'attachement, c'est-à-dire qui contient le numéro du nouveau sous-réseau.

I.2.2 Mobilité des connexions au niveau transport

Quand les protocoles TCP/IP ont été conçus, on a implicitement considéré que les machines sont statiques et ne changeront pas le point d'attachement et l'adresse IP pendant la durée de vie d'une connexion. En conséquence, au lieu d'identifier les hôtes par d'autres moyens, les protocoles du niveau transport et les applications ont été développés en utilisant les adresses IP comme des identificateurs stables des extrémités d'une connexion.

Dans la composition du trafic global Internet, TCP est majoritaire, il représente 80% des flots de données échangés. Puisqu'il offre des garanties pour le transport des données à destination, TCP est utilisé par toutes les applications qui ont besoin de la fiabilité dans l'échange de données sur l'Internet. Néanmoins, il présente un problème majeur vis-à-vis de la mobilité, car une connexion TCP est identifiée par un quadruple (adresse IP source, port source, adresse IP destination, port destination). Une fois qu'une connexion TCP est établie, chacune des deux extrémités va envoyer et recevoir des données seulement vers et à partir d'une machine située à une adresse IP fixe. Cela entre en conflit avec la mobilité d'hôtes : si une machine se déplace vers un sous-réseau différent, elle ne peut pas garder l'ancienne adresse IP, car les paquets qui y sont adressés seront acheminés à l'ancien sous-réseau. À la place, la machine doit changer son adresse pour pouvoir réceptionner les données au nouvel endroit où elle se trouve. Mais dans ce cas elle ne pourra pas continuer les connexions TCP déjà ouvertes, car ces connexions n'acceptent pas de données envoyées ou reçues d'une ou à une adresse IP différente ; en conséquence, ces connexions TCP seront interrompues.

Le même problème apparaît également dans le cas des hôtes ayant un multi-accès au réseau. Les différentes interfaces sont en général connectées à des sous-réseaux différents et donc configurées avec des adresses IP différentes. Dans cette situation également, il est impossible, du point de vue du protocole TCP, de changer d'une manière transparente les interfaces utilisées pour envoyer ou recevoir les datagrammes.

L'autre protocole de transport, UDP, est un protocole simple, sans connexion et non fiable, qui ne garantit pas la livraison des paquets à destination ou leur arrivée dans l'ordre qu'ils ont été transmis. Par rapport au TCP, dans UDP chaque paquet est envoyé indépendamment des autres, et le même socket UDP peut être utilisée pour envoyer et recevoir des paquets vers et de n'importe quelle adresse IP. Ceci permet des changements dans les adresses IP des machines source et destination et il semble faciliter la mobilité des hôtes. En pratique les choses ne sont pas si simples : des applications construites autour de l'UDP utilisent un seul socket pour plusieurs connexions avec des

correspondants différents. Ces applications utilisent alors l'adresse IP source des paquets IP reçues pour démultiplexer les données de ces connexions virtuelles. Ce type d'applications doit être informé en avance sur un changement d'adresse IP distante pour continuer à garder un état correct sur leurs échanges de données. Néanmoins, cette notification devrait suffire pour pouvoir envoyer et recevoir des paquets de la nouvelle adresse, sans la surcharge introduite par le redémarrage d'une nouvelle connexion dans le cas du TCP.

I.3 Conclusion

Ce chapitre a été consacré à la définition de la mobilité et sa gestion dans les réseaux en général et le modèle TCP/IP en particulier. Nous avons présenté les différents types de mobilité, entre-autres la mobilité des terminaux et la mobilité des utilisateurs. À un niveau d'abstraction plus élevé nous avons pu discuter de la mobilité d'une session de connexions réseaux, concept qui permet de regrouper la mobilité des terminaux ou des utilisateurs. Ensuite nous avons vu que les problèmes vis-à-vis de la mobilité des hôtes dans l'Internet sont issus principalement du conflit entre les deux rôles d'une adresse IP : adresse d'hôte utilisé comme indicateur de routage, ainsi qu'identificateur d'hôte dans une connexion TCP.

Dans le chapitre suivant nous allons voir les différentes solutions proposées jusqu'ici par les groupes de recherches pour résoudre les différentes questions posées.

Chapitre 2

Solutions de Gestion de la mobilité des Nœuds dans les Réseaux IP

II Solutions de Gestion de la Mobilité des Nœuds dans les Réseaux IP

II.1 Introduction

Une multitude des solutions ont été proposées pour faire face aux défis posés par la mobilité des hôtes dans les réseaux IP. La solution officielle proposée à l'IETF n'a pas réussi d'obtenir le consensus nécessaire pour lui permettre un déploiement à grande échelle. D'autres propositions intervenant à différents endroits de l'infrastructure Internet ont été étudiées, chacune avec ses avantages et ses faiblesses. En conclusion, au moment actuel, aucune solution ne s'est imposée pour répondre d'une manière satisfaisante à tous les problèmes de la mobilité évoqués dans le chapitre précédent.

Les différentes propositions ont été classifiées en fonction de la couche protocolaire où elles opèrent et apportent des extensions. Une analyse de chaque solution sera présentée dans la suite de ce chapitre.

II.2 Solution au niveau réseau : Mobile IP

L'adresse IP est composée de deux parties : le préfixe qui détermine le réseau sur lequel le nœud se trouve, et l'identifiant de ce nœud sur son réseau. Internet est un réseau à grande échelle, c'est la raison pour laquelle que chaque routeur ne peut mémoriser qu'une route vers tous les nœuds qui y sont attachés. Les routeurs ne stockent que des entrées correspondant à des réseaux en considérant que des paquets destinés à des nœuds ayant le même préfixe seront tous routés d'une manière identique.

Dans ce contexte, la mobilité du nœud introduit un nouveau problème de routage : le Nœud Mobile (Mobile Node – MN) se déplace d'un réseau vers un autre réseau. S'il ne change pas son adresse IP, il aura un différent préfixe sur ce nouveau réseau. Cependant, le nœud doit être situé sur un réseau avec le même préfixe indiqué par son adresse IP afin de pouvoir recevoir les paquets qui lui sont destinés.

Pour qu'un MN puisse changer de réseau et garder la connexion à Internet, il doit changer d'adresse IP à chaque fois qu'il change du réseau.

Mais une fois que le MN change son adresse IP, il ne peut plus conserver les communications en cours au niveau de la couche transport ou des couches supérieures. Comme nous l'avons dans le premier chapitre.

Pour qu'un MN puisse maintenir les communications en cours et garder la connexion à Internet tout en se déplaçant d'un réseau vers l'autre, le protocole IP Mobile [07] propose de gérer la mobilité du MN au niveau IP. Il permet au MN d'utiliser deux adresses IP et un mécanisme de Mise à jour d'association (Binding Update – BU) pour masquer le changement d'adresses IP aux applications exécutées entre le MN et ses correspondants. Par conséquent, les communications en cours peuvent être maintenues lorsque le MN change du réseau.

Avant d'expliquer le fonctionnement du protocole IP Mobile, nous présentons d'abord son architecture et ses composants (Fig 2.1).

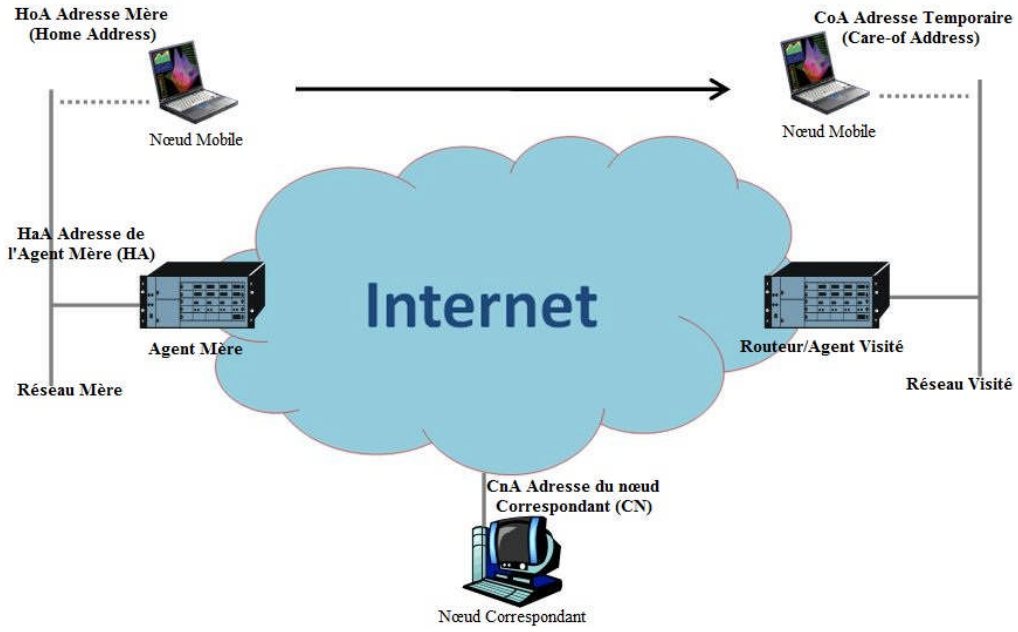

Fig. 2.1- Architecture du protocole IP Mobile

- Nœud Mobile (Mobile Node – MN) : un terminal qui peut changer de point d'attachement d'un réseau à un autre.
- Routeur Mobile (Mobile Router – RM) : Un routeur qui peut changer de point d'attachement d'un réseau à un autre et il peut servir de passerelle pour sont réseau mobile.
- Adresse Mère (Home Address – HoA) : est l'adresse IP du MN sur son réseau mère. Elle permet d'identifier le MN de façon unique sur tous les réseaux.
- Adresse Temporaire (Care-of Address – CoA): permet de localiser le MN sur un réseau visité

afin de lui permettre d'envoyer et recevoir des paquets sur ce réseau.

- Agent Mère (Home Agent – HA) : correspond à un routeur d'accès particulier situé dans le réseau mère. Il est chargé d'assurer l'association entre l'adresse mère et l'adresse temporaire du MN lorsque celui-ci est attaché à un réseau visité. Cet agent est également chargé de rediriger les paquets IP à destination de l'adresse mère du MN vers son adresse temporaire sur son réseau visité.
- Nœud Correspondant (Correspondant Node – CN) : est un terminal qui communique avec le MN.
- Réseau Mère : est un réseau auquel le MN et son HA s'attachent.
- Réseau Visité : est un réseau autre que le réseau mère pour un MN. Le MN aura une adresse temporaire quand il s'attache à ce réseau.
- Réseau mobile (Network Mobile-NM) : est un réseau dont le routeur de bord change dynamiquement sans point d'attachement. Ils sont connectés à l'Internet par le biais d'un ou plusieurs routeurs mobiles (MRs).
- Agent Visité (Foreign Agent – FA) : correspond à un routeur d'accès du réseau visité auquel le MN est attaché. Il fournit des services de routage au MN lorsque le MN est enregistrée auprès de ce dernier.

II.2.1 Protocole IPv4 Mobile

Quand un MN se trouve sur son réseau mère, il communique de la même manière que n'importe quel nœud sur Internet en utilisant son adresse mère comme son adresse source. Les paquets qui lui sont destinés comprennent son adresse mère comme son adresse de destination et sont routés en fonction du préfixe du réseau mère. Le HA est inactif pour le MN.

Lorsque le MN est attachée à un réseau visité, il devrait d'abord obtenir une adresse temporaire. Le protocole IPv4 Mobile permet au MN d'utiliser deux différents types d'adresses temporaires:

- Le MN utilise l'adresse IP publique de l'agent visité comme son adresse temporaire. L'agent visité attribue une adresse IP privée au MN pour le localiser sur son réseau. L'agent visité fonctionne comme un relais entre le HA et le MN. C'est lui qui reçoit les paquets envoyés par le HA et les redirige au MN. Ce mécanisme peut être comparé au système de translation d'adresses (Network Address Translation – NAT). L'utilisation de ce type d'adresse temporaire est préférable dans le protocole IPv4 Mobile car elle permet aux nombreux MNs de partager une même adresse temporaire publique et évite d'allouer les nouvelles adresses IP publiques aux MNs à cause de la pénurie des adresses IPv4 publiques.
- Le MN auto-configure son adresse temporaire publique avec le serveur DHCP. Le MN utilise cette adresse publique sur le réseau visité, donc le HA n'envoie plus les paquets vers son agent visité, mais les envoie directement vers son adresse temporaire publique. Dans ce cas, l'agent visité fonctionne

comme un routeur d'accès (Access Router – AR). L'utilisation de ce type d'adresse temporaire permet au MN de fonctionner sans l'agent visité, mais elle demande toutefois la réservation d'un pool d'adresses IP publiques pour les MNs et elle pose un problème au niveau de l'espace d'adresses d'IPv4.

Une fois que le MN a eu l'adresse temporaire, il doit mettre à jour l'association entre son adresse mère et son adresse temporaire avec le HA. En fait, le HA maintient une table d'associations contenant l'association entre l'adresse mère et l'adresse temporaire du MN qu'il gère. Cette table d'associations doit être réactualisée à chaque fois que le MN change de réseaux.

Si le MN utilise l'adresse de l'agent visité, il envoie d'abord le message – Requête d'enregistrement (Registration Request) à son agent visité. Ensuite, l'agent visité vérifie la validité du message "Requête d'enregistrement", met à jour sa table d'associations et retransmet ce message au HA. Le HA reçoit ce message, réactualise sa table d'associations et envoie un message – Réponse d'enregistrement (Registration Reply) au MN via l'agent visité. Si le MN utilise une adresse temporaire publique, il échange directement les messages avec le HA sans impliquer l'agent visité.

Quelle que soit l'adresse temporaire utilisée par le MN, dès que le HA reçoit le message "Requête d'enregistrement", il diffuse une requête ARP (Address Resolution Protocol) sur son réseau. Ceci permet au HA d'associer son adresse MAC avec l'adresse mère du MN afin de pouvoir intercepter tous les paquets destinés à l'adresse mère du MN.

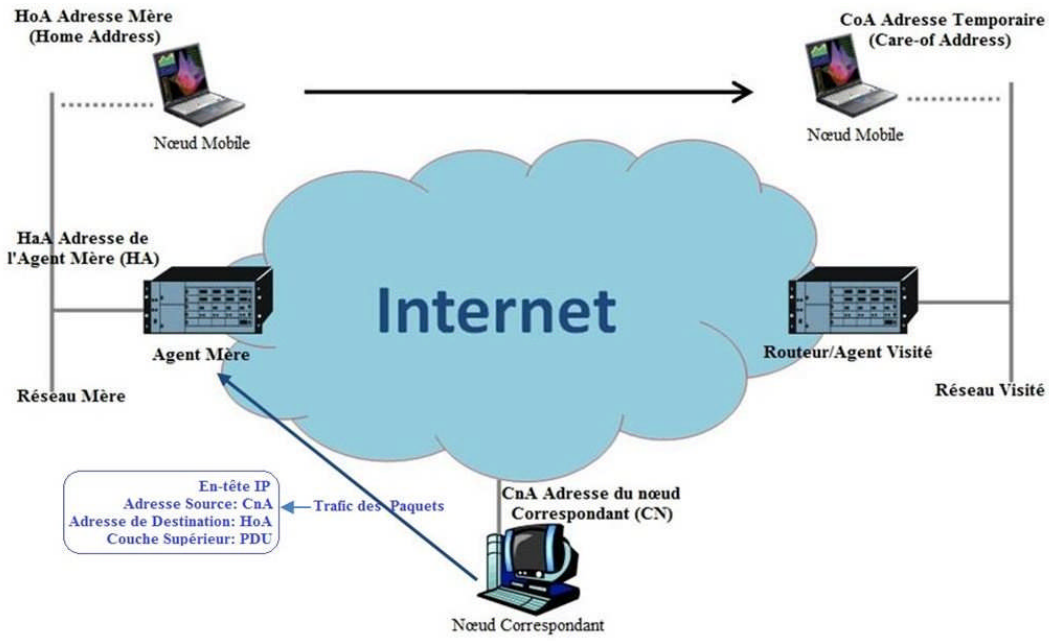

Fig. 2.2- Paquets envoyés par le CN à l'adresse mère du MN

Les figures (Fig 2.2), (Fig 2.3), (Fig 2.4) et (Fig 2.5) permettent d'expliquer le mécanisme de routage de paquets triangulaire géré par le protocole IPv4 Mobile lorsque le MN utilise une adresse temporaire publique dans le réseau visité.

La figure (Fig 2.2) présente la situation dans laquelle le MN est connecté à un réseau visité et a obtenu une adresse temporaire publique sur ce nouveau réseau visité et que le CN continue d'envoyer les paquets à l'adresse mère du MN.

La figure (Fig 2.3) présente la situation dans laquelle le HA encapsule les paquets interceptés et les redirige à l'adresse temporaire publique du MN via le tunnel IPSec [08]. Le tunnel IPSec est créé d'une manière optionnelle pour protéger les paquets transmis. Comme il est inconcevable que le HA modifie les paquets interceptés, les paquets d'origine sont justement encapsulés en ajoutant un nouvel en-tête IP devant l'en-tête existant. Le nouvel en-tête contient l'adresse du HA comme adresse source, et l'adresse temporaire du MN comme adresse destinataire. Le paquet encapsulé peut atteindre le réseau visité puisque l'adresse temporaire a un préfixe comme celui du réseau visité. Ce procédé permet de continuer à utiliser le mécanisme de routage IP conventionnel tout en permettant une redirection des paquets.

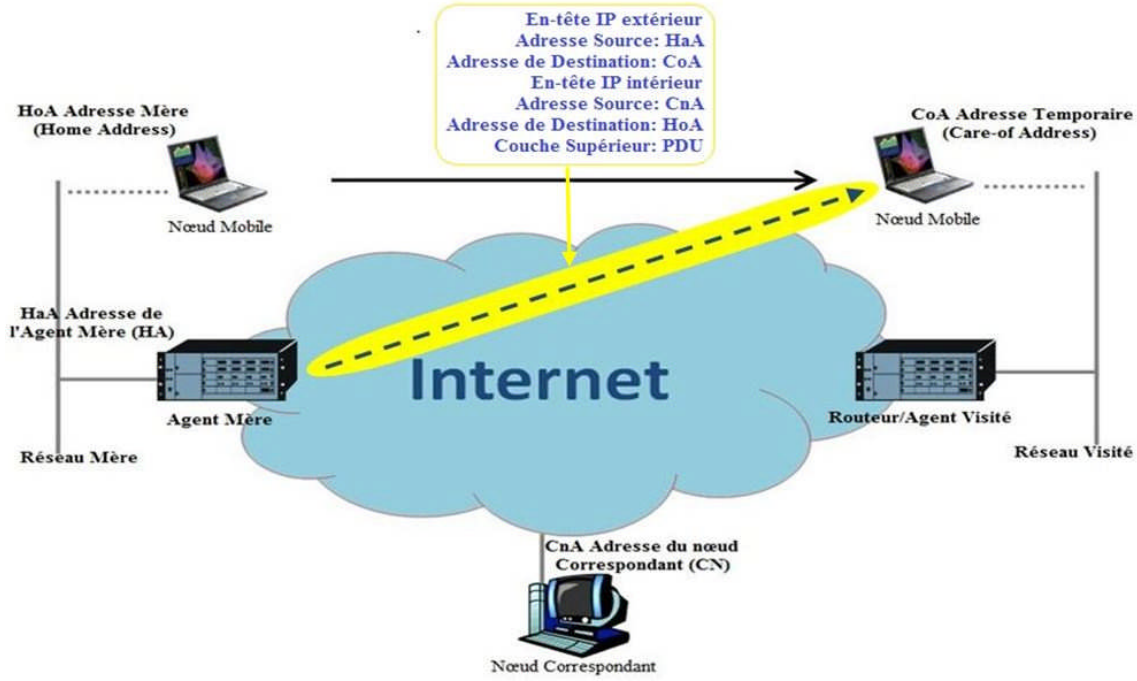

Fig. 2.3- Paquets interceptés, encapsulés et redirigés par le HA au MN via le tunnel IPSec

Les paquets issus du MN sur un réseau visité et à destination du CN utilisent un principe similaire. Les paquets IP d'origine comportent comme adresse source l'adresse mère du MN et comme adresse destination celle du CN. Ensuite les paquets IP d'origine sont encapsulés par le MN lui-même, l'adresse source de l'en-tête extérieure est l'adresse temporaire du MN et l'adresse destination de l'en-tête extérieure est l'adresse du HA. Les paquets transmis entre le MN et le HA sont aussi protégés par le protocole IPSec, ils traversent les routeurs intermédiaires jusqu'au HA. Ce processus est présenté dans la figure (Fig 2.4).

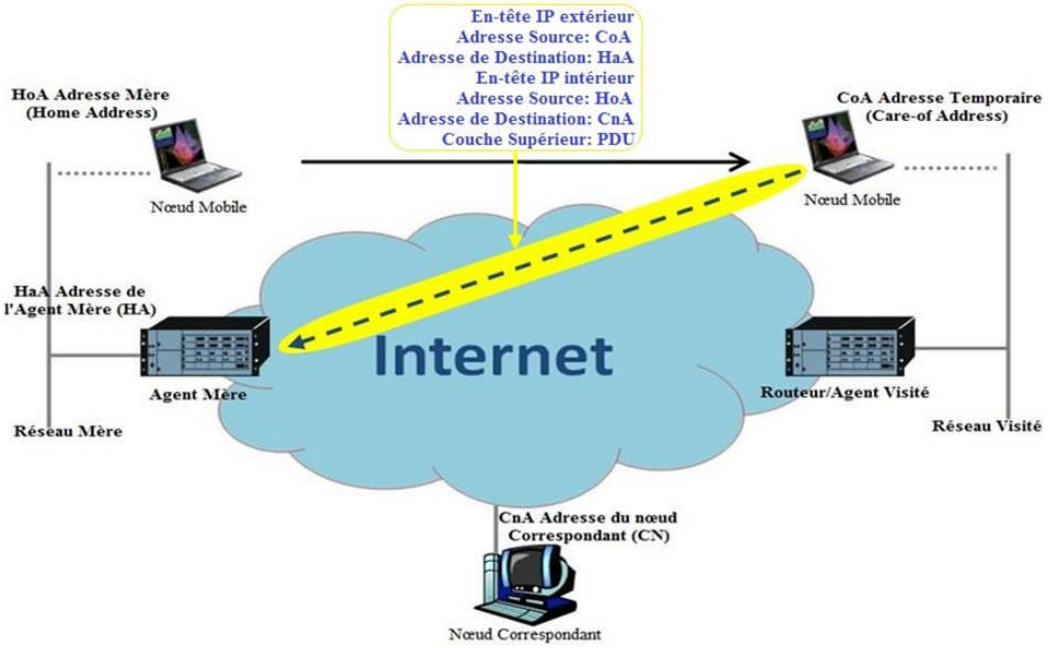

Fig. 2.4- Paquets encapsulés et envoyés par le MN au HA via tunnel IPSec

Une fois que le HA a reçu les paquets, il supprime l'en-tête extérieur des paquets et redirige les paquets au CN. Le CN reçoit les paquets du MN comme si le MN était dans le réseau mère (Fig 2.5).

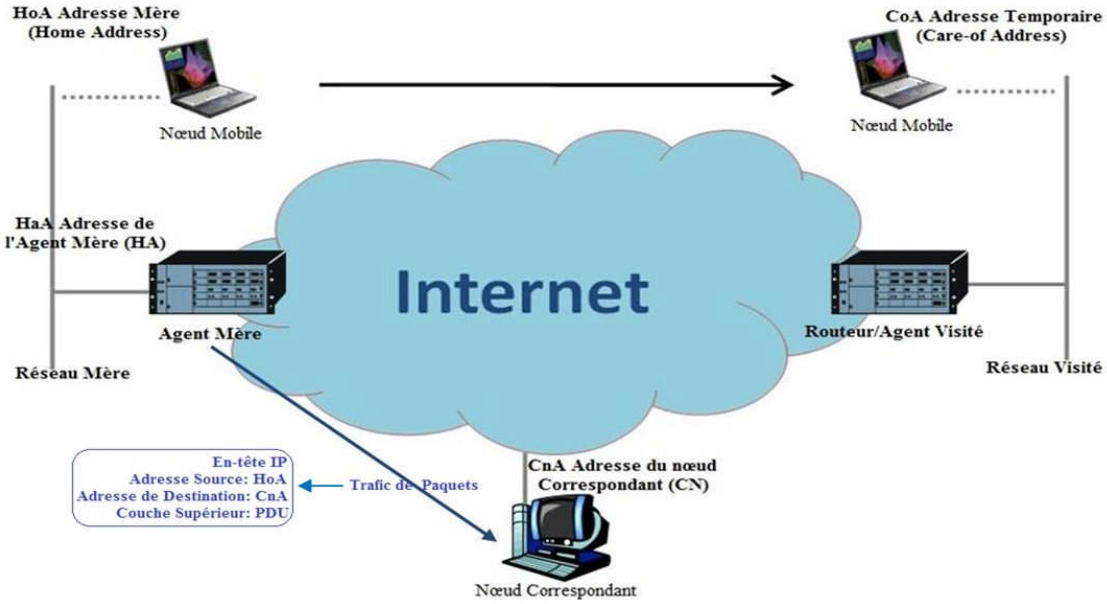

Fig. 2.5- Paquets dés-encapsulés et redirigés par le HA au CN

Les figures ci-dessus présentent la situation dans laquelle le MN a acquis une adresse temporaire publique par le mécanisme DHCP. Il se trouve lui-même à l'extrémité du tunnel et décapsule les paquets qui traversent le tunnel et arrivent à lui. Si le MN utilise l'adresse de l'agent visité comme son

adresse temporaire, c'est donc l'agent visité qui est à l'extrémité du tunnel. Lorsque l'agent visité reçoit les paquets tunnelés, il les décapsule et remet le paquet d'origine au MN.

En conclusion de notre description, grâce à l'utilisation du protocole IPv4 Mobile, le MN peut se déplacer sur le réseau sans interrompre la communication en cours, cependant, la gestion de la mobilité par le protocole IPv4 Mobile implique des problèmes de performances: la taille des paquets est augmentée à cause d'encapsulation des paquets par le HA ou par le MN, cela provoque la perte de bande passante en réseau; les paquets doivent suivre un chemin triangulaire pour être acheminé à destination, cela ajoute un délai d'acheminement supplémentaire significatif. Par conséquent, il est difficile de garantir la qualité de service requise pour les applications temps réel.

II.2.2 Protocole IPv6 Mobile

Le protocole IPv6 Mobile [09] est proposé pour résoudre les problèmes cités ci-dessus en utilisant les nouvelles fonctionnalités du protocole IPv6 :

- Le mécanisme de Détection de mouvement permet au MN de détecter le changement de réseau.
- Le mécanisme d'Auto-configuration d'adresses sans état permet au MN d'acquérir une adresse IPv6 globale aussitôt qu'il s'attache à un nouveau réseau visité.
- Le support du mécanisme de Mise à jour d'association pour CN et l'utilisation du nouvel en-tête IPv6 permet au MN et au CN de pouvoir communiquer directement sans passer par le HA.

Nous présentons d'abord les nouveaux messages de signalisation du protocole IPv6 Mobile, ensuite nous expliquons la gestion de la mobilité par ce protocole.

II.2.2.1 Messages de signalisation

Le protocole IPv6 Mobile définit un en-tête d'extension de mobilité pour les nouveaux messages de signalisation, tels que la Mise à jour d'association (Binding Update – BU), l'Acquittement de mise à jour d'association (Binding Acknowledgement – BA), l'Initialisation de test d'adresse mère (Home Address Test Init – HoTI), etc.

La valeur du champ d'en-tête suivant de l'en-tête IPv6 est défini comme 135, c'est-à-dire il y a un en-tête d'extension de mobilité dans le paquet IPv6.

Le format général d'en-tête d'extension de mobilité est donné dans la figure (Fig 2.6) [09]:

<table>
<tr><td>Next Header (8 bits)= 59</td><td>Header Length (8 bits)</td><td>MH Type (8 bits)</td></tr>
<tr><td colspan="2">Checksum (16 bits)</td><td></td></tr>
<tr><td colspan="3">Message Data</td></tr>
</table>

Fig. 2.6- Format d'en-tête d'extension de mobilité

- Le champ en-tête suivant (Next Header) : à la même fonction que celui de l'en-tête IPv6. Il identifie le prochain en-tête d'extension. Dans le cas du message de signalisation d'IPv6 Mobile, il doit valoir 59, c'est-à-dire qu'il n'y a pas d'en-tête d'extension suivant).
- Le champ longueur d'en-tête (Header Length) représente la longueur d'en-tête d'extension de mobilité. Il ne prend pas en compte les 8 premiers octets de l'en-tête.
- Le champ type d'en-tête de mobilité (MH Type) décrit les types des messages de mobilité.
  Le tableau II.1 présente les différents types des messages de mobilité.

Type d'en-tête	Type de message de mobilité
0	Demande de rafraîchissement de mise à jour d'association (en anglais Binding Refresh Advice)
1	Initialisation de test d'adresse mère (HoTI)
2	Initialisation de test d'adresse temporaire (en anglais Home Address Test Init – CoTI)
3	Test d'adresse mère (en anglais Home Test – HoT)
4	Test d'adresse temporaire (en anglais Care-of Test – CoT)
5	Mise à jour d'association (BU)
6	Acquittement de mise à jour d'association (BA)
7	Erreur de mise à jour d'association

Tableau II.1- Différents types des messages de mobilité

II.2.2.2 Gestion de la mobilité

La procédure du handover de niveau 3 gérée par le protocole IPv6 Mobile se décompose en trois phases :

→ La phase de Détection de mouvement,

→ La phase d'Auto-configuration d'adresses,

→ La phase de Mise à jour d'association.

La phase de Mise à jour d'association est décomposée également en trois phases : la phase de Mise à jour d'association avec le HA, la phase de routabilité de retour et la phase de Mise à jour d'association avec le CN.

a. Détection de mouvement

La phase de Détection de mouvement est la première phase de la procédure du handover de niveau L3, elle a pour objectif de détecter le changement de réseau afin que le MN puisse réagir à ce

changement et lancer les procédures correspondants, telles que Découverte de routeur, Auto-configuration d'adresses IPv6 sur le nouveau réseau, Mise à jour d'association avec HA et CN.

Selon le protocole IPv6 Mobile, trois méthodes sont proposés pour que le MN puisse recueillir les événements suivants comme les signes du déclenchement du handover de niveau L3. Ces événements sont Routeur non-accessible, Absence de réception du message "Annonce de routeur", ou Signe du déclenchement du handover de niveau 2. Une fois que ces événements sont engendrés, le MN lance la procédure de Découverte de routeur. Si le MN découvre le nouveau routeur avec un préfixe différent du sien, il conclut que le handover de niveau L3 est entamé, qu'il s'est connecté à un nouveau réseau. Cependant ces événements qui se sont manifestés n'indiquent assurément pas le déclenchement du handover de niveau L3, par exemple, le routeur peut être non-accessible à cause d'une panne.

b. Auto-configuration d'adresses

Une fois que le MN détecte le changement de réseau, il doit lancer la phase d'Auto-configuration d'adresses pour générer une nouvelle adresse IP sur ce nouveau réseau. Le MN ne peut pas communiquer avec les CN avant l'accomplissement de cette phase. Cette phase peut prendre une seconde au minimum ou durer jusqu'à cinq secondes, le délai de la procédure DAD (Duplicate Address Detection) représente une grande partie du délai total.

c. Mise à jour d'association

La phase de Mise à jour d'association a pour objet de mettre à jour la table d'associations du HA et celle du CN afin que le HA puisse faire suivre les paquets à destination d'adresse mère du MN vers l'adresse temporaire de ce dernier et que le CN puisse communiquer directement avec le MN sans passer par le HA.

Comme nous l'avons décrit, la phase de Mise à jour d'association est décomposée également en trois phases : la phase de Mise à jour d'association avec le HA, la phase de Routabilité de retour et la phase de Mise à jour d'association avec le CN.

Mise à jour d'association avec le HA

Dès que le MN finit la phase d'Auto-configuration d'adresses, il envoie le message "Mise à jour d'association" (Binding Update) au HA pour mettre à jour la table d'associations (Binding Cache) du HA qui contient l'association entre l'adresse mère et l'adresse temporaire du MN. Lorsque le HA reçoit ce message, il actualise cette table d'associations et envoie un message "Acquittement de Mise à jour d'association" (Binding Acknowledgement) au MN comme la réponse. Le HA intercepte ainsi les paquets destinés à l'adresse mère du MN, soit en diffusant un message "Annonce de voisin" comme s'il

était le MN, soit en répondant aux messages "Sollicitation de voisin" à la place du MN, ensuite, il envoie ces paquets à l'adresse temporaire du MN qui est enregistré dans la table d'associations.

D'ailleurs, le MN et le HA doivent utiliser le protocole IPSec pour protéger les messages de signalisation, mais cella n'ai pas obligés.

La figure (Fig 2.7) présente la phase de Mise à jour d'association ainsi que le message "Mise à jour d'association" et le message "Acquittement de Mise à jour d'association".

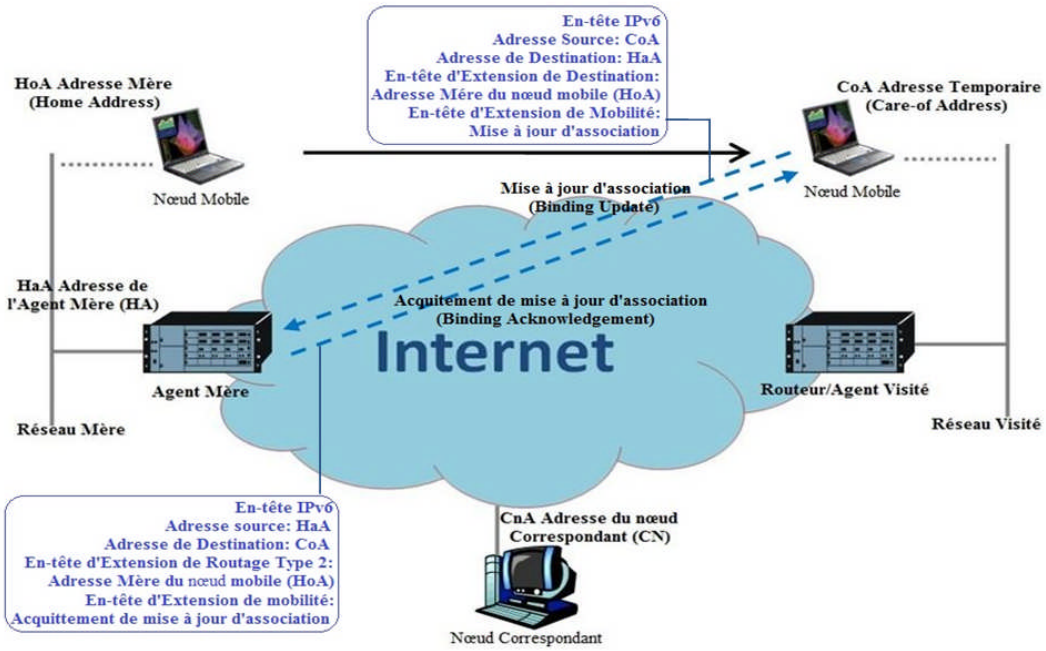

Fig. 2.7- Mise à jour d'association entre le MN et le HA

Routabilité de retour

A la différence du protocole IPv4 Mobile, pour le protocole IPv6 Mobile, lorsque le MN se trouve sur un réseau visité, il peut communiquer directement avec le CN sans passer par le HA. Ce mécanisme est baptisé "le routage de paquets optimisés ". Cependant, pour pouvoir utiliser ce mécanisme, non seulement le MN et le HA doivent supporter le protocole IPv6 Mobile, mais le CN doit aussi supporter le protocole IPv6 Mobile. C'est-à-dire que le CN supporte le protocole IPv6, il a une table d'associations et maintient la mise à jour de cette table. Si le CN ne supporte pas le protocole IPv6 Mobile, la communication entre le MN et le CN doit passer par le HA, comme pour IPv4 Mobile.

Comme le MN et le CN ne peuvent pas utiliser le protocole IPSec pour sécuriser les messages échangés entre eux, la procédure routabilité de retour est déployée. Elle a pour but d'établir la preuve au CN que le MN est accessible à son adresse mère et à son adresse temporaire et de déterminer les jetons qui sont utilisés pour décrire une clé de gestion de Mise à jour d'association. Cette clé est employée pour calculer des valeurs de données d'autorisation pour les messages de Mise à jour d'association. le CN envoie le message HoTI à l'adresse mère du MN en passant par le HA et envoie le message CoTI à

l'adresse temporaire du MN en utilisant le chemin direct. Le MN envoie ainsi le message HoT et CoT comme réponse en utilisant le même chemin respectivement.

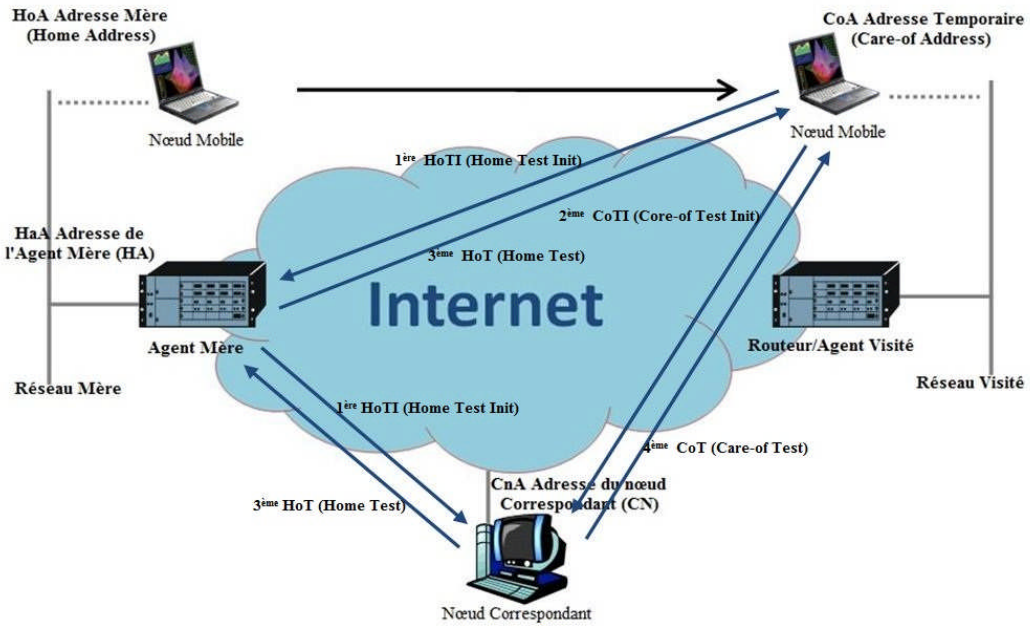

Fig. 2.8- Routabilité de retour

Mise à jour d'association avec le CN

Le MN et le CN utilisent la clé générée dans la phase routabilité de retour pour authentifier les messages – Mise à jour d'association et Acquittement de mise à jour d'association. Lorsque le CN reçoit le message "Mise à jour d'association", il met à jour sa table d'associations et envoie le message "Acquittement de mise à jour d'association" au MN.

Une fois que la table d'association du CN est actualisée, le MN et le CN peuvent communiquer en utilisant le mécanisme de routage de paquets optimisé comme dans la figure (Fig 2.9).

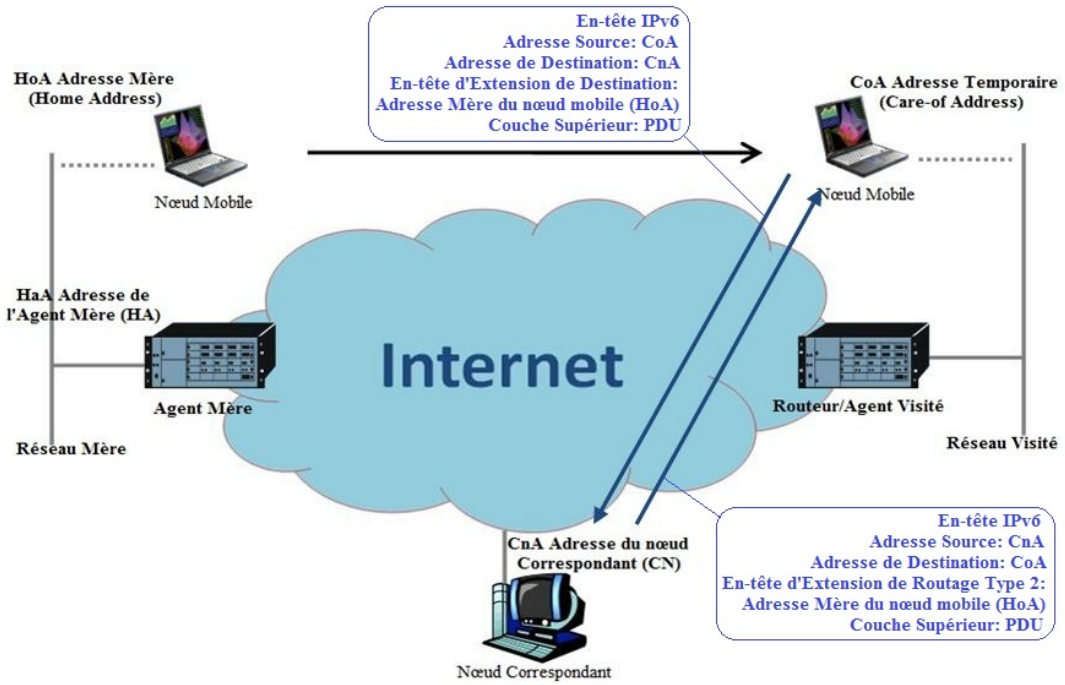

Fig. 2.9- Mécanisme de routage de paquets optimisé

Bien que le protocole IPv6 Mobile permette de résoudre le problème de routage de paquets triangulaire utilisé dans le protocole IPv4 Mobile, il souffre encore de plusieurs faiblesses. Parmi ces faiblesses, nous citons:

- Le délai du handover est long. Particulièrement, le délai de la phase de Détection de mouvement, celui de la phase d'Auto-configuration d'adresses et celui de la phase de Mise à jour d'association sont très long pour les applications en temps réel.
- La perte de paquets pendant le handover peut être important. Le protocole IPv6 Mobile n'a pas proposé une solution pour réduire la perte de paquets.

Pour faire face aux limites du protocole IPv6 Mobile, plusieurs solutions ont été proposées. Le processus de handover peut être amélioré soit en réduisant la perte de paquets, soit en diminuant la charge de la signalisation, soit encore en rendant le processus le plus rapide possible. Parmi les différentes propositions, nous citons les deux principales solutions : le protocole IPv6 Mobile Hiérarchique [10] et le protocole Fast handover pour IPv6 Mobile [11].

II.3 Solutions au niveau transport

La plupart des solutions qui agissent au niveau transport concernent le protocole TCP. Une première classe de propositions lui ajoutent des extensions pour permettre à chacune des deux hôtes à l'extrémité d'une connexion de changer d'adresse IP. Puisque les systèmes d'exploitation implémentent souvent les fonctions TCP dans leur noyau, les modifications au niveau transport sont difficilement applicables à toutes les machines. À cause de ce fait, une deuxième classe de propositions rajoute un proxy dans la connexion TCP qui sépare la communication en deux connexions TCP indépendantes l'une de l'autre. L'hôte mobile et le proxy coopéreront pour continuer une connexion TCP existante à partir d'une adresse IP différente, par contre l'hôte correspondante ne sera pas concernée par ce changement. Nous présentons à la fin de cette section un nouveau protocole de transport proposé pour remplacer TCP et gérer mieux la mobilité des hôtes.

II.3.1 E-TCP

Une première extension du TCP a été introduite par Huitema et s'appelle E-TCP (Extended TCP**)** [12]**.** Les deux hôtes aux extrémités d'une connexion incluent dans chaque segment TCP envoyé un identificateur unique du flot TCP, appelé PCB-ID et qui a une taille de 32 bits.

À l'initiation d'une connexion, les deux hôtes rajoutent leurs PCB-ID locaux aux segments SYN envoyés. Ensuite, les hôtes peuvent s'échanger des paquets E-TCP, dans lesquels la paire de numéros de port est remplacée par leur PCB-ID local. C'est le PCB-ID qui permet ensuite à l'hôte destinataire de regrouper tous les paquets du même flot, même s'ils sont issus d'adresses IP source différentes.

L'auteur reconnaît le fait que sa proposition introduit une brèche dans la sécurité de la connexion, car quelqu'un d'autre pourrait observer le réseau et s'introduire dans la communication en fabriquant des paquets avec le même PCB-ID. Cependant, il soutient que cette brèche est inhérente dans le contexte de la mobilité et explique qu'on peut obtenir une sécurité accrue en utilisant une solution de type IPSec.

II.3.2 TCP Migrate

TCP Migrate [13] utilise les segments SYN échangés à l'initiation d'une connexion pour y rajouter une option appelé Migrate-Permitted et négocier un identificateur unique de la connexion. Cette négociation est sécurisée par une clé secrète, partagée par le protocole Diffie-Hellman [14]. L'identificateur de la connexion est formé par les 64 bits les plus représentatifs du hash SHA-1 (Secure Hash Algorithm) [15] calculé sur les numéros de séquence initiaux et sur la clé secrète partagée. Suite à un déplacement qui implique un changement d'adresse, l'hôte mobile peut reprendre une connexion en cours en envoyant un segment SYN qui contient l'identificateur négocié auparavant (voir Fig 2.11). L'hôte destinataire interprète ce segment comme une demande de reconnexion et l'ancienne connexion est reprise.

II.3.3 Le protocole SCTP

SCTP (Stream Control Transmission Protocol) est un autre protocole de niveau transport qui a été proposé par l'IETF. À la base, SCTP est un protocole de transport orienté connexion qui garanti la réception de paquets de données. Il a eu comme but principal d'apporter des extensions liées au multihoming et multi-flot qui manquaient au TCP. SCTP fournit un moyen pour que chaque hôte puisse envoyer à son correspondant la liste de ses adresses IP possibles. Ceci est fait au début de la connexion et ensuite il est possible d'envoyer et de recevoir des paquets entre n'importe quelles adresses faisant partie des listes échangées par les deux extrémités de la connexion. Une seule paire d'adresses définit le chemin principal utilisé à un instant donné, et toutes les autres forment des chemins de réserve. Cette fonctionnalité de multihoming offerte par SCTP permet d'envisager une solution pour gérer la mobilité et la qualité de service, puisqu'il permet aux deux hôtes d'utiliser plus d'une adresse IP. Puisque notre travail s'articule principalement sur ce protocole, une étude détaillée lui a été consacrée dans le chapitre suivant.

II.4 Solutions au niveau session : Session Migrate

Le concept de session - une association de longue durée entre deux applications qui inclut plusieurs connexions réseau, simultanées ou successives. Dans la suite de protocoles TCP/IP il n'y a pas un niveau session explicit qui se charge de l'établissement d'une session entre deux points en regroupant une ou plusieurs connexions qui partagent un contexte commun. En conséquence, certaines applications ont implémenté elles-mêmes le concept session. Un bon exemple est celui des applications web qui utilisent des cookies dans les requêtes et les réponses HTTP (HyperText Transfer Protocol) [16] pour permettre à une session de continuer en dépit des terminaisons de connexions passagères. Un autre exemple est celui des applications de partage de fichiers P2P, qui reprennent le transfert d'un fichier du point où celui-ci a été interrompu.

Le projet Migrate de MIT est un des premiers qui introduit le niveau session. Il fournit une interface API aux applications pour décrire les points finaux de la session par des identificateurs uniques et stables, différents des adresses IP des points d'attachement où les deux machines se trouvent. Ensuite, les applications fournissent la composition de la session : une ou plusieurs connexions entre les deux points. Un mode de fonctionnement transparent aux applications est aussi possible, dans lequel chaque connexion ouverte par une application est considérée comme une session. Dans ce cas, les identificateurs des points finaux assumés par défaut sont les noms d'hôte correspondants aux adresses IP utilisées dans la connexion.

L'API de session Migrate permet aux applications d'utiliser un espace de noms et un système de résolution de noms quelconque qui n'est pas restreint seulement aux noms d'hôte et au DNS. Ce mécanisme est séparé du déroulement ultérieur de la session. L'application spécifie les noms et le

système de résolution désirés et le niveau session se charge ensuite de la continuation des connexions ouvertes entre les deux points. Au début de la session, il y a une phase de négociation qui tente de créer un canal de contrôle entre les deux points distants. Si les deux point distants sont compatibles Migrate et la création du canal de contrôle réussit, un échange de clés cryptographiques a lieu. Ces clés seront ensuite utilisées pour authentifier les mises à jour échangées ultérieurement.

Suite à un changement d'adresse, un hôte doit informer son correspondant en lui envoyant un message de mise à jour. La réception de ce message peut s'avérer impossible dans le cas où l'autre point a changé d'adresse IP en même temps. Dans ce cas les deux hôtes doivent refaire la résolution des noms, en utilisant les identificateurs et le système de résolution spécifiés au début par l'application. Si le contact réciproque ne réussit toujours pas, Migrate suppose qu'une interruption plus longue a lieu et suspend la connexion. On sauvegarde le contexte de la session et on libère une partie des ressources système et réseau, ce qui pourrait être utile pour les autres connexions.
L'implémentation de session Migrate est réalisée par une couche intermédiaire interposée entre les applications et le système d'exploitation réseau.

II.5 Solutions au niveau application : SIP

SIP (Session Initiation Protocol) [17] est un protocole qui fournit une solution simple l'établissement d'une session multimédia entre deux utilisateurs. Il comporte également un support pour la mobilité de personnes et des machines [18]**,** car l'établissement de la session inclut la localisation des utilisateurs et implicitement de leurs terminaux.

Dans l'architecture SIP, chaque utilisateur fait partie d'un domaine d'origine. Un serveur SIP présent dans chaque domaine contient des informations mises à jour sur l'emplacement actuel de chaque utilisateur du domaine. La partie la plus simple des mécanismes de mobilité fournis par SIP est la localisation de l'utilisateur par ses correspondants, appelée pre-call mobility. Chaque fois qu'une personne se déplace (avec son terminal ou en le changeant), le serveur SIP du domaine d'origine est informé sur l'adresse IP où elle peut être jointe. Les correspondants utilisent un nouveau type d'enregistrement DNS apprendre l'adresse du serveur SIP d'un certain domaine. Ensuite, les correspondants contactent ce serveur, en lui envoyant une requête INVITE.

II.6 Conclusion

Dans ce chapitre, nous avons présenté un panorama des solutions de gestion de la mobilité des nœuds dans les réseaux IP. Nous avons expliqué particulièrement le mécanisme de gestion de mobilité du protocole IPv6 Mobile. Comme il existe des problèmes en termes de délai et de perte de paquets pour la mobilité du MN géré par le protocole IPv6 Mobile, plusieurs propositions sont faites pour résoudre ces problèmes, telles que le protocole HMIPv6 et FMIPv6.

D'autres solutions proposent d'actionner au plus haut niveau. Ces propositions introduisent des nouvelles options de TCP qui permettent de notifier à l'hôte correspondant un changement d'adresse. Le nouveau protocole SCTP permet quant à lui d'utiliser plusieurs adresses IP aux deux extrémités, et cela même simultanément pour les machines multi-domiciliées. À la fin, nous avons présenté deux autres solutions qui opèrent cette fois au dessus des niveaux IP et transport : la solution session Migrate et le protocole SIP. Le chapitre suivant est consacré au protocole SCTP et à son extension pour supporter la qualité de service basée sur le type de flot.

Chapitre 3

Extension du SCTP avec le Support d'Equilibrage de Charge

III Extension du SCTP avec le Support d'Equilibrage de Charge

III.1 Introduction

Ce chapitre présente quelques aspects et fonctionnalités du protocole de transport SCTP. Le SCTP est un nouveau protocole de couche transport qui a été conçu par l'IETF pour fournir un transport fiable et sécurisé d'une variété d'applications, principalement la signalisation du réseau de téléphonie publique, sur les réseaux IP. Bien que TCP soit le protocole de transport le plus populaire dans le monde de l'IP, SCTP fournit en plus des fonctionnalités offertes par TCP. Parmi les contributions apportées par SCTP nous identifions principalement le Multihoming. Nous étudierons ensuite le protocole Mobile SCTP (mSCTP), et nous nous intéresserons en particulier à la problématique de couplage de la mobilité avec le multihoming pour la gestion de la qualité de service. Introduire l'aspect du multihoming pour une meilleure gestion de la qualité de service, dans les réseaux mobiles, reste un défi majeur et encore largement ouvert.

Enfin, notre contribution sera l'extension de mSCTP avec des mécanismes d'équilibrage de charge en nous basant sur la nature des accès multiples disponibles pour le nœud mobile et le type de flux transmis par le nœud correspondant.

III.2 Présentation générale du protocole SCTP

III.2.1 Généralités

SCTP (Stream Control Transmission Protocol) est un protocole de transport proposé par l'IETF [20]. Il a été conçu pour palier certaines limitations inhérentes à TCP lors du transport de signalisation téléphonique sur IP. SCTP apporte un service de transport fiable de messages issus des applications utilisateurs (e.g. protocoles de signalisation). Il est orienté connexion. Une connexion SCTP est

appelée association (Fig 3.1). Il est possible de multiplexer plusieurs flux de messages au sein d'une même association (service de multistreaming). Ceci se traduit au niveau paquet par la possibilité de transporter plusieurs messages de signalisation dans un même paquet SCTP. Les messages sont encapsulés dans des structures de données appelés chunks. Les chunks sont eux mêmes encapsulés dans des paquets SCTP.

SCTP permet également de mettre en œuvre le multihoming en introduisant la possibilité d'associer plusieurs adresses IP à un même port SCTP (Fig 3.1). Plusieurs chemins sont alors disponibles pour mettre en relation deux nœuds SCTP distants. A un instant donné, seul un chemin est actif (i.e. est utilisé pour transporter les données) les autres chemins sont utilisés comme sauvegarde au cas où le chemin actif deviendrait indisponible.

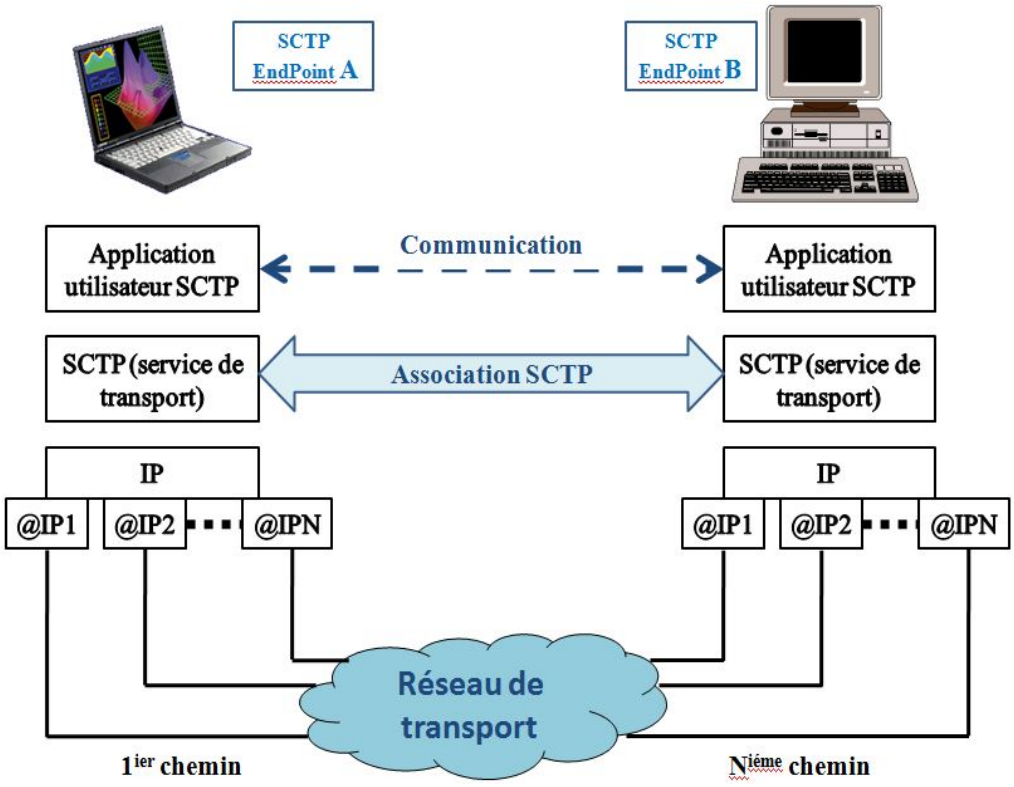

Fig. 3.1- Schéma d'une association SCTP.

Une association SCTP est établie à la demande d'une application. L'établissement de l'association se fait en deux phases impliquant l'échange de quatre messages (4-way Handshake ,cf. Fig 3.5). Durant la première phase d'établissement de l'association, un mécanisme de sécurité est mis en place afin d'éviter les attaques de Deny of Service (ce mécanisme qui utilise des Cookie, est décrit dans **[19]**. La deuxième phase est celle qui établit effectivement l'association en réservant les ressources associées aux sockets1 de l'association sur chaque point terminal. Il est également possible au cours de cette deuxième phase d'envoyer des données utilisateurs dans les paquets d'établissement (chunk Cookie Echo et Cookie Ack).

Le service de transport SCTP peut être décomposé en un ensemble de fonctions (Fig 3.2) dont le descriptif de chacune d'elles est donné par [20].

Une des caractéristiques du protocole SCTP est qu'en cas de nécessité, il permet de fragmenter les messages utilisateurs afin de s'assurer que les paquets SCTP soient passés aux couches inférieures conformément au PMTU (Path Maximum Transmission Unit) spécifié. En réception les fragments sont rassemblés en messages complets avant qu'ils ne soient dirigés vers le nœud SCTP de destination. Les différents fragments d'un même message portent le même SSN (Stream Sequence Number). SCTP attribue un TSN (Transmission Sequence Number) à chaque fragment de données utilisateur. Le protocole SCTP utilise une procédure d'acquittement sélectif. La retransmission de paquet est conditionnée par les procédures de contrôle de congestion décrites ultérieurement dans ce chapitre.

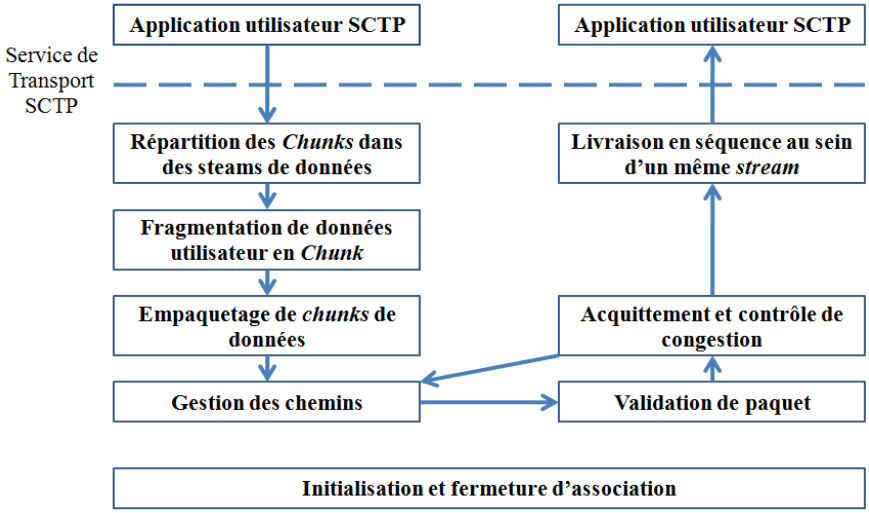

Fig. 3.2 – Fonctions du service de transport SCTP.

III.2.2 Format générale d'un paquet SCTP

Un paquet SCTP (Fig 3.4), constituant la charge utile d'un paquet IP, est constitué d'un entête commun suivi d'un ou plusieurs chunks. Les chunks étant des blocs de données de taille variable qui peuvent appartenir à des streams différents. Chaque chunk (Fig 3.5) peut contenir aussi bien des informations de contrôle (nous parlons alors de chunk de contrôle) que des données utilisateur (nous parlons alors de chunk de données (Data) (Fig 3.3). Les chunks de contrôle précèdent toujours des chunks de données dans un paquet SCTP.

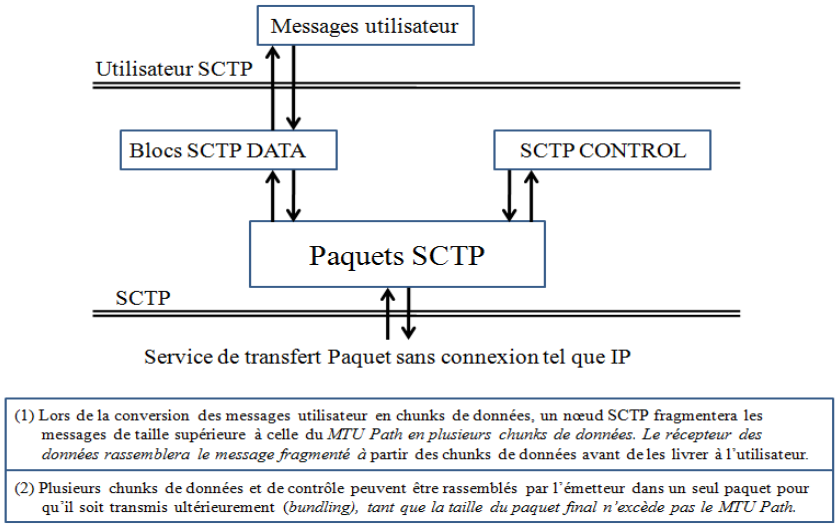

Fig. 3.3- Transfert de données utilisateur.

Source Port Number	Destination Port Number
Verification Tag	
Checksum	
Chunk – 1	
⋮	
Chunk – N	

Fig. 3.4- Format du Paquet SCTP.

Chunk type	Chunk flags	Chunk Lenth
TSN		
Stream ID		SSN
Protocol ID		
User Data		

Fig. 3.5- Format d'un Chunk-I (DATA, Type=0) SCTP.

L'entête commun d'un paquet SCTP est de taille 12 octets. Pour la détection des erreurs de transmission chaque paquet SCTP est protégé par un Checksum de taille 32 bit (Suivant l'algorithme Adler-32). Le checksum sert pour la détection des erreurs. Il est plus robuste que les 16 bit de TCP et UDP [21]. L'entête commun contient également un champ appelé «Verification Tag» de taille 32 bit. Ce champ identifie l'association au niveau d'un nœud SCTP (il identifie l'émetteur). Le Verification

Tag est choisi aléatoirement par chaque nœud SCTP lors de l'ouverture de l'association. L'entête commun est suivi des différents chunks de contrôle et de données.

Chaque chunk est formé également d'un entête suivi de la charge utile de longueur variable. Il s'agit des données transmises de l'émetteur vers le récepteur. L'entête de chaque chunk contient les champs suivants :

– Chunk Type (8 bits) : permet de déterminer le type de la charge utile du chunk (données ou informations de contrôle...),

– Chunk Flag (8 bits) : utilisé différemment selon le type de chunk. À ce niveau le champ Flag est plus précis c'est qu'il contient des indicateurs du type de livraison en séquence ou dans le cas de séquencement de données non requis. Le tableau 3.1 résume les états de fragmentation possibles pour un message utilisateur en fonction des bits du champ Flag. De plus ce champ contient des bits réservés (Reserved 5bits) qui doivent être mis à "0" et ignorés par le récepteur.

U (1bit) : Unordered	B (1bit) : Begining	E (1bit) : Ending	Description
1	X	X	Ce bit indique au récepteur dignorer le numéro de séquence du Chunk
0	1	0	Premier Chunk du messege fragmenté
	0	0	Chunk intermédiaire un message fragmenté
	0	1	Dernier Chunk du message fragmenté
	1	1	Message non fragmenté

Tableau 3.1- Description des bits du champ Chunk flag.

– Chunk Length (16 bits) : indique la taille en octets des données transmises. Ce paramètre représente la taille du chunk de données en octets incluant les champs chunk type, chunk flags, chunk length et chunk value. Si le chunk ne contient pas des informations à transmettre (i.e le champ chunk value est de longueur nulle), le champ length sera affecté à 4 [20]. Le champ chunk length ne compte pas les bits de bourrage (padding). En effet, la longueur d'un chunk de données doit être un multiple de 4 octets. Si elle ne l'est pas, l'émetteur doit remplir le chunk avec des zéros (en octets) [**20**] et ce bourrage n'est pas inclut dans le champ chunk length. L'émetteur ne peut jamais remplir plus que 3 octets. Le récepteur doit ignorer les octets de bourrage.

L'entête n'est pas suivi uniquement de charge utile à transmettre, mais aussi des champs de contrôle qui varient en fonction du type de chunk. Dans ce qui suit nous abordons seulement le type 0 relatif au chunk Data (Fig 3.5), les autres types sont détaillés dans [20] et [22]. Les champs de contrôle sont principalement :

– Le numéro de séquence TSN (Transmit Sequence number) sur 32 bits : numéro de séquence des fragments. Le TSN permet de reconstruire un message fragmenté.

– L'identité du flot (stream ID) (sur 16 bits) : permet d'identifier un flot (stream). Ce qui offre la possibilité d'avoir plusieurs streams dans un même paquet SCTP.

– Le numéro de message SSN (Stream Sequence Number) dans le flot, sur 16 bits : il s'agit du numéro de séquence du chunk dans le stream. Le SSN permet le réordonnancement des données par le récepteur.

– Le champ Protocol_ID, codé sur 32 bits, sert à indiquer le type d'information contenue dans les chunks de données. C'est une couche non utilisée par SCTP mais par la couche application.

Enfin la charge utile est contenue dans le champ User Data. C'est un champ de taille variable, il s'agit des données appartenant au stream échangées entre deux nœuds SCTP (émetteur/récepteur).

III.2 .3 Établissement d'une association

La procédure d'ouverture d'une association SCTP tel qu'elle est décrite dans [20] est donnée par la figure (Fig 3.6).

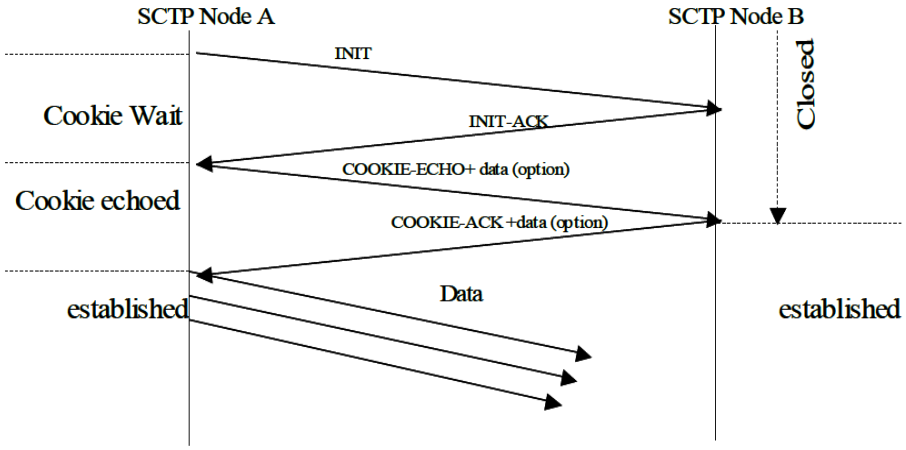

Fig. 3.6- Phase d'initiation : échange quadruple

L'établissement d'une association s'effectue en quatre temps (alors que en 3 temps en TCP). Cela peut paraître comme inconvénient pour le temps mis pour le déclenchement, mais des données peuvent être transmises au 3ième et 4ième temps du «4-way startup handshake» [20]. De plus, ce mécanisme permet au serveur SCTP de se protéger de plusieurs types d'attaques. Dans ce qui suit nous décrivons les quatre étapes de l'établissement d'une association SCTP entre un nœud A (client) et un nœud B (serveur).

Étape n°1

L'association étant à l'état CLOSED du côté du serveur (le nœud B) qui attend un message du nœud A avec un chunk INIT, dont le format est donné par la figure (Fig 3.7). Ce chunk signale la requête qu'un client (nœud A) demande la création d'une association. Le client A envoie un chunk INIT à destination du serveur B et déclenche un temporisateur lui permettant de réémettre ce paquet

(contenant le chunk INIT) en l'absence de réponse. Ensuite le client A se met en attente du paquet de réponse contenant le cookie.

Type=1	Chunk flags	Chunk length
Initiate Tag		
Advertised Received Window Credit (a_rwnd)		
Number of Outbound Streams	Number of Inbound Streams	
Initial TSN		
Optional/Variable –Length Parameters		

Fig. 3.7 – Format du Chunk INIT.

Dans ce chunk, le client A copie son Verification Tag (nous le notons par Tag-A dans la suite de ce chapitre) dans le champ Initiate Tag du chunk INIT. Le Tag-A est un nombre choisi aléatoirement par le client A entre 1 et (232-1). Après avoir émis le chunk INIT, le client A déclenche le compteur T1-INIT (Timer) et l'association, du côté client A, aura un état COOKIEWAIT.

Il est important que la valeur initiale du Tag (Initiate-Tag) soit attribuée aléatoirement afin de se protéger contre les attaques de type «man in the middle» «sequence number» ([20]). De même pour le champ initial TSN. Deux champs de 16 bits chacun (Number of Inbound Streams et Number of Outbound Streams) servent à indiquer le nombre maximal de streams entrants et sortants que le client et le serveur peuvent supporter au niveau de l'association en cours d'établissement.

Étape n°2

Une fois le chunk INIT reçu, le serveur B génère un cookie regroupant les informations permettant d'établir la connexion avec le client A. Ensuite le serveur B crée une signature numérique [22] du cookie appelée MAC (Message Authentication Code). Le serveur B inclut le MAC au cookie et envoie l'intégralité au client A dans un chunk INIT-ACK dont le format est donné par la figure (Fig 3.8). L'adresse IP de destination, utilisée par le serveur B, doit être celle de l'adresse IP source du chunk INIT. L'association garde l'état CLOSED du côté serveur B.

Type=2	Chunk flags	Chunk length
Initiate Tag		
Advertised Received Window Credit (a_rwnd)		
Number of Outbound Streams	Number of Inbound Streams	
Initial TSN		
Optional/Variable –Length Parameters		

Fig. 3.8 – Format du Chunk INIT-ACK

Dans sa réponse, le serveur B doit recopier la valeur de Tag-A dans le champ Verification-Tag et fournir son propre Verification-Tag (Tag-B) au niveau du champ Initiate Tag. Le chunk INIT-ACK contient (dans le champ State COOKIE Parameter) en plus du cookie et du MAC, une indication sur l'instant de création du cookie, sa durée de vie, ainsi que toutes les informations qui lui sont nécessaires afin d'établir l'association.

Pour générer un cookie certaines étapes doivent être considérées par le serveur B [**20**] :

– Créer une association TCB (Transmission Control Block) en utilisant les informations du chunk INIT reçu et du chunk INIT-ACK à émettre,

– Dans le TCB on conserve la date de création du cookie et sa durée de vie contenue dans le paramètre «valid cookie life»,

– A partir du TCB, on collecte un minimum d'informations utilisé pour recréer le TCB, et on génère un MAC qui est un hash du cookie chiffré avec la clé privée du serveur B,

– Générer le cookie en combinant ces informations et le MAC résultant.

Après l'émission du chunk INIT-ACK avec les paramètres du cookie, le serveur B doit supprimer le TCB et toute information liée à la demande d'établissement de la nouvelle association avec le client A. Toutes les données nécessaires à la connexion étant envoyées dans le cookie, aucun buffer de mémorisation n'est alloué à l'association tant que la connexion n'est pas totalement établie. Cet aspect protège le protocole SCTP de certaines attaques de type «SYN Attacks»[**23**].

Étape N°3

Après la réception d'INIT-ACK du serveur B, le client A doit arrêté son compteur T1-INIT Timer et quitter l'état COOKIE-WAIT. Le client A doit alors :

a)- Émettre le cookie qu'il a reçu dans un chunk COOKIE-ECHO (Fig 3.9),

b)- Déclencher son compteur T1-COOKIE Timer et

c)- Entrer dans l'état de COOKIE-ECHOED.

Type=10	Chunk flags	Chunk length
Cookie		

Fig. 3.9 – Format du Chunk COOKIE-ECHO

A ce stade de la procédure d'initialisation de l'association, le serveur B vérifie la signature électronique MAC du cookie contenue dans le chunk COOKIE-ECHO venant du client A, et qui a été retournée telle quelle. Cette vérification permet au serveur B de contrôler qu'il s'agit bien du cookie qu'il l'a déjà envoyé au client A et que les données de connexion n'ont pas été modifiées. Le client A, de son côté, peut ajouter au paquet envoyé, contenant le chunk COOKIE-ECHO, des chunks de données mais il doit respecter l'ordre des chunks dans le paquet (les chunks de contrôle sont placés les premiers devant les chunks de données). Cependant, le client A ne doit émettre aucun autre paquet vers le serveur B jusqu'à la réception de la confirmation d'établissement de l'association de la part du serveur B.

Étape N°4

Suite à la réception du chunk COOKIE-ECHO, le serveur B doit répondre avec un chunk COOKIE-ACK (Fig 3.10) après la construction du TCB. Le serveur B, après vérification du cookie et du MAC, signale au client A que l'association est établie par l'envoie d'un paquet contenant un chunk COOKIE-ACK et l'association, du côté serveur, passe à l'état ESTABLISHED. Un chunk COOKIE-ACK peut être empaqueté avec des chunks de données ou de contrôle (SACK chunks par exemple).

Le client A à la réception d'un chunk COOKIE-ACK, conclut l'ouverture de l'association SCTP et informe son ULP (Upper Layer Protocol) du succès de l'établissement de l'association avec une primitive Communication UP Notification3. C'est à partir de cet instant que le client A commence à émettre des paquets à destination du serveur B. L'association du côté client A passe donc à l'état ESTABLISHED.

Fig. 3.10 – Format du Chunk COOKIE-ACK

Remarque :

– Lorsque le TCB est crée, chaque point terminal doit affecter son Cumulative TSN Ack Point interne à la valeur de son Initial TSN transmis auquel on retranche 1 [20]**.**

– Les paquets SCTP contenant les chunks COOKIE-ECHO et COOKIE-ACK peuvent tous comporter des chunks DATA. Ces données ne seront traitées par le serveur que si l'association est établie.

– Si un nœud SCTP recevant soit un chunk INIT, INIT-ACK ou COOKIE-ECHO décide de ne pas établir la nouvelle association suite à un manque de paramètres obligatoires dans les chunks INIT ou INIT-ACK reçus, ou à des valeurs de paramètres invalides ou à un manque de ressources locales, il doit répondre par un chunk ABORT (Fig 3.11). Il doit spécifier aussi la cause de ce message ABORT, en incluant les paramètres causant l'erreur dans le chunk ABORT (par exemple le type des paramètres obligatoires manquants).

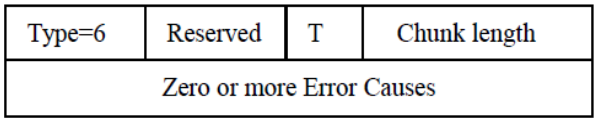

Fig. 3.11 – Format du Chunk ABORT.

III.2.4 Terminaison d'une association

Pour accomplir sa fiabilité, SCTP doit assurer une procédure de fermeture de connexion. La terminaison normale d'une association SCTP est basée sur une procédure à trois temps (Fig 3.12). Cette fermeture normale de l'association est similaire à TCP à la différence que SCTP ne supporte pas

l'état de «semi-connecté» [**24**]. Chaque nœud SCTP attend la confirmation de la réception des chunks de données en cours de transmission avant de libérer l'association. Il existe dans SCTP une autre manière de terminer une association entre le client et le serveur, c'est une méthode plus brutale. Il s'agit d'une procédure ABORT, dont un exemple est présenté ci-dessus, qui signale uniquement qu'un des nœuds terminaux de l'association s'est retiré. Cette fermeture brutale se produit lorsqu'un arrêt immédiat est requis par les applications (situation d'exception, occurrence d'erreurs irrécupérables).

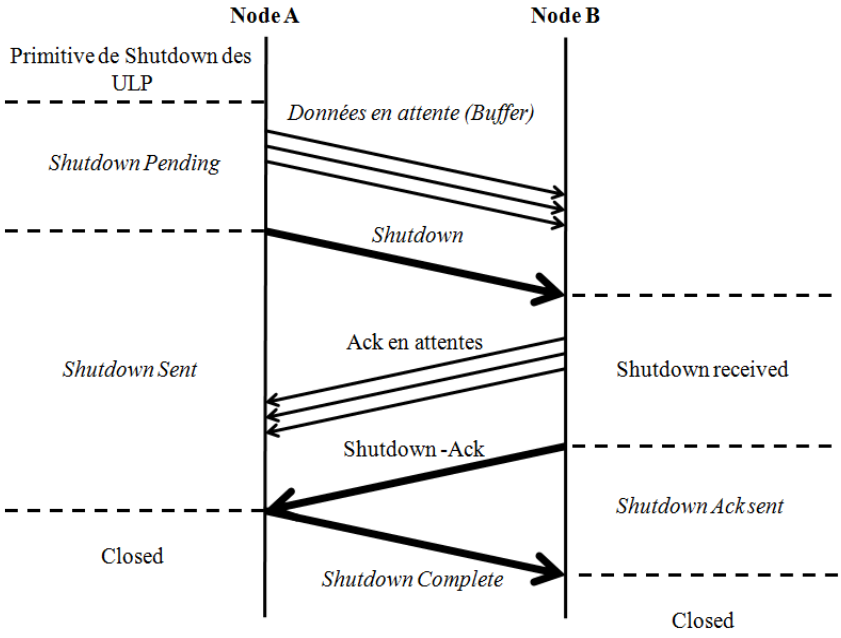

Fig. 3.12 – Terminaison d'une association.

1ère Phase :

Le client A désire mettre fin à l'association suite à la réception d'une primitive SHUTDOWN de sa couche supérieure. À cet instant l'association du côté client A change d'état en SHUTDOWN–PENDING. Le client A n'accepte plus de données à émettre de sa couche supérieure et assure le transfert de tous les messages qui sont stockés dans le buffer à destination du serveur B avant d'envoyer le chunk Shutdown vers le serveur B. Le client A vérifie enfin que tous les acquittements des paquets SCTP précédemment envoyés ont été reçus. Cependant, le client A peut retransmettre des données qui ont été perdues et signalées par le serveur B et l'association du côté du client passe à l'état SHUTDOWN-SENT.

2ème Phase :

Ensuite, le client A doit émettre un chunk SHUTDOWN (Fig 3.13) vers son nœud de destination (le serveur B). Le client A précise dans ce chunk le dernier TSN reçu du serveur et également les paquets perdus. Lors de l'émission du SHUTDOWN le client A doit déclencher un compteur T2-shutdown Timer. Dans le cas où ce compteur expire, le message SHUTDOWN doit être réémis avec la mise à jour du dernier TSN reçu en séquence. À la réception d'un chunk SHUTDOWN l'association du côté du serveur B change d'état en SHUTDOWN-RECEIVED. À son tour le serveur B n'accepte plus

de nouveaux paquets en provenance du client A et arrête de transmettre de nouvelles données de ses couches supérieures (ULP). Le serveur B fini par émettre les données stockées dans son buffer de transmission et vérifie que tous les paquets précédemment transmis ont été reçus au moyen de la valeur de TSN fournie dans le chunk SHUTDOWN. Pour chaque paquet reçu, le client A acquitte les paquets reçus par un chunk SACK, envoie un chunk SHUTDOWN et réinitialise son temporisateur.

Type=7	Chunk flags	Length =8
Cumulative TSN Ack		

Chunk flags : 8 bits, mis à zéro lors de la transmission et ignoré en réception
Length : 16 bits (unsigned integer), indique la longueur du paramètre mis à 8,
Cumulative TSN Ack : 32 bits (unsigned integer), ce paramètre contient le TSN du dernier chunk reçu en séquence avant toute pertes (gap).

Fig. 3.13 – Format du Chunk shutdown

Une fois que le serveur B a atteint l'état SHUTDOWN-RECEIVED, il doit ignorer toute demande de shutdown de l'association provenant de son ULP.

Enfin, une fois tous les paquets en attente envoyés et reçus par le client A, le serveur B émet un chunk SHUTDOWN-ACK (Fig 3.14) et déclenche son temporisateur T2-shutdown Timer. C'est à partir de là que l'association du côté serveur change d'état en SHUTDOWN-ACK-SENT. Si son temporisateur expire, le serveur B doit réémettre le chunk SHUTDOWN-ACK.

Type=8	Chunk flags	Chunk length=4

Fig. 3.14 – Format du Chunk Shutdown-Ack

3ème Phase :

Suite à la réception d'un SHUTDOWN-ACK, le client A, s'assure qu'il n'a pas eu de perte de données de côtés des deux intervenants dans l'association, arrête son temporisateur T2-shutdown Timer et envoie un chunk SHUTDOWN-COMPLETE (Fig 3.15) au serveur B. Le client A peut alors quitter l'association et supprimer toute information liée à la connexion et l'état de l'association passe définitivement à CLOSED.

De son côté le serveur B est garanti de la bonne réception des données au cours de l'association courante en recevant un chunk SHUTDOWN-COMPLETE. À cet instant, le serveur B vérifie qu'il est dans l'état SHUTDOWN-ACK-SENT, si ce n'est pas le cas le chunk sera ignoré. S'il est bien dans l'état SHUTDOWN-ACK-SENT, le serveur B arrête son T2-shutdown Timer et supprime l'association (et ainsi l'association entre dans l'état CLOSED).

Type=14	Reserved	T	Length =4

Reserved : 7 bits (mis à 0 lors de transmission et ignoré en réception)
T : 1bit, T=0 si l'émetteur a un TCB qui a été détruit sinon T=1.

Fig. 3.15 – Format du Chunk SHUTDOWN-COMPLETE

III.2.5 Transfert des données utilisateurs (gestion des acquittements)

D'après la **[20**] la transmission de données en SCTP doit se produire uniquement lors des états, décrits précédemment, suivants :

– ESTABLISHED

– SHUTDOWN-PENDING

– SHUTDOWN-RECEIVED

L'exception de ceci, est lorsque les chunks de données sont autorisés d'être transmis avec un chunk de contrôle de type COOKIE-ECHO quand l'association est dans l'état COOKIE-WAIT. Un récepteur SCTP doit être capable de recevoir au minimum un paquet SCTP de taille 1500 octets, c'est à dire qu'un nœud SCTP ne doit pas indiquer moins de 1500 octets dans son envoie initial (arwnd = 1500 byte) de INIT ou INIT-ACK [22]. Pour assurer l'efficacité de transmission, SCTP définit des mécanismes pour l'empaquetage des messages utilisateurs de petites tailles et la fragmentation de ceux de tailles importantes [20]**.**

Le protocole SCTP, possède des mécanismes de contrôle de flux et de congestion. Il permet, grâce à sa caractéristique de Multistreaming, l'envoie de données correspondant à plusieurs flots (streams) dans un même paquet grâce à un mécanisme d'identification de stream dans chaque chunk. Lors de transfert de données, si un message utilisateur est de taille supérieure au PMTU (Path Maximum Transmission Unit) d'une association, il sera fragmenté en différents chunks. Les chunks formant un même message ont des TSNs successifs, et sont identifiés par le même SSN. L'identification de l'ordre des différents morceaux d'un message utilisateur est assurée au moyen des flags B/E (premier chunk, les chunks intermédiaires, dernier chunk). Pour la transmission des données, SCTP assure un acquittement sélectif au moyen du SACK chunk. Ce dernier est utilisé par le récepteur SCTP qui l'envoie en retour afin d'acquitter les DATA chunks reçus. Le SACK chunk permet à l'émetteur d'identifier la fenêtre avertie de réception, les chunks qui ont été perdus, les chunks qui ont été reçus dupliqués, etc. Ainsi toutes les informations nécessaires à d'éventuelles retransmissions ou à la gestion de streams.

Lorsque un chunk de données (DATA chunk) (Fig 3.16) atteint un récepteur SCTP, ce dernier doit envoyer un SACK (Fig 3.16) afin d'acquitter sa réception. Un champ Cumulative TSN Ack est utilisé pour acquitter la réception de tous les chunks reçus en séquence sans erreurs. Ce champ indique jusqu'à quelle valeur de TSN des données ont été correctement reçues. Un SACK contient aussi les

Gap Ack chunks qui sont utilisés pour acquitter des plages de chunks de données reçues séparément. Le Gap Ack chunk Start et Gap Ack chunk End sont codés sur 16 bits. Ils indiquent la position relative (par rapport au Cumulative TSN Ack) des différentes plages de chunks isolés correctement reçus. Le SACK comporte également une indication sur les TSNs dupliqués, donnée par les champs Number of duplicate TSNs et Duplicate TSN. Le champ Number of duplicate TSNs, codé sur 16 bits, indique le nombre de chunks dupliqués. Le champ Duplicate TSN (sur 32 bits) donne le nombre de fois qu'un TSN a été reçu de façon dupliquée à partir du dernier SACK transmis. Un TSN dupliqué apparaît dans le SACK autant de fois que le récepteur l'a reçu de façon redondante.

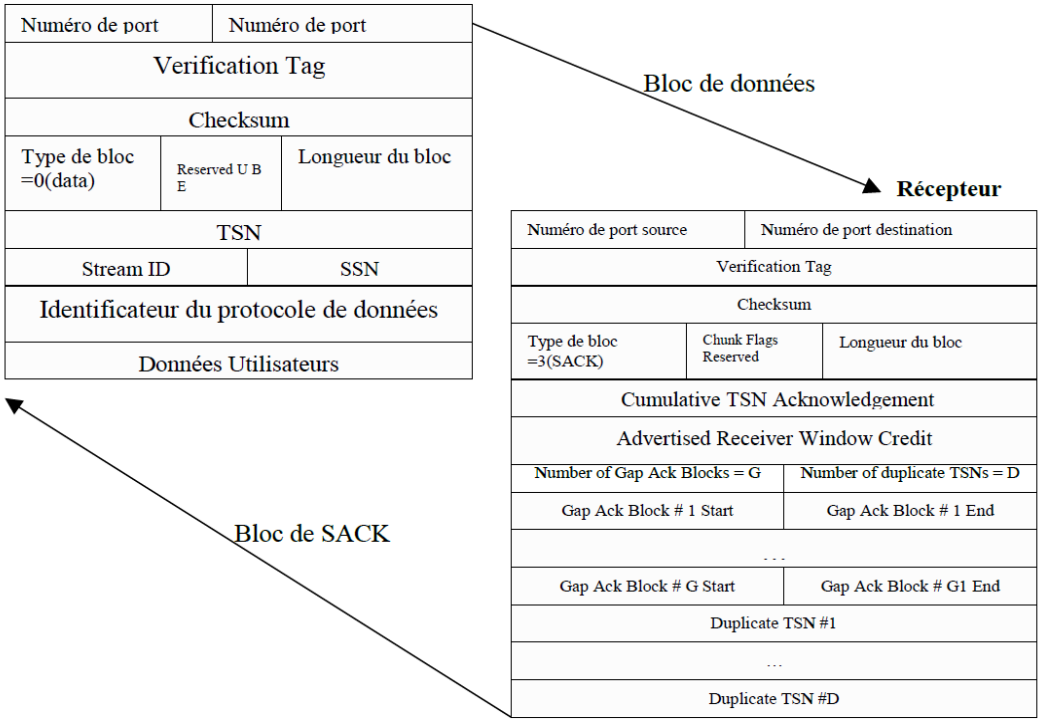

Fig. 3.16 – Transmission de données.

L'exploitation des données transmises par le chunk SACK, permet à l'émetteur SCTP de réémettre les DATA chunks qui n'ont pas été reçus et qui sont identifiés par leur numéro TSN. Cette retransmission s'effectue après l'expiration du temporisateur ou la réception de 4 chunks SACK indiquant la perte du même DATA chunk [**20**].

Nous constatons que SCTP acquitte les paquets reçus non pas les octets comme c'est le cas de TCP. S'il y a eu un problème lors de la transmission des données, et qu'une retransmission de ces dernières a eu lieu, les DATA chunks hors séquence sont acquittés à l'aide du champ Gap Ack Blocks, qui permet l'acquittement d'une ou plusieurs séquences de TSN. Notons que comparativement à TCP, SCTP permet d'avoir de plus de GAP Blocks dans un chunk SACK [**25**]. Il offre un espace réservé de taille 216 octets comme le spécifie le champ chunk length.

Exemple illustratif pour Gap Ack chunks :

Supposant qu'un point terminal SCTP (serveur) vient de recevoir les paquets de données suivants (Fig 3.17) :

TSN1	TSN2	Gap1	TSN4	TSN4	Gap2	TSN6

.

Fig. 3.17 – Numéros des paquets SCTP reçus.

Le SACK à émettre sera le suivant (Fig 3.18) :

Cumulative TSN Ack = 2	
a_rwnd = X	
G= 2	D= 1
# 1 start =2	# 1 end =2
# 2 start =4	# 2 end =4
TSN = 4	

Fig. 3.18 – Le Chunk SACK transmis.

Ce SACK indique la bonne réception des paquets dont le TSN vérifie l'une des conditions suivantes :

– TSN≤cumulative TSN Ack : TSN1 et TSN2,

– n ° 1 start + cumulativeTSN Ack≤TSN≤n ° 1 end + cumulativeTSN Ack soit 2 + 2≤TSN≤2 + 2 : TSN4,

– n ° 2 start + cumulativeTSN Ack≤TSN≤n ° 2 end + cumulativeTSN Ack soit 2 + 4≤TSN≤2 + 4 : TSN6.

Les TSN 3 et TSN 5 réfèrent aux paquets perdus. Ce SACK indique également la réception dupliquée du paquet dont le TSN est égale à 4.

III.3 Contrôle de Congestion en SCTP

SCTP utilise les mêmes modes de contrôle de congestion que TCP en se basant sur la norme [**26**], à savoir le slow-start et la congestion avoidance. La principale différence avec TCP réside dans le contrôle adapté au multihoming. Les paramètres de congestion étant gérés indépendamment du chemin qu'il soit primaire ou alternatif. Le contrôle de congestion dans SCTP est toujours appliqué à l'association entière et non pas à des flots (stream) individuels. Pour contrôler les chunks reçus, SCTP attribue un TSN à chaque chunk de données utilisateur. Ce TSN est indépendant de tous les SSNs attribués au niveau du flot de données (stream).

Dans un réseau, de manière générale, la congestion peut se produire à deux niveaux : au niveau du récepteur (taille du buffer de stockage assez faible) ou au niveau du réseau (bande passante de la liaison saturée). Dans le premier cas, le problème est résolu avec le champ Advertised Receiver Window Credit (a_rwnd), qui se trouve dans les chunks de contrôle de type INIT, INIT-ACK et SACK. Le deuxième cas, plus complexe à gérer, est résolu à l'aide de plusieurs algorithmes. Ces algorithmes sont les mêmes que ceux utilisés dans TCP [**20**], et utilisent les deux variables suivantes pour chaque adresse de réception d'une association :

– Congestion window (cwnd) : qui limite le nombre d'octets que le nœud émetteur des données peut avoir en cours de transmission. Il s'agit du nombre d'octets pouvant être transmis sans provoquer la congestion du réseau.

– Slow Start Threshold (ssthresh) : c'est une valeur seuil permettant de choisir le bon algorithme de congestion selon les événements du réseau.

Les algorithmes utilisés sont le Slow Start Algorithm, le Fast Retransmit et le Fast Recovery.

III.3.1 Slow Start

Le mode slow start correspond au mode de démarrage lent. Le déclenchement de transmission dans un réseau, exige au protocole SCTP d'explorer le réseau afin de déterminer la capacité disponible. L'algorithme Slow Start est utilisé, pour cette raison, au lancement du transfert ou après la correction de pertes détectées par le temporisateur de retransmission. Une association SCTP, étant en mode slow start, doit respecter les règles suivantes :

1- Initialement avant la transmission de données ou après une période de silence (Idle) suffisamment importante on doit avoir : cwnd = 2*MTU, (Maximum Transmission Unit)

2- Le cwnd initial, après l'expiration du temps de retransmission ne doit pas être supérieure à 1*MTU

3- La valeur initial de ssthresh peut être arbitrairement élevée (par exemple, les implémentations peuvent utiliser la taille de fenêtre indiquer par le récepteur)

4- Chaque fois que cwnd > 0, le nœud SCTP est autorisé d'avoir cwnd octets de données en attente d'acquittement sur cette adresse de transport.

5- Si cwnd ≤ ssthresh, un nœud SCTP doit utiliser l'algorithme Slow Start pour augmenter cwnd (supposant que la fenêtre de congestion actuelle est entièrement utilisée). Si un chunk SACK entrant incrémente le Cumulative TSN Ack Point, cwnd doit être incrémentée par au plus le minimum de :

a- La taille totale des blocs de données précédemment acquittés

b- Le path MTU de destination

Dans le cas où le point terminal de réception est multihomed, si l'émetteur reçoit un chunk SACK qui incrémente son Cumulative TSN Ack Point, alors il devrait mettre à jour sa cwnd (ou cwnds) répartie(s) sur les adresses destination vers lesquelles il a transmis les données acquittées. Cependant,

si le chunk SACK reçu n'incrémente pas le Cumulative TSN Ack point, le point terminal d'émission ne doit pas ajuster la cwnd.

Quand le nœud SCTP ne transmet pas de données sur une adresse de transport donnée (qui correspond à l'adresse primaire et actuellement utilisée comme destination), le cwnd de cette adresse doit être ajusté au max (cwnd/2, 2*MTU) par RTO.

En conclusion, le slow start démarre avec une taille de la fenêtre de congestion égale à deux paquets SCTP. La fenêtre de congestion augmente ensuite au fur et à mesure de la réception des acquittements. Au démarrage, l'anticipation se fait sur deux paquets **[20**], à chaque acquittement le nombre courant de paquets anticipés est augmenté d'un paquet SCTP, la croissance de la fenêtre d'anticipation sera donc exponentielle. L'objectif de base de cet algorithme est que la vitesse d'émission est égale à la vitesse de réception qui est reflétée par les acquittements. Par contre, après une première absence d'acquittement détectée par un acquittement dupliqué, le mécanisme dit d'évitement de congestion (congestion avoidance) se déclenche.

III.3.2 Congestion Avoidance

Le mécanisme d'évitement de congestion correspond au régime permanent d'une association SCTP. Au cours de cette phase la taille de la fenêtre de congestion est incrémentée de façon linéaire au delà du seuil de détection de congestion (ssthreshold). Le seuil de détection de congestion est déterminé à la détection de perte d'un segment et correspond à la moitié de la taille de la fenêtre de congestion [**26**].

L'évolution algorithmique du mécanisme de congestion avoidance se résume dans les étapes suivantes :

– Si cwnd > ssthresh, cwnd ◄– cwnd +1*MTU par RTT, si l'émetteur a "cwnd" ou plus d'octets dedonnées en attente d'acquittement pour l'adresse de transport correspondante.

En pratique une implémentation peut assurer cette évolution de la fenêtre de congestion de la façon suivante :

1- partial-bytes-acked =0,

2- Chaque fois que cwnd > ssthresh, à chaque réception d'un chunk SACK on incrémente Cumulative TSN Ack Point. Le partial-bytes-ack est incrémenté par le nombre total d'octets de tous les nouveaux blocs acquittés par le chunk SACK reçu incluant les blocs acquittés par le nouveau Cumulative TSN Ack et par les blocs Gap-Ack,

3- Si partial-bytes-ack cwnd, et qu'avant l'arrivée du chunk SACK l'émetteur a cwnd ou plus d'octets de données en attente d'acquittement5, on augmente cwnd d'un MTU et on remet partial-bytes-acked à pba= p a r t i a l.bytes.acked−cwnd [**20**],

4- Comme en Slow Start, quand l'émetteur ne transmet pas de données sur une adresse de transport donnée, le cwnd de cette adresse de transport doit être ajusté au max (cwnd/2, 2*MTU) par RTO.

5- Quand toutes les données transmises par l'émetteur ont été acquittées par le récepteur, partial-bytes-acked est ré-initialisé à 0.

III.4 Multihoming

Le multihoming est une propriété essentielle de SCTP. Il permet à une association SCTP d'être associée à plusieurs adresses sources et destination. Chaque terminal peut ainsi être atteint via plusieurs adresses IP. SCTP apporte uniquement un mécanisme de sauvegarde de chemin [**20**]. Un seul chemin est actif à la fois ; le partage de charge ne fait pas partie de la spécification actuelle du protocole [**20, 22**]. Chaque nœud peut être accessible par plusieurs adresses de transport (Fig 3.19) fixées au moment de l'ouverture de l'association. Les adresses de transports (c'est-à-dire, liste d'adresses IP + port SCTP) sont échangées lors de la phase d'initialisation d'une association SCTP (chunks INIT et INIT-Ack).

Les transmissions vers des nœuds multihomed peuvent ainsi gagner en robustesse contre les défaillances du réseau ou les problèmes de congestion en choisissant dynamiquement une alternative, à la condition que des chemins différents soient constitués pour chaque adresse IP du nœud. SCTP voit les adresses IP d'un client multihomed comme «différents chemins» possibles pour l'atteindre [**27**].

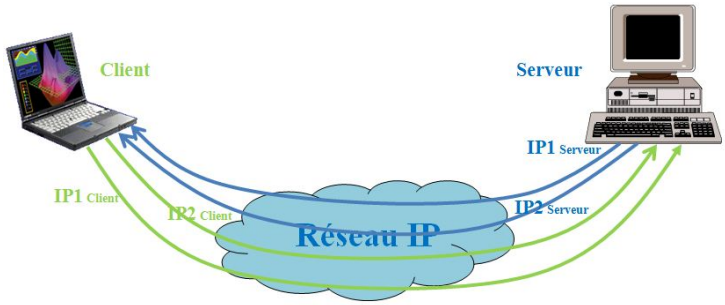

Fig. 3.19 – Exemple de nœuds SCTP MultiHomed.

III.4.1 Gestion des adresses IP

Les protocoles des couches supérieures (ou ULP Upper Layer Protocol) déterminent toutes les adresses transport et fixent le chemin primaire (chemin par lequel transite le trafic utilisateur en condition normale). Pour effectuer une transmission vers un nœud mulihomed, l'émetteur choisit préalablement une des adresses possibles qui correspondra au chemin primaire (lors de l'établissement de l'association). L'émetteur ne doit par la suite envoyer des données que par ce chemin primaire. De plus, pour acquitter des paquets SCTP d'un nœud multihomed, l'émetteur des chunks SACK devrait utiliser le même chemin que celui emprunté par les chunks de données reçus **[20]**. Cependant, lorsque

le récepteur obtient plusieurs chunks dupliqués, il peut supposer que ses chunks SACK empruntent un chemin défaillant, il serait alors peut être judicieux d'utiliser une autre adresse secondaire.

Lors de retransmissions il serait également intéressant d'utiliser une autre adresse secondaire possible, la perte de paquets pouvant être liée à un problème de l'adresse primaire, et permet ainsi de réduire les risques de congestion. Lorsque le chemin primaire devient inaccessible (suite à une congestion, ou à une perte de données importante), SCTP basculera tout le trafic sur un des chemins secondaires de l'association considérée (vers une adresse de transport secondaire du nœud de destination). La fonction de multihoming est uniquement utilisée à des fins de sauvegarde.

III.4.2 Contrôle des adresses empruntées (ou des chemins)

SCTP se doit de contrôler régulièrement les différentes adresses d'un nœud multihomed. Pour cela il considère deux états possibles pour chacun des chemins possibles : actif ou inactif. Le chemin primaire étant considéré actif, la disponibilité des alternatives est contrôlée par l'envoi régulier de chunks HEARTBEAT Request, qui doivent être acquittés par un chunk HEARTBEAT ACK. Si une adresse ne répond pas après plusieurs HEARTBEAT infructueux, le chemin est considéré inactif.

La décision sur le fait qu'une adresse est accessible ou non est prise selon un mécanisme d'écoute appelé heartbeating. Un chemin primaire est supposé inactif, si l'émetteur ne peut plus y accéder suite à l'occurrence de plusieurs expirations de timer RTO (Retransmission Time Out) consécutives. Dans ce cas les paquets SCTP seront routés vers une autre adresse supposée active. Le choix de l'adresse IP secondaire de destination à activer se fait en contactant le nœud destination via un chunk Heartbeat Request (Fig 3.20). Le nœud SCTP distant doit répondre avec un chunk Heartbeat Ack (Fig 3.21) en indiquant l'adresse IP secondaire qui sera active dans l'association en cours.

Chunk type = 4	Flag s	Heartbeat length
Heartbeat info type = 1		Heartbeat info length
Sender specific Heartbeat info		

Fig. 3.20 – Format du Chunk Hearbeat Request.

Chunk type = 5	Flags	Heartbeat-Ack length
Heartbeat-Ack info type = 1		Heartbear-Ack info length
Sender specific heartbeat info		

Fig. 3.21 - Format du Chunk Hearbeat-Ack.

Un chunk heartbeat est envoyé périodiquement vers les adresses de transport de destination en veille [**22**] (peut être active ou inactive) afin de mettre à jour leurs états d'accessibilité. La période est

donnée par Hi. La période d'émission d'un chunk heartbeat est contrôlée par le paramètre HB. Interval (30 seconde) [**20**].

Hi=RTOi + HB.Interval (1+δ) (1)

Où, RTOi est le RTO sur le chemin secondaire «i» calculé sur les chunks heartbeat, et d est une valeur choisie aléatoirement dans [-0.5,0.5] à l'initialisation de l'association.

Lors de transfert de données au sein d'une association multihomed des considérations doivent être prises en compte :

– Quand l'émetteur est multihomed, le récepteur ne doit pas nécessairement envoyer un SACK vers l'adresse de transport primaire de l'émetteur.

– Lorsque le récepteur est multihomed et que l'émetteur a besoin de retransmettre un ou plusieurs chunks de données, l'émetteur peut considérer la retransmission vers une adresse secondaire de l'association [**20**].

Il faut noter qu'en cas d'erreur sur la liaison primaire, l'émetteur utilisera un des chemins secondaires pour émettre les données vers le récepteur. Pour une association un seul chemin est considéré comme primaire les autres sont des chemins de secours. Seulement des messages d'écoute (Heartbeat) circulent sur les chemins secondaires dits aussi alternatifs.

III.4.3 Transfert de données dans une association avec multihoming

Comme nous l'avons signalé ci-dessus, SCTP supporte la fonctionnalité de multihoming afin de permettre à des sessions, ou des associations dans la terminologie SCTP, de se maintenir même lorsqu'une adresse IP d'un hôte n'est plus accessible. SCTP a un système intégré de détection et de rétablissement d'erreurs, qui permet aux associations d'envoyer dynamiquement le trafic vers une adresse IP alternative de destination si cela s'avère nécessaire.

Comme indiqué dans la [20], l'utilisation du multihoming en SCTP ajoute au protocole les fonctions de base suivantes :

– Dans une association, un chemin unique est considéré primaire. Ceci signifie qu'une des adresses IP affectées au récepteur de l'association est choisie pour être l'adresse primaire. Le chemin primaire correspond au chemin réseau qui mène à l'adresse primaire du point terminal qui lui est associé. Sauf contre indication par l'utilisateur SCTP, un point terminal devrait toujours transmettre sur le chemin primaire

– Lors de l'acquittement des chunks reçus, les chunks SACK doivent emprunter le même chemin qui a été emprunté par les chunks reçus.

– Dans le cas de retransmission de chunk vers un point terminal multihomed, le récepteur doit choisir une adresse de destination autre que celle à laquelle le chunk de données original a été envoyé. Aussi bien, quand un point terminal reçoit un chunk de données dupliqué, il peut changer l'adresse de destination et ne pas utiliser l'adresse source de ce chunk de données dupliqué pour envoyer un

acquittement. En fait, il faut considérer toutes les alternatives d'adresses de transport (source/destination) pour sélectionner l'adresse source-destination la plus adéquate pour la retransmission. Il n'y a aucune stratégie adoptée pour le choix de l'adresse définie par la norme (e.g. une approche RR: Round Robin).

– Une autre spécificité de SCTP, est utilisée pour contribuer à l'exécution de multihoming : le mécanisme heartbeating. Ce mécanisme détecte les problèmes sur les chemins en veille et les points terminaux. Il peut donc détecter si une adresse destination est active ou inactive. Les heartbeat chunks sont envoyés périodiquement à toutes les destinations en veille (quelles soient actives ou inactives) [22], et un compteur (Ei) (Fig 3.22) calcule le nombre de chunks heartbeat envoyés vers une destination inactive sans le retour du chunk heartbeat-Ack.

Quand ce compteur excède une valeur maximale (Path Maximum Retransmission), cette adresse de destination est déclarée comme inactive. De même en absence d'une réponse après un certain temps (RTO : T3-rtx expire) l'émetteur peut considérer que l'adresse de transport de destination en cours comme inaccessible, et incrémente le compteur E (error counter) de cette adresse IP.

Ainsi, si le besoin s'impose une autre adresse IP, différente de l'adresse IP primaire, est utilisée afin d'atteindre un point terminal multihomed, SCTP connaît laquelle des adresses est active et peut ainsi éviter d'utiliser un autre chemin défectueux.

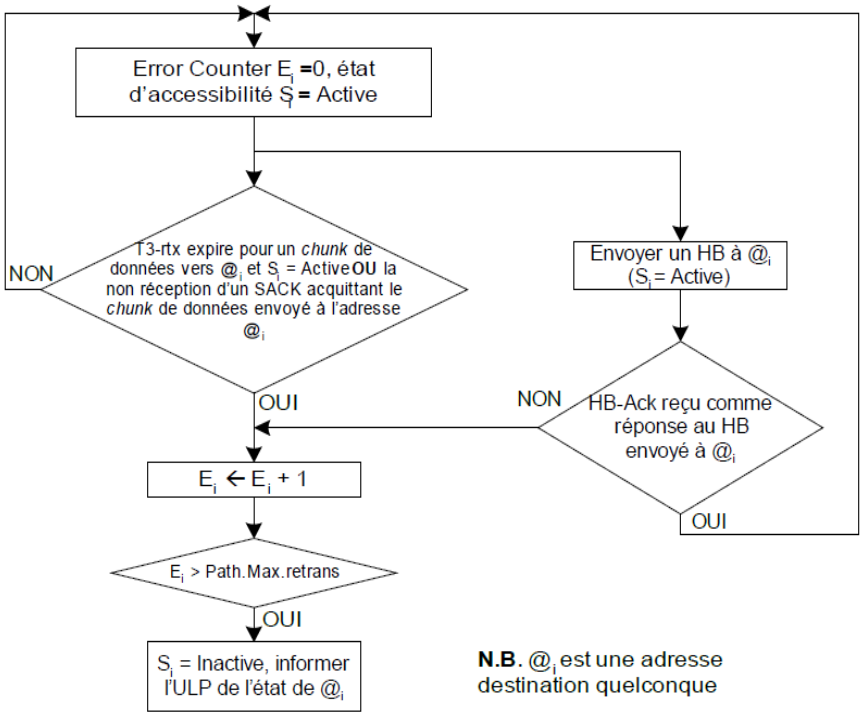

Fig. 3.22 - Procédure de transfert de données relativement à un nœud SCTP MultiHomed.

Ainsi le mécanisme multihoming, supporté par des machines et des équipements réseau, est une solution techniquement faisable et de plus en plus économique [**28**]. Un hôte est multihomed s'il peut être adressé par des adresses IP multiples, comme c'est le cas quand l'hôte a plusieurs interfaces réseau. Par conséquent, des équipements radio peuvent être connectés simultanément à plusieurs

technologies d'accès (Fig 3.23). Ainsi, les machines pourront avoir des connexions filaires et radio. Les différentes interfaces actives suggèrent également l'existence simultanée de plusieurs chemins entre les hôtes multi-adressés (multihomed).

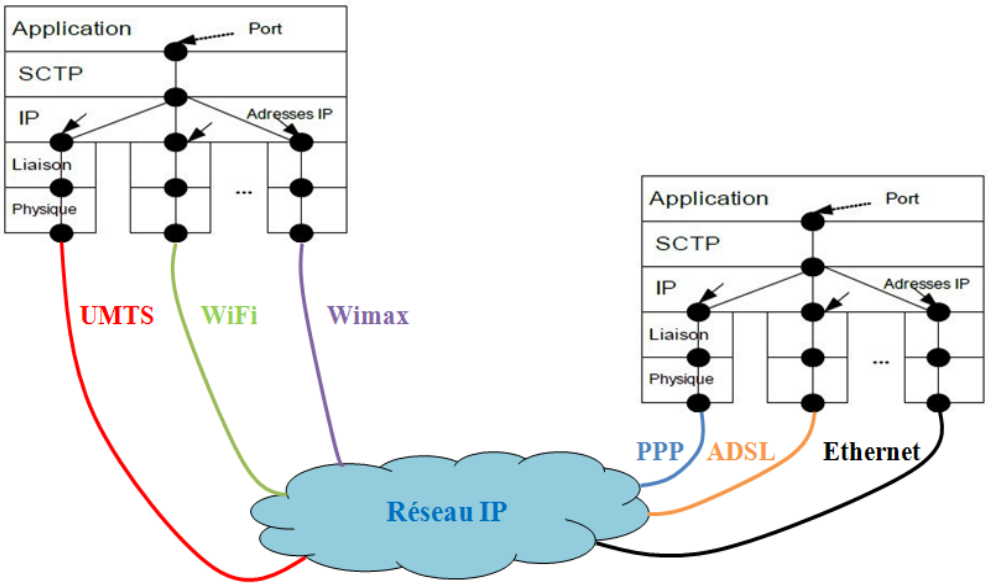

Fig. 3.23 – Exemple de nœud SCTP MultiHomed connecté à plusieurs technologies d'accès.

III.5 Multihoming et mobilité

Pour trouver une solution qui permet le déplacement inter-réseaux IP, sans interruption de l'association en cours, deux standards sont en cours de validation. Il s'agit des drafts IETF suivants : «Stream Control Transmission Protocol (SCTP) Dynamic Address ReconFiguration» **[29]**, et «Mobile SCTP» [30]. Ceux ci sont une extension à la fonctionnalité mobile IP disponible dans IPv4 [**31**] et dans IPv6 [29].

III.5.1 Extension de SCTP : Adressage dynamique

Cette fonctionnalité introduit une extension au SCTP [**29**] qui lui offre la possibilité de :

1) Reconfigurer des adresses IP au cours de l'association en cours

2) Changer la route primaire (i.e. Routage vers une nouvelle adresse primaire)

3) Échanger des informations d'adaptation de couche durant l'établissement d'association [29]. Ce qui assure un basculement de lien sans pertes de données.

Cette extension consiste à la création de deux nouveaux types de chunk : Address ConFiguration Change Chunk ASCONF (Fig 3.24) et Address ConFiguration Ackowledgement ASCONFACK (Fig 3.25). Ces deux chunks contribueront à l'ajout et la suppression dynamiques des adresses IP à une association ainsi qu'à la génération d'une demande de changement d'adresse primaire au cours de la même association.

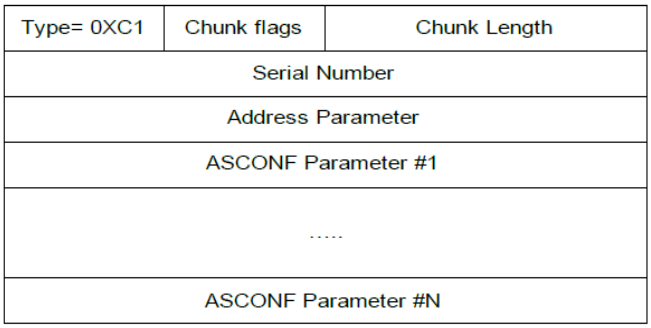

Fig. 3.24 – Format du Chunk ASCONF.

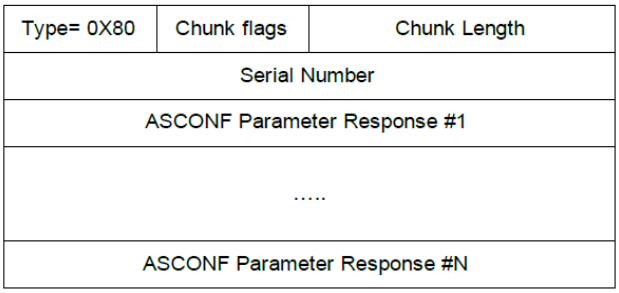

Fig. 3.25 – Format du Chunk ASCONF-ACK.

Le chunk ASCONF est utilisé pour communiquer au point terminal distant une des demandes de changement de configuration. Ces demandes sont spécifiées par l'un des nouveaux paramètres introduits par cette extension. Ces demandes doivent être acquittées au moyen du chunk ASCONFAck comprenant les paramètres spécifiques pour assurer la réponse à une demande reçue (échec ou succès de l'opération). Ces chunks sont transmis de façon authentifiée.

De nouveaux paramètres sont introduits pour décrire la nature de l'opération exigée par l'un des deux points terminaux d'une association afin de changer sa configuration. Les deux types de chunks associés aux six paramètres nouvellement introduits par cette extension servent pour ajouter et supprimer dynamiquement des adresses IP d'une association et donc de changer d'adresse primaire (changer la configuration d'une association). Les six nouveaux paramètres introduits par cette extension sont :

– Add IP Address (ASCONF) : permet d'ajouter une nouvelle adresse IP à l'association en cours.

– Delete IP Address (ASCONF) : permet de supprimer une adresse IP de l'association en cours.

– Error Cause Indication (ASCONF-ACK) : c'est un paramètre de réponse, qu'est utilisé pour contourner une ou plusieurs causes d'erreur usuelles rencontrées en SCTP.

– Set primary IP Address : ce paramètre peut être intégré dans les chunks ASCONF, INIT ou INITAck. L'intégration de ce paramètre dans INIT ou INIT-Ack peut être utilisée pour indiquer une préférence d'une adresse primaire.

– Success Indication (ASCONF-ACK) : ce paramètre est utilisé pour indiquer le succès de réception des données contenues dans le chunk ASCONF.

– Adaptation Layer Indication : ce paramètre peut apparaître dans les chunks INIT et INIT Ack et devrait être remonté aux protocoles de couche supérieure du récepteur. Ce paramètre ne doit pas apparaître dans le chunk ASCONF. Il est envisagé qu'il soit utilisé pour le contrôle de flux et pour l'adaptation à d'autres couches qui exigent une indication qui soit contenue dans les chunks INIT et INIT-Ack.

Le format des paramètres est donné par la Fig. 3.26 :

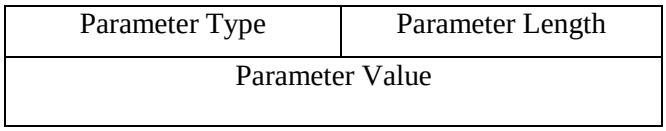

Fig. 3.26 – Format des paramètres.

Parameter type: sur 16 bits décrie le type du paramètre chunk le tableau 3.2 résume les différents type de paramètre.

Paramètre	Valeur des Types de paramètres
Add IP Address	0xC001
Delete IP Address	0xC002
Error Cause Indication	0xC003
Set primary IP Address	0xC004
Success Indication	0xC005
Adaptation Layer Indication	0xC006

Tableau 3.2-type de paramètre ASCONF.

Parameter Length: sur 16bits détermine la taille en octet du paramètre chunk y compris le type et le champ valeur.

Parameter Value: sur 32 bits est la valeur du paramètre, contient l'information utile à transférées dans le paramètre chunk.

III.5.2 Procédure ASCONF de SCTP

Une des conditions nécessaire de transfert du chunk ASCONF est que ces transferts ne doivent pas causer de congestion dans le réseau. Quand un point terminal veut transmettre un ASCONF signaled change à un point terminal SCTP distant il doit procéder comme suit :

1) Créer un chunk ASCONF. Ce chunk devrait contenir toutes les informations TLV (Type Length Value) nécessaires à être émises au point terminal SCTP distant, et une relation d'identités (Correlation Ids) unique relativement à chaque demande.

2) Un serial number doit être attribué au chunk. Ce paramètre doit être un numéro croissant. Le serial Number doit être initialisé, à l'établissement de l'association. Il a la même valeur que l'Initial TSN et chaque fois qu'un nouveau chunk ASCONF est créé il est incrémenté de 1 après avoir attribué le Serial Number au nouveau chunk créé.

3) Si aucun chunk ASCONF n'est en cours de transmission (non acquitté) au point terminal SCTP associé, alors un chunk ASCONF lui est transmis.

4) Déclencher le temporisateur T-4 RTO, en utilisant la valeur RTO de l'adresse destination sélectionnée (normalement c'est le primaire).

5) Quand un chunk ASCONF-Ack, acquittant le Serial Number précédemment envoyé, est reçu, le temporisateur T-4 RTO est arrêté et l'association relative est réinitialisée ainsi que les compteurs des erreurs de destination.

6) Traiter tous les TLVs (les données) contenues le chunk ASCONF-Ack afin de trouver des informations sur des états particuliers en réponse aux différentes demandes qui ont été envoyées. Utilisation des Correlation IDs afin de lier la demande aux réponses correspondantes.

7) Si une réponse d'erreur est reçue pour un paramètre TLV, alors tous les TLVs sans réponse sont considérées comme réussies s'ils ne sont pas remontées. Tous les TLVs après l'échec de réponse sont considérées comme non réussies à moins qu'une indication spécifique de succès soit présente pour le paramètre.

8) S'il n'y a pas de réponse(s) à des paramètres TLVs spécifiques, et que pas d'échecs indiqués, alors toutes les demandes sont considérées comme réussies.

9) Si le récepteur SCTP répond à un chunk ASCONF avec un chunk d'erreur rapportant qu'il n'a pas reconnu le type de chunk ASCONF. L'émetteur du chunk ASCONF ne doit pas envoyer des chunks ASCONF supplémentaires et doit arrêter son temporisateur T-4.

Si le temporisateur T-4 RTO expire, le point terminal SCTP devrait procéder selon les étapes suivantes [29] :

1) Incrémenter les compteurs d'erreurs et exécuter la détection d'échec de route concernant l'adresse destination appropriée.

2) Incrémenter les compteurs d'erreurs au cours de l'association et exécuter l'échec de détection au niveau du point terminal dans l'association.

3) Modifier la valeur RTO de l'adresse destination vers laquelle le chunk ASCONF a été envoyé en doublant la valeur du temporisateur RTO.

4) Retransmettre le chunk ASCONF précédemment transmis, si c'est possible, sur une adresse destination alternative. Un point terminal SCTP ne doit pas ajouter de nouveaux paramètres à ce chunk, il doit conserver le même chunk ASCONF précédemment transmis.

5) Réinitialiser le temporisateur T-4 RTO.

Notons que si une adresse destination différente est sélectionnée, alors le RTO utilisé sera celui de la nouvelle adresse destination.

III.5.3 Règles générales de gestion des adresses

Suite à une demande d'ajout d'une nouvelle adresse IP à une association en cours, cette adresse IP n'est pas totalement exploitée avant que l'ajout ne soit acquitté. Dans ce cas l'émetteur ne doit pas utiliser la nouvelle adresse IP comme une source pour n'importe quel paquet SCTP, sauf celui comportant le chunk ASCONF. Alors que le récepteur, de la demande d'ajout d'adresse IP, peut utiliser immédiatement cette adresse comme destination. Après la réception d'un chunk ASCONFAck d'un IP-address add, le point terminal SCTP peut commencer à utiliser l'adresse IP ajoutée comme adresse source pour n'importe quel type de chunk SCTP. D'autre part, la suppression d'une adresse IP d'une association, cette adresse IP doit être considérée comme adresse destination valide pour la réception de paquets SCTP jusqu'à l'arrivée de ASCONF-Ack. Alors qu'elle ne doit pas être utilisée comme adresse source pour aucun des paquets envoyés ultérieurement. C'est-à-dire qu'aucun des datagrammes qui arrivent avant ASCONF-Ack, destinés à l'adresse IP à supprimer, n'est ignoré par le récepteur. Tandis que les chunks ABORT arrivant à destination de l'adresse IP à supprimer doivent être ignorés.

III.5.4 Mobile SCTP (mSCTP)

Le draft IETF «Mobile SCTP» [30], décrit le Mobile SCTP comme étant une extension du protocole de transport SCTP par l'introduction de l'aspect d'adresse IP dynamique pour le multihoming ce qui favorise la mobilité.

La fonctionnalité principale permettant de maintenir une association active lors de changement de réseau IP est la fonction ADDIP [29]. Cette dernière permet de modifier/supprimer/ajouter une adresse IP faisant partie de l'association en cours, sans provoquer son interruption.

Mobile SCTP (mSCTP) est conçu pour une architecture de type client/serveur, avec un client mobile qui initie l'association avec le serveur fixe. mSCTP doit être utilisé avec des algorithmes de gestion de mobilité tels que Mobile IP ou Dynamic DNS.

Nous décrivons dans la suite mSCTP en proposant un exemple d'étude. Nous considérons, alors, un client mobile dans une zone comprenant deux accès radio distincts (Fig 3.27). Un client mobile (c_mobile) se connecte à Internet en passant par certaines technologies radio. Il lui sera attribué une adresse IP à partir de l'espace d'adressage de la localisation A par exemple (IPc_mobile_LocA). Ceci est accomplit par une des techniques d'attribution dynamique d'adresse comme DHCP (Dynamic Host ConFiguration Protocol). Le client mobile se déplace de la zone de localisation A vers la zone de localisation B et il est informé qu'il a atteint la zone de couverture d'un nouveau réseau à partir des informations fournies par la couche physique de son NIC (Network Interface Card).

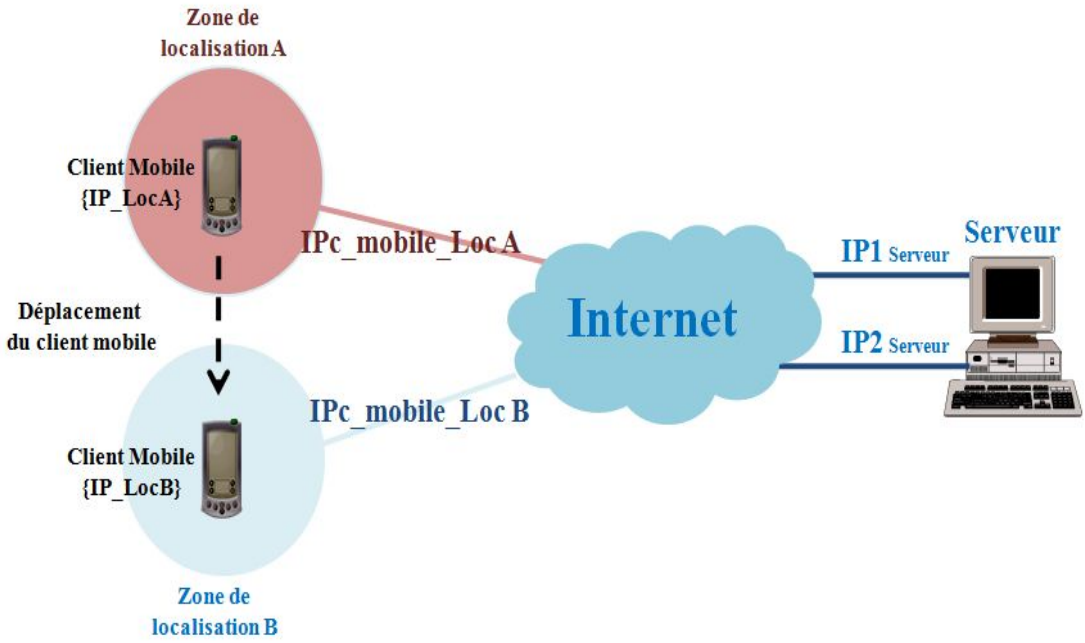

Fig. 3.27 – Mobile SCTP.

En plus de son ancien lien (avec la zone de localisation A), le client mobile établit un nouveau lien avec le réseau de la zone de localisation B et une nouvelle adresse IP (Ipc_mobile_LocB) lui est affectée sur sa seconde interface. Le client mobile est donc multihomed et il est accessible par deux réseaux différents. Dans ce cas le mobile ajoute la nouvelle adresse IP, qui lui a été attribué par identification de la connexion au serveur. Afin d'avoir une distinction facile des deux liens au niveau du client mobile, plusieurs adresses IP doivent être affectées aux interfaces réseau du serveur, ce qui permet de représenter différents liens par des entrées différentes de la table de routage du client mobile.

En accédant à la zone de localisation B, le client mobile peut quitter la zone de couverture du point d'accès dont la zone de localisation est A et peut perdre le lien correspondant à sa première adresse IP. Le flux de données entre serveur et mobile est interrompu et le comportement fiable du protocole de transport assure que toutes les données sont envoyées sur le second lien dans le cas d'échec permanent sur le premier lien. Dans ce cas le client mobile informe son destinataire qu'il n'est plus accessible par la première adresse IP et demande de supprimer cette adresse de l'association. Si le client mobile a accès aux informations sur la puissance du signal radio, le handover vers le second lien sera initié avant que la perte de paquets se produise au niveau serveur.

En résumé, avec mSCTP, un client mobile peut avoir au moins deux adresses IP au cours de l'association existante, et durant le mode handover si deux points d'accès sont simultanément disponibles. En effet, un client mobile se trouvant dans la zone de localisation A, obtient une seule adresse IPc_mobile_LocA (l'association SCTP établie à une seule adresse). En se déplaçant vers la zone de localisation B, dans la zone d'intersection des deux zones de couverture (ou localisation A et B), le client mobile obtient une nouvelle adresse IPc_mobile_LocB, et ceci au moyen du mécanisme

DHCP par exemple. L'ajout d'une nouvelle adresse IP se fait donc au moyen de DHCP par l'envoie d'un chunk ADD-IP (Fig 3.28). Après la réception d'un chunk ADDIP-Ack à partir de son point terminal SCTP qui lui est associé, la nouvelle adresse est utilisée selon le mécanisme de gestion de route (path management mechanism) [32] qui est responsable de la détection d'adresses IP non disponibles et le basculement sur les adresses secondaires.

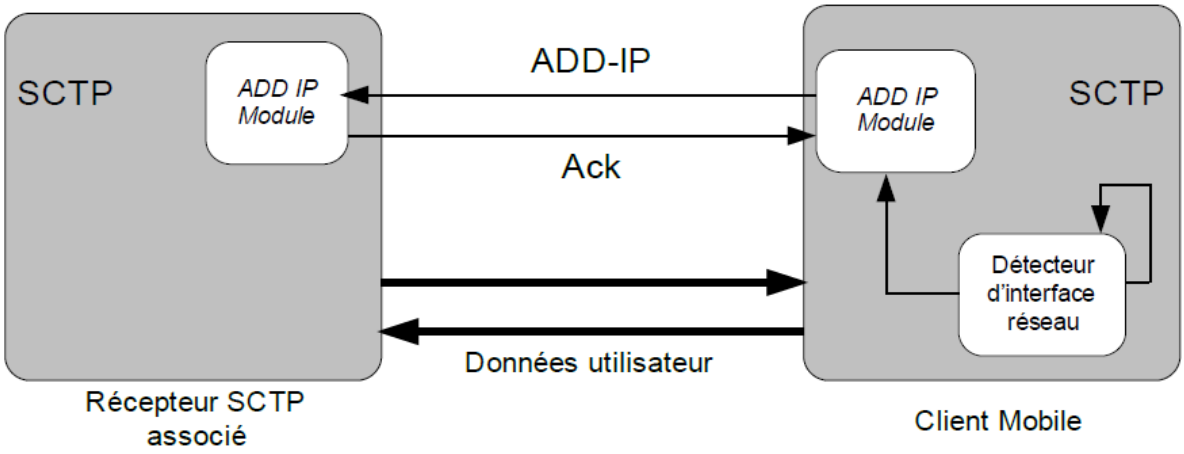

Fig. 3.28 – Un Prototype Mobile SCTP.

III.5.5 Combinaison de la mobilité au niveau couche liaison avec celle au niveau transport

Certaines technologies d'accès radio comme IEEE802.11 fournissent des fonctionnalités de gestion de mobilité au niveau couche liaison. Un Handover couche 2 est la plupart du temps restreint à la micro mobilité mais peut être avantageux en le combinant avec la gestion de mobilité niveau transport.

Si le mobile SCTP est utilisé dans un environnement IEEE802.11, un client mobile doit prendre 4 décisions [30] :

1) En fonction des paramètres physiques le client mobile doit décider quand est ce qu'il s'associe à une station de base. Pour prendre cette décision il peut prendre en compte le niveau du signal.

2) Après l'établissement du lien radio couche 2 une méthode de configuration IP doit être activée (si le lien radio est stable).

3) Après avoir terminé la conFiguration du nouveau lien, une décision doit être prise lorsque cette nouvelle route est remontée au nœud destinataire associé en utilisant le mécanisme ADDIP.

4) La dernière décision concerne l'instant où le récepteur doit demander le changement de route primaire.

III.5.6 Insuffisances de mSCTP

SCTP, avec l'extension ADDIP (mSCTP), offre une solution pour la gestion de handover et de localisation dans les réseaux de radiocommunication. mSCTP a fait l'objet de plusieurs travaux [32, 33, 34]. En effet, mSCTP peut être utilisé pour permettre un handover sans pertes lors des sessions mobiles qui sont déclenchées par le serveur vers le client mobile et inversement. La principale hypothèse d'avoir un handover sans pertes est qu'un MN (Mobile Node) est capable d'obtenir une nouvelle adresse IP à partir d'une nouvelle localisation. Ceci est implémenté en utilisant DHCP pour les réseaux IPv4 ou d'autres mécanismes similaires [30]. Malgré ses avantages, mSCTP présente des insuffisances.

D'une part, le mobile SCTP ne maintient pas un handover simultané pour les 2 points terminaux SCTP (formant l'association). Si les deux points terminaux exécutent un HO en même temps, l'association SCTP sera perdue. Autrement dit, l'activation de deux adresses primaires simultanées n'est pas faisable avec mSCTP. Une seule adresse primaire est active à la fois, ce qui peut entraîner un échec du handover.

D'autre part, mobile SCTP, tel qu'il est décrit par [30], présente des insuffisances de point de vue gestion des liens radio lors du basculement. En effet, en mSCTP les paquets de données sont émis vers l'ancienne adresse IP avant que le MN considère la nouvelle adresse IP comme adresse primaire pour l'association. Ce qui engendre des pertes de paquets lors de chaque handover du côté du Correspondent Node avant que celui ci décide finalement de changer de route primaire. De plus, mSCTP ne permet pas de gérer pratiquement la mobilité c'est qu'il ne décrit pas comment changer la route primaire lors des handovers. Le mSCTP actuel ne supporte pas le roaming, c'est que s'il n'y a pas d'entité pour la gestion de localisation, le chunk INIT pour une nouvelle association avec le MN's home address ne peut pas être correctement routé vers la nouvelle localisation du MN après qu'il se déplace vers un nouveau réseau. Pour remédier à ces insuffisances, les auteurs dans [34] proposent un nouveau protocole basé sur le mobile SCTP, il s'agit du cellular SCTP (cSCTP).

III.5.7 Cellular SCTP : cSCTP

Cette alternative du mobile SCTP, consiste en une modification des chunks ASCONF et ASCONF-Ack afin d'informer le Correspondant Node (CN) du déclenchement d'un mode handoff au niveau du MN (Mobile Node). Cette modification consiste à l'utilisation d'un bit H des chunks flags pour indiquer le déclenchement du mode Handover (Fig 3.29). En fait, le CN à la réception d'un Add-IP, ajoute la nouvelle adresse à l'association. Si de plus le bit H=1 (mode handover), alors les deux adresses (l'ancienne adresse primaire et la nouvelle adresse ajoutée) seront considérées comme primaires pour le MN (en même temps). Dans ce cas le CN envoie des paquets dupliqués sur les deux adresses primaires vers le MN. Et la valeur de cwnd pour chacune des deux adresses est égale à la

moitié de celle de l'ancienne adresse primaire. Contrairement au mSCTP où les paquets de données sont émis vers l'ancienne adresse IP avant que le MN considère la nouvelle adresse IP comme une adresse primaire pour l'association en cours. La suppression de l'ancienne adresse IP de l'association se fait lorsque cSCTP du MN décide qu'une adresse est inactive. Le MN quitte le mode HO (handoff_mode = false), supprime l'ancienne adresse IP de l'association et envoie au CN un chunk DELETE-IP ASCONF. D'où la procédure de handover proposée par le cSCTP fonctionne de la manière suivante :

1- Détection et obtention d'une nouvelle adresse IP : le HA (Host Agent) envoie des messages ROUTER SOLICITATION vers les ARs (Access Router) et ceux ci répondent par des messages ROUTER ADVERTISEMENT au moyen du DHCP ou Stateless Address Auto ConFiguration.

2- Ajout d'une nouvelle adresse dans l'association. Après l'obtention d'une nouvelle adresse IP au niveau du point d'attachement, le Host Agent informe le composant cSCTP du nœud mobile de sa nouvelle adresse IP.

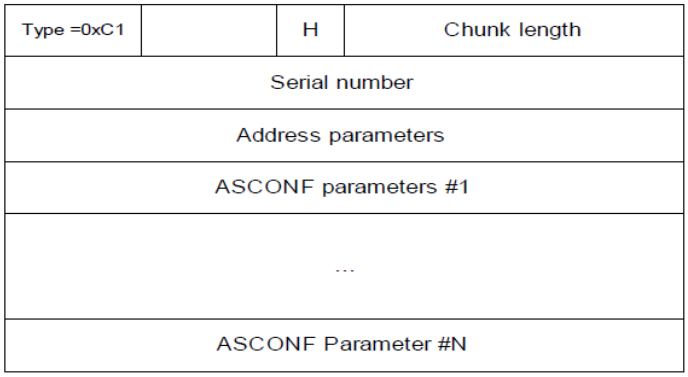

Fig. 3.29 – Modification du Chunk ASCONF pour activer le mode handover.

III.5.8 Mécanisme de gestion de mobilité au niveau transport (basé sur mSCTP)

Une autre alternative de mSCTP a été proposée dans [36]. Elle consiste à proposer un mécanisme qui détermine les conditions d'ajout et de suppression des adresses IP, en exploitant la puissance du signal radio au niveau couche liaison dans le but d'améliorer les performances du mSCTP.

Il s'agit d'une amélioration du protocole mSCTP. Étant donné que mobile SCTP, tel qu'il est décrit par [**30**], ne permet pas de gérer pratiquement la mobilité. De plus, mSCTP ne décrit pas comment changer la route primaire lors des handovers. Ainsi, si le changement de route primaire est appliqué tel qu'il est décrit par mSCTP, le CN (Correspondant Node) subira plusieurs pertes de paquets lors de chaque handover avant qu'il décide finalement de changer de route primaire.

Cette amélioration est basée sur des mesures de qualité de signal au niveau couche liaison. En effet, mSCTP au niveau MN (Mobile Node) exécutera ADDIP chaque fois que le niveau de puissance du signal reçu à partir du nouveau Access Router dépasse la valeur seuil du niveau de signal autorisant des communications. Cette fonctionnalité liée à la gestion des adresses IP (ajout/suppression) est

implémentée dans un bloc logique appelé AMM (Address Management Module) (Fig 3.30). Le AMM détermine quand est ce qu'il déclenche ADDIP et DELETE IP, ainsi que le changement de la route primaire et informe le mSCTP de l'action à exécuter en fonction des signaux reçus à partir de la couche liaison et du module IP Address Acquisition Module. L'AMM maintient des informations telles que l'interface correspondante à la route primaire courante et l'interface offrant la puissance maximale du signal au niveau couche liaison.

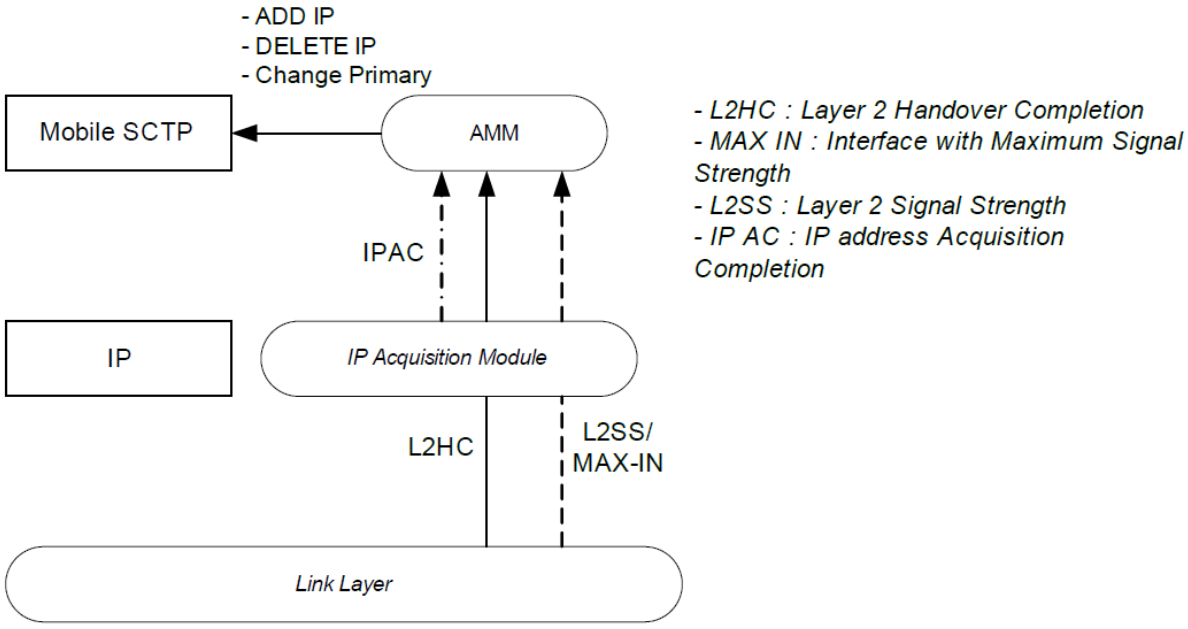

Fig. 3.30 – Amélioration proposée du Mobile SCTP.

III. 6. Proposition d'Extension avec le support d'équilibrage de charge basé sur le type de flux

Le multihoming pourrait améliorer les procédures de gestion de ressources et résoudre la problématique de congestion dans les réseaux sans fils. Notre contribution est d'activer deux à trois adresses primaire en même temps, chacune correspondant à une interface avec un réseau d'accès (UMTS, WIFI et WIMAX). L'idée consisterait à envoyer les données volumineuses demandant une bande passante importante sur le réseau présentant plus de ressources disponibles (Débit).

Dans ce cadre, notre contribution consiste à l'extension de mSCTP avec des mécanismes d'équilibrage de charge en nous basant sur la nature des accès multiples disponibles pour le nœud mobile et le type de flux transmis par le nœud correspondant. A cet effet, une extension au message ASCONF a été faite pour supporter l'aiguillage des flots selon la politique décrite ci-dessous :

	Différents technologies d'accès disponible					
Différents Flux	**WIFI**	**UMTS**	**WIFI & UMTS**	**WIFI & WIMAX**	**UMTS & WIMAX**	**UMTS & WIFI & WIMAX**
Audio	**WIFI**	**UMTS**	**WIFI**	**WIMAX**	**WIMAX**	**WIFI**
Vidéo	**WIFI**	**UMTS**	**WIFI**	**WIMAX**	**WIMAX**	**WIMAX**
Data	**WIFI**	**UMTS**	**UMTS**	**WIFI**	**UMTS**	**UMTS**
	Interface d'écoulement pour chaque flux					

Tableau. 3.3-Politique de routage.

Cette politique de routage doit être implémentée au niveau du MN.

III. 6.1. Définition des types de flux

Nous avons défini dans ce travail 3 classes pour le type de flux, correspondant respectivement aux flux de données, audio et vidéo. Cette classification peut être étendue à d'autres types de flux. Le CN consulte le champ Protocol_ID pour classer les flux, les applications sont classifiées dans chaque flux selon le champ Protocol_ID figurant dans un Chunk SCTP (fig 3.4).

Type de flux	Paramètre associé	Protocol ID
Données	0	Http, Ftp, Telnet…
Audio	1	RTP, MGCP, RTSP…
Vidéo	2	RTSP, H323…

Tableau. 3.4 – Type de flux.

III.6.2. Définition des types d'interfaces d'accès

Nous nous sommes restreints à trois types de technologie d'accès : WiFi, WiMAX et UMTS, ce travail peut être étendu à d'autres technologies. Le MN peut déterminer ces paramètres à partir des informations obtenue de la NIC (Network Inteface Connection). Le tableau 3.5 doit être implémenté dans la mémoire cache du MN. Nous avons défini pour cela trois paramètres associés :

Type d'interface	Paramètre associé
UMTS	0
WiFi	1
Wimax	2

Tableau. 3.5 – Type d'interface.

III.6.3. Modification des paquets chunk asconf et chunk asconf-ack

Les modifications proposer au chunk asconf à était au niveau des asconf parameters, il existe 6 paramètres (voir III.5.1 Extension de SCTP : Adressage dynamique – Fig 3.24 et Fig 3.25 -), nous nous somme intéresser au paramètre « Set primary IP Address ». La Fig 3.31 présente le format du paramètre Set Primary IP address contenue dans le chunk asconf.

a- Le parametre Set Primary IP Address

Type =0xC004	Length = Variable
ASCONF-Request Correlation ID	
Address Parameter	

Fig. 3.31 – le parametre Set Primary IP Address.

ASCONF-Request Correlation ID :

Une valeur de 32bits attribué par la source pour identifier chaque paramètre de la requête, le destinataire du chunk ASCONF doit recopier la valeur est la mètre dans le ASCONF-Request Correlation ID dans le chunk ASCONF-ACK.

Address Parameter :

Une adresse IPv4 ou IPv6 est envoyé au récepteur pour spécifier l'adresse comme adresse principale pour l'envoi des données.

b- Le nouveau Parametre Set Primary IP Address For Eath Flow

Type =0xC010	Length = Variable
ASCONF-Request Correlation ID	
Address Parameter for flow	

Fig. 3.32 –le paramtre Set Primary IP Address for Eath flow.

Address Parameter :

Le format du paramètre reste le même avec une modification dans le champ Address Parameter. Ce nouveau paramètre est nommé Set Primary Address For Each Flow. Pour indiquer au récepteur tel adresses IP utilisé pour telle flux. Nous avons introduit deux valeurs, les adresses IP des types d'interfaces disponible pour le MN et les classifications des types de flux (Tableau 3.4) à écouler dans ces interfaces.

III.6.4. Mécanisme proposé

Cette modification permet au protocole SCTP de sélectionner une interface pour chaque type de flot ainsi profité de plusieurs associations pour écouler plusieurs types de flux dans un but d'équilibrage de charge dans le cas ou plusieurs technologies d'accès sois présentes.

Le MN envois le chunk ASCONF avec le paramètre Set Primary Address For Each Flow pour indiquer au CN les adresses IP des technologies d'accès obtenue à partir de la NIC (Network Inteface Connection) et les types des flux qui doivent être écouler dans chaque interface.

Le CN classe les différents types de flux à partir du champ Protocol_ID contenue dans le chunk SCTP (fig 3.5) et du tableau Type de flux (Tableau 3.4) contenue sur sa mémoire cache. Quand le CN reçoit le chunk asconf et prend connaissance des technologies d'accès disponible pour le MN à partir du paramètre Set Primary Address For Eath Flow décide d'apert cette dernière d'orienté telle trafic sur telle interface.

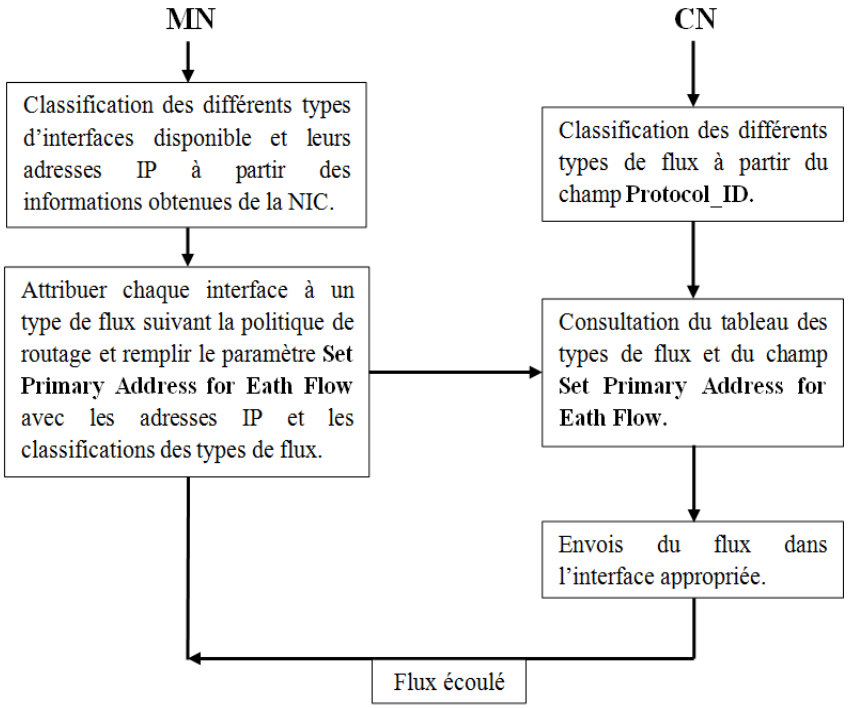

Fig. 3.33-Schéma de la contribution.

Par la suite nous allons voir les différentes métriques de la qualité de services et les besoins des applications en bonde passantes pour clarifier l'approche d'équilibrage de charge.

III.7 Les principales métriques liées à la QoS

La notion de qualité de service (QoS) a progressivement fait son apparition pour faire face aux différentes contraintes qu'exigeaient certains types d'applications, essentiellement dans le monde de l'interactif. En effet, à l'origine, la majorité du trafic Internet était constituée de données textuelles n'ayant pas d'exigences spécifiques fortes mais progressivement, des outils faisant intervenir simultanément du transfert de fichier, de la messagerie instantanée, de l'audio ou encore de la vidéo sont apparus. Dès lors, des garanties sur la bande passante, le délai ou encore la gigue devaient être fournies aux utilisateurs pour en assurer le bon fonctionnement.

Cependant, l'architecture de l'Internet, basée sur la pile TCP/IP, n'a pas été conçue dans le but de différencier les types de trafic et est actuellement dominée par un seul modèle de service : le best effort. Cette architecture ne peut garantir un fonctionnement correct de tous les types d'applications qu'en proposant le surdimensionnement du réseau qui consiste à le doter d'une capacité qui dépasse largement les besoins. Mais cette approche ne fait que repousser le problème et peut difficilement s'appliquer aux technologies d'accès sans fil limitées en bande passante.

Ainsi, une gestion efficace des ressources est nécessaire pour fournir aux utilisateurs de l'Internet des garanties de QoS adaptées à leur besoin. Dans cette partie, nous allons tout d'abord définir les principales métriques qui nous permettront d'évaluer la qualité de service offerte par un réseau puis

nous résumerons les exigences de QoS des principales catégories d'applications actuellement existantes.

Selon le standard ISO 8402 [37], la qualité de service se définit comme « l'ensemble des caractéristiques d'un service qui déterminent sa capacité à satisfaire des besoins formulés ou supposés ». La recommandation E800 de l'ITU-T [38] définit quant à elle la QoS comme « l'effet collectif de la performance d'un service qui détermine le degré de satisfaction d'un utilisateur de ce service ». Enfin, dans [39], l'IETF définit la qualité de service comme « un ensemble de service prérequis à remplir par le réseau lors du transport d'un flux ».

Différents types de QoS se dégagent alors de ces définitions et peuvent être séparés en trois catégories [40]:

La qualité de service **intrinsèque** qui est directement fournie par le réseau lui-même et décrite par des paramètres objectifs tels que par exemple le délai ou les pertes. C'est sur ce point que l'IETF se focalise essentiellement.

• La qualité de service **perçue** qui correspond à la qualité ressentie par l'utilisateur (aussi appelée QoE, Quality of Exeprience). Elle dépend fortement des performances du réseau mais est mesurée par une moyenne des opinions des utilisateurs. La méthode la plus utilisée est le MOS (Mean Opinion Score) dans laquelle un ensemble d'utilisateurs évaluent séparément la qualité ressentie d'une application entre 1 et 5, une moyenne de leur note étant ensuite réalisée. Le MOS est généralement utilisé pour la qualité audio ou vidéo d'une application mais la QoS perçue peut aussi concerner le temps de connexion, la sécurité perçue par l'utilisateur, la disponibilité du service, etc. De plus, il n'y a pas forcément correspondance entre QoS intrinsèque et QoS perçue, cette dernière étant très subjective. L'ETSI et l'ITU utilisent essentiellement le terme QoS en tant que QoS perçue et préfèrent le terme de performance réseau pour ce qui correspond à la partie technique.

• La qualité de service **évaluée** qui se réfère à la volonté d'un utilisateur de continuer à utiliser tel ou tel service. Cela dépend de la QoS perçue mais aussi du prix, du service d'assistance offert par le fournisseur ainsi que d'autres aspects commerciaux.

Les principaux paramètres qui permettent de décrire la QoS intrinsèque dans les réseaux

IP sont les suivants :

• Le **délai** de transfert des paquets, exprimé en millisecondes. Il est généralement mesuré de bout en bout mais peut l'être sur une portion du réseau.

• La **gigue** ou variation du délai de transfert des paquets, exprimée en millisecondes.

• Le **débit** d'informations, exprimé en bits par seconde (bit/s ou bps) ou en octets par seconde.

• Le **taux de perte** de paquet, défini comme le pourcentage de paquets perdus par rapport au nombre total de paquets émis.

III.7.1. Exigences de QoS pour les applications audio et vidéo

Les applications audio et vidéo font parties des applications les plus exigeantes en termes de QoS, surtout quand elles font intervenir une conversation entre plusieurs participants. Le tableau 3.6 présente les différentes recommandations de l'ITU-T [41] concernant les paramètres que doivent respecter ces applications pour fonctionner correctement.

On remarque effectivement que les applications de conversation audio et vidéo (vidéoconférence) sont plus exigeantes en termes de délai. Pour garantir un fonctionnement correct, le délai aller doit idéalement être inférieur à 150 ms ; cependant, elles peuvent fonctionner si ce dernier ne dépasse pas les 400 ms, seuil à partir duquel la dynamique de conversation commence à clairement se dégrader. Par contre, la gigue doit rester très faible (inférieur à 1ms) et dans le cas où elle devient trop importante un tampon de compensation de gigue doit être utilisé. De plus, l'oreille et l'œil humain peuvent tolérer des pertes d'informations, lorsqu'elles sont faibles mais l'utilisation de codecs de compression performants, utilisant des mécanismes de recouvrement d'erreur par exemple, permet bien souvent de les limiter.

Application	Degré de symétrie	Débit typiques	Délai aller	Gigue	Taux de perte de paquets
Conversation audio	Bidirectionnel	4-64 kbit/s	Idéal :<150 ms Limite :< 400 ms	< 1 ms	< 3 %
Messagerie Vocal	Unidirectionnel	4-128 kbit/s	Lecture :< 1 s Enregistrement : < 2s	<< 1 ms	< 1%
Streaming audio	Unidirectionnel	16-128 kbit/s	< 10 s	<< 1 ms	< 1 %
Vidéoconférence	Bidirectionnel	16-384 kbit/s	Idéal :< 150 ms Limite :< 400 ms	< 1 ms	1 %
Streaming vidéo	Unidirectionnel	16-384 kbit/s	< 10 s	< 1 ms	< 1 %

Tableau 3.6 – Recommandations G1010 de l'ITU-T pour les applications audio et vidéo.

En ce qui concerne les applications de streaming, on constate qu'elles sont moins exigeantes en termes de délai ; cependant, elles nécessitent aussi un taux de perte de paquets faible, inférieur à 1 %, pour garantir un fonctionnement idéal.

Il est à noter que ces valeurs ne sont que des recommandations et qu'il est possible que la qualité de service ressentie par un utilisateur soit mauvaise bien que ces valeurs soient respectées et,

inversement que l'utilisateur soit satisfait de la qualité de service alors que ces valeurs ne sont pas respectées.

III.7.2 Exigences de QoS pour les applications de données

Le tableau 3.7 présente les exigences de QoS recommandées par l'ITU-T concernant les applications de données.

On peut constater que ces applications sont généralement moins exigeantes que les applications vidéo et audio en termes de délai (excepté pour les jeux interactifs et Telnet), mais par contre, elles nécessitent pour la plupart un taux de perte nul. Dans le cas d'un transfert de fichiers, par exemple, le délai recommandé pour qu'un utilisateur soit satisfait est très fortement lié à la taille du fichier lui-même. Dans le cas d'un fichier de plusieurs mégaoctets, l'utilisateur sera plus tolérant que pour un fichier de quelques kilooctets. Par contre, contrairement au cas des conversations audio et vidéo, l'utilisateur souhaite que son fichier soit transmis sans aucune erreur.

Application	Degré de symétrie	Quantités de données typiques	Délai aller	Taux de perte
Navigation Web HTML	Unidirectionnel	~ 10 ko	Idéal :< 2s/page Acceptable :< 4 S/page	0 %
Transfert de données	Unidirectionnel	10 ko- 10 Mo	Idéal :< 15 s Acceptable :< 60 s	0 %
Transactions à haute priorité (ex. :e-commerce)	Bidirectionnel	< 10 ko	Idéal :< 2 s Acceptable :< 4 s	0 %
Images fixes	Unidirectionnel	< 100 ko	Idéal :< 15 s Acceptable :< 60 s	0 %
Jeux interactifs	Bidirectionnel	< 1 ko	< 200 ms	0 %
Telnet	Bidirectionnel (asymétrique)	< 1 ko	< 200 ms	0 %
E-mail	Unidirectionnel	< 10 ko	Idéal :< 2 s Acceptable :< 4s	0 %
Fax	Unidirectionnel	~ 10 ko	< 30 s/page	$< 10^{-6}$ (Bit error ratio)
Application d'arrière plan (ex. :Usenet)	Unidirectionnel	~ 1 Mo	Plusieurs minutes	0 %

Tableau 3.7 – Recommandations G1010 de l'ITU-T pour les applications de données.

III.8 Conclusion

Dans ce chapitre nous nous sommes intéressés à l'étude du protocole SCTP en présentant certains de ses caractéristiques fondamentales. L'aspect fonctionnel de SCTP qui nous intéresse particulièrement est le multihoming.

En tenant compte de la possibilité qu'un hôte peut communiquer sur plusieurs technologies d'accès et de l'importance de la mobilité dans les réseaux actuels, notre proposition faite sur la base de mSCTP pour l'équilibrage de charge avec l'intégration du multihoming s'avère une bonne solution à adopter pour gérer la QoS. Pour se faire, nous avons été emmené à présenter les différentes exigences des applications les plus couramment utilisées au travers de critères tels que le délai, la gigue, la bande passante ou encore le taux d'erreur.

Chapitre 4

Implémentation sous NS2 de mSCTP avec Equilibrage de charge & Résultats de Simulation

IV Implémentation sous NS2 de mSCTP avec Equilibrage de charge & Résultats de Simulation

IV.1 Introduction

Dans ce chapitre nous allons décrire l'implémentation sous NS2 du protocole msctp avec l'extension du support d'équilibrage de charge basé sur le type de flot. Nous présentons également les scenarios de simulation avec et sans congestion, les paramètres de simulation et les métriques utilisées pour l'évaluation des performances de la proposition. Les résultats de simulation sont analysés à la fin du chapitre.

IV.2 Implémentation du protocole sctp dans ns-2 :

Dans nos simulations nous utilisons l'agent SCTP développé sous NS-2 par le laboratoire Protocol Engineering Lab de l'University de Delaware [42].

L'agent SCTP de base, tel qu'il est développé par [42], supporte les caractéristiques suivantes :

– Établissement d'une association

– Transmission des *Data Chunks*

– Acquittement des *Data Chunks* reçus

– Gestion du *Timer* de retransmission

– Des points terminaux SCTP à multi-accès (*multihomed)*

–SSN

– Livraison en/hors séquence

– Rapport des trous sur les TSNs des données reçues

– SCTP *slow start congestion Avoidance*

– Détection d'échec au niveau noeud SCTP *(Endpoint Failure Detection)*

– *Path Failure Detection*

– *Path Heartbeat* (sans le contrôle de couche supérieure)

La figure 4.1 présente la structure fondamentale modélisant le SCTP tel qu'il est implémenté sous NS 2.

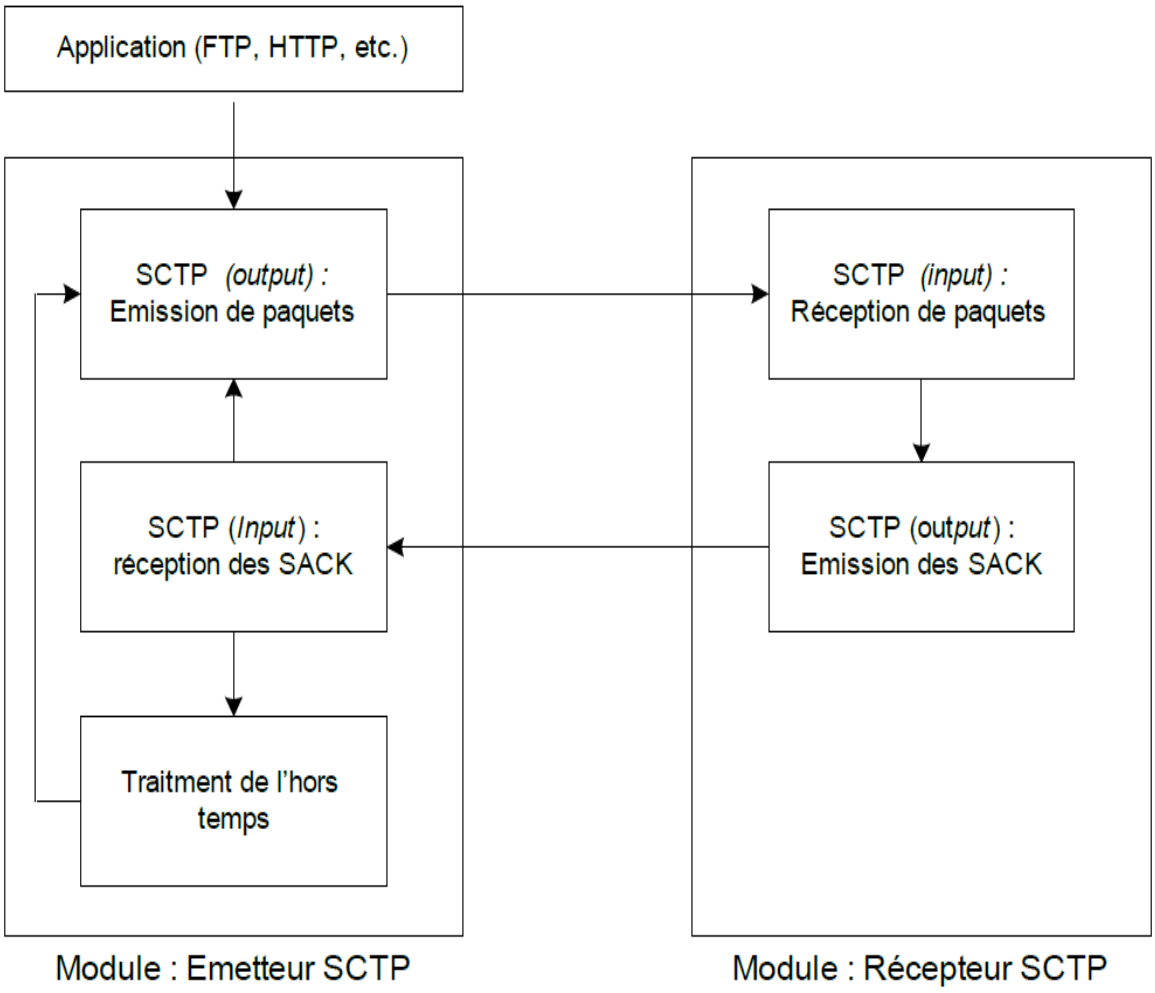

Fig. 4.1 – Structure de la modélisation SCTP sous NS-2.

En ce qui concerne un nœud à multi-accès, l'architecture actuelle de NS ne le permet pas. Par conséquent pour implémenter cette fonctionnalité, une solution a été proposée dans [43] qui consiste à voir un tel nœud comme l'ensemble d'un Core Node et d'un Interface Node pour chaque interface simulée (c'est à dire par adresse). Autrement dit chaque adresse ajoutée à un nœud multihomed est équivalente à une interface Node (c.f. la figure 4.2).

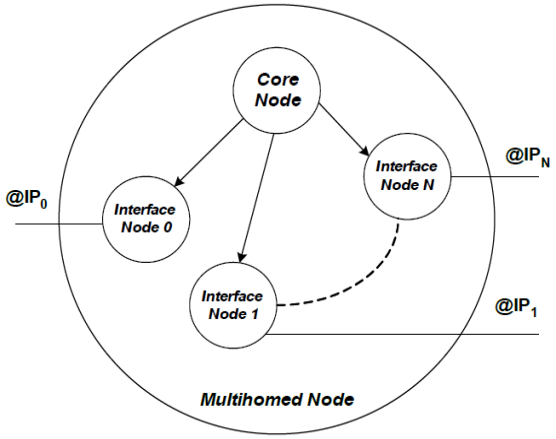

Fig. 4.2 – Nœud SCTP MultiHomed sous NS-2.

Le trafic est écoulé uniquement à travers (de/vers) les nœuds d'interface, le nœud cœur est utilisé pour le routage et est connecté à chaque nœud d'interface par un lien uni-directionnel vers ces nœuds. Par contre ces liens sont utilisés pour déterminer dynamiquement quel nœud d'interface à utiliser pour émettre vers une destination particulière. Les numéros de TSN, dans l'implémentation, commencent à partir de 1 et un TSN non défini prend la valeur de (-1) afin de contrôler le séquencement des Chunks .

Dans notre étude nous considérons une communication entre des hôtes fixe et des hôtes mobiles pour étudier le contrôle de congestion du protocole de transport SCTP, nous exploitons la fonctionnalité de multihoming de SCTP.

IV.3 Set primary address

La commande **Set Primary Address** est une commande exécutée sous le script tcl qui permet d'aiguiller un flux sur une interface donnée à un temps spécifique en indiquant une préférence d'une adresse primaire pour un type de flux. Dans l'exemple suivant :

$ns at 60.1 "$sctp38 set-primary-destination $host1_if0"

Au temps égal à '**60.1**s' le flux correspondant à '**sctp38**' change d'adresse primaire pour une autre interface qui correspond à '**host1_if0**'.

Cette commande est possible grâce au paramètre **Set primary IP Address** utilisé dans le chunk ASCONF SCTP (voir III.5.1 Extension de SCTP : Adressage dynamique – Fig 3.24 et Fig 3.25). Cela nous permet de changer une adresse primaire dans le cas du multihoming non pas pour un flux particulier mais pour l'association entière. Pour ce qui nous concerne, nous avons exploité cette propriété dans nos scripts tcl pour implémenter l'équilibrage de charge dans mSCTP.

IV.4 Model de simulation :

Le multihoming pourrait améliorer les procédures de gestion de ressources et résoudre la problématique de congestion dans les réseaux sans fils. Notre contribution est d'activer deux à trois adresses en même temps, une primaire et deux secondaires correspondant chacune à une interface avec un réseau d'accès (UMTS, WIFI et WIMAX). L'idée consiste donc à envoyer les données volumineuses demandant une bande passante importante sur le réseau présentant plus de ressources disponibles.

La simulation sous NS2 d'un scénario sans fil en utilisant le module du SCTP retourne des erreurs. En effet, l'infrastructure filaire et sans fil sur le ns-2 requièrent un schéma de routage spécial appelé « hierarchical routing », mais l'implémentation du SCTP ne permet pas d'adresser les interfaces du core-nœud, après plusieurs essais nous avons décidé de modélisé les liens sans fils (UMTS, WIFI et WIMAX) avec des liens filaires en attribuant à chacun le débit et le délai correspondants.

La **figure 4.3** présente un schéma descriptif de notre modélisation avec possibilité de plusieurs connexions concurrentes.

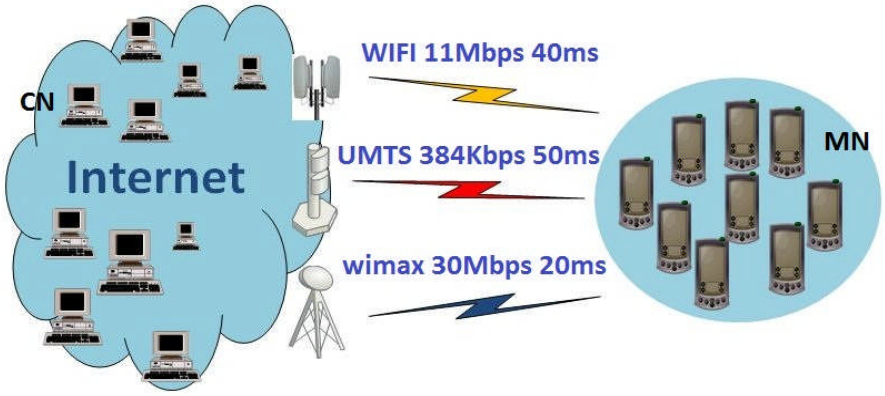

Fig. 4.3 – Schéma de modélisation.

IV.5 Caractéristiques du trafic écoulé

Nous avons utilisé un trafic suivant le model Constant Bit Rate (CBR) avec comme protocole de transport SCTP, le choix du model CBR est motivé par le fait que, d'une part le trafic CBR contrairement au trafic Variable Bit Rate (VBR) (surcharge le réseau avec des piques de débit périodique) surcharge rapidement et pour une durée illimitée le réseau.

Les caractéristiques des flux CBR utilisés tout au long de la simulation sont affichées dans le tableau suivant :

Paramètres	Audio	Video	Data
Taille de paquet (octet)	160	768	1000
Intervalle paquet (ms)	20	16	400
Taux de transfert (Kbps)	64	384	20

Tableau 4.1-Caractéristiques des flux CBR.

Trafic Audio :

Nous avons supposé l'utilisation du codeur G.711 pour la modélisation du trafic audio. Le codeur G.711 quantifie les données audio sur 8 bit avec une fréquence d'échantillonnage de 8khz et un intervalle de 20ms pour chaque paquet de taille 160 octets.

Trafic Vidéo :

Pour le trafic vidéo nous avons supposé l'utilisation de la norme mpeg-4 avec un taux de transfert de 384 kbps qui se traduit par l'envoie d'un paquet de taille 768 octets chaque 16 ms.

Trafic de Donnée :

Le trafic de donnée est variable de 10 ko à 10 Mo nous avons supposé une taille de paquet de 1000 octets avec un intervalle de 400ms.

IV.5.2 Scenarios de simulation

Pour étudier la congestion nous avons simulé deux scénarios dans différentes technologies (UMTS, WIFI et WIMAX) , l'un avec une forte congestion et l'autre avec une congestion quasi nul.

IV.5.2.1 Scenarios avec une forte congestion

Pour congestionner le réseau nous avons choisi de simuler 10 nœuds correspondants, chaque nœud correspondant émet trois trafics différents (Audio, video et data) vers 10 nœuds mobile. Le temps de simulation est de 90s (trois période de 30s).

1er cas :

Pour la 1ère période (0-30 s), les nœuds mobiles ont accès à une station de base UMTS (tous les trafics sont écoulés sur un seul lien –UMTS-).

Pour la 2ème période (30-60 s), on ajoute une station de base WIFI (les nœuds mobiles ont accès à deux technologies en même temps ce qui nous permet d'équilibrer la charge du trafic en suivant le tableau de routage) ;

Pour la 3ème période (60-90 s), une station de base WIMAX est ajoutée (ce qui rend l'équilibrage de charge optimale).

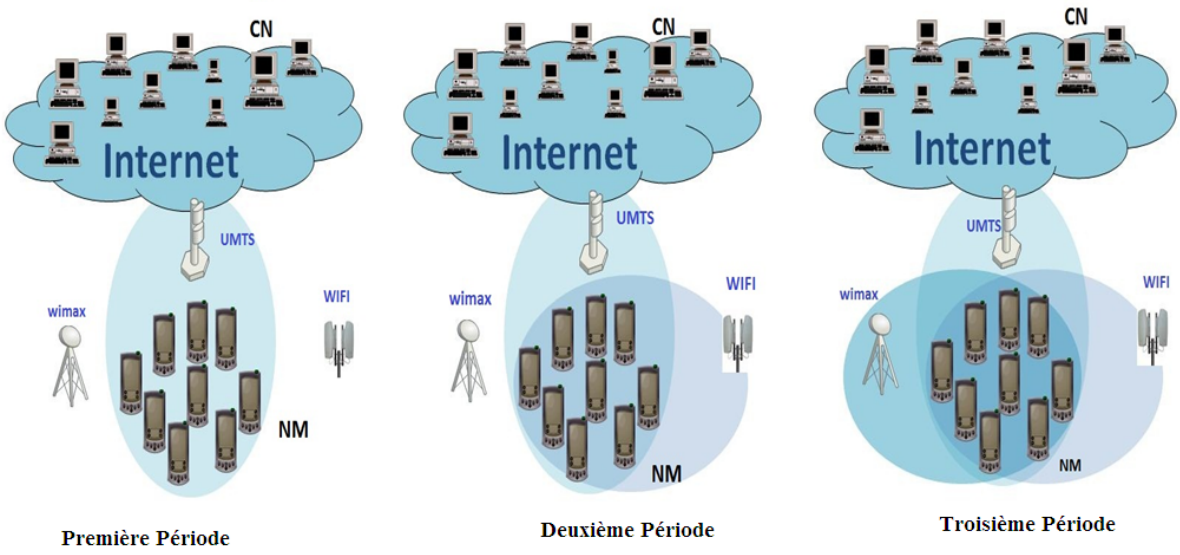

Fig. 4.4 – Scénario avec forte congestion (1er cas de simulation).

2ème cas :

Pour la 1ère période (0-30 s), les nœuds mobiles ont accès à une station de base WIFI (tous les trafics sont écoulés sur un seul lien –WIFI-).

Pour la 2ème période (30-60 s), on ajoute une station de base WIMAX (les nœuds mobiles on accès à deux technologies en même temps ce qui nous permet d'équilibrer la charge du trafic en suivant le tableau de routage).

Pour la 3ème période (60-90 s), on ajoute une station de base UMTS et on supprime la station de base WIFI.

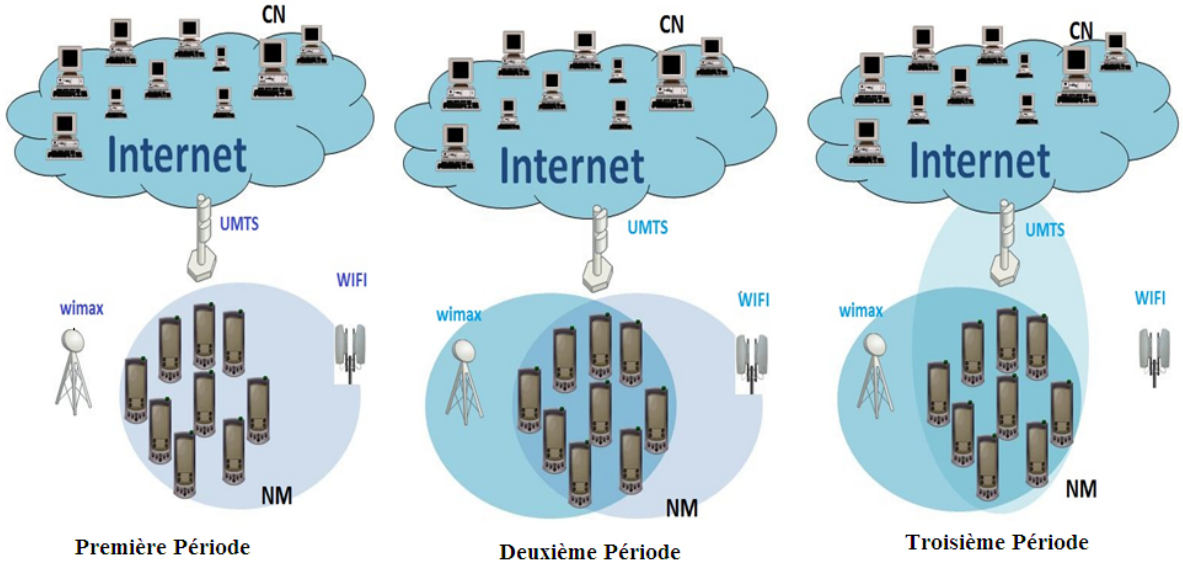

Fig. 4.5 – Scénario de forte congestion (2ème cas de simulation).

IV.5.2.2 Scenario avec une congestion quasi nulle

Pour simuler une congestion quasi nulle nous avons choisi 3 nœuds correspondants, chaque nœud émettant un trafic différent (Audio, vidéo et data) vers 3 nœuds mobiles. Le temps de simulation est de 90s (trois période de 30s). Nous reprenons les 2 cas précédents.

IV.6. Métriques utilisés pour l'évaluation de notre proposition

Le délai et le taux de perte de paquet d'un flux sont des métriques obligatoires dans l'évaluation de la Qualité de service (tableau : exigence de la QoS). Ces métriques sont calculées à partir du fichier trace récupéré à la fin de chaque simulation.

Le fichier trace est structuré en lignes (comme déjà vu dans la présentation du ns-2). Pour extraire et faire des calculs sur les données trace nous avons eu recours au langage Awk.

IV.6.1 Les métriques calculées

Le délai de bout en bout :

Il est dit aussi temps de réponse. Il s'agit du temps d'attente pour mesurer le temps écoulé pour la transmission du flux. La maîtrise du délai de transmission est un élément essentiel pour bénéficier d'un véritable mode conversationnel et minimiser la perception d'écho. Or la durée de traversée d'un réseau dépend de nombreux facteurs:

• Le débit de transmission sur chaque lien ;

• Le nombre d'éléments réseaux traversés.

Le délai de transport est la durée passée à traverser les routeurs, les commutateurs et les autres composants du réseau. Le temps de traversée de chaque élément est fonction de la puissance et la charge de ce dernier, du temps de mise en file d'attente des paquets, et du temps d'accès en sortie de l'élément. L'ordre de grandeur est de quelques dizaines de millisecondes. Le délai de propagation de l'information, est généralement très faible par rapport aux autres composantes du délai de transit.

Le délai est le temps que prend un paquet qui sort du nœud source pour atteindre le nœud de destination. Pour le calcul du délai de chaque paquet nous avons utilisé les données enregistrées dans le trace file en utilisent awk de la manière suivante :

Délai du paquet= temps à la réception du paquet – le temps a la sortie du paquet

Le script AWK pour le calcul du délai pour le trafic Audio:

```
BEGIN { highest_packet_id = 0;}
{   action = $1;
   time = $2;
   flow_id = $8;
   packet_id = $12;
```

```
    if ( action == "-"  && flow_id == 123456789 )
        start_time[packet_id] = time;
    if ( flow_id == 123456789 && action == "r" ) {
        end_time[packet_id] = time; }
}
END {  for ( packet_id = 0; packet_id < NF; packet_id++ ) {
      start = start_time[packet_id];
      end = end_time[packet_id];
      packet_duration = end - start;
      if ( start < end ) printf("%f %f\n", end, packet_duration);}
   }
```

La Gigue:

La gigue représente la variation du délai se phénomène transforme un trafic périodique en non périodique, elle doit être stable pour une bonne qualité de service. L'une des raisons de cette irrégularité d'arrivée des paquets est la congestion du réseau à un instant donné.

La gigue est calculée par la différence entre le délai de deux paquets successif, nous avons utilisé le fichier obtenu par le script awk du délai pour calculer la gigue en lui ajoutons un compteur de ligne dans la troisième colonne.

Le script AWK pour le calcul de la gigue pour le trafic Audio:

```
BEGIN { dala = 0 ;
       time = 0 ; }
{
  temps = $1;
  delay = $2;
  i = $3;
  dela[i] = delay ;
  dela[i+1] = delay;
  time[i+1] = temps;
}
END { for ( i = 1; i < 7977; i++ ) {
      newdelay = dela[i];
      lastdelay = dela[i+1];
      start = time[i+1];
      jitter = newdelay - lastdelay;
      printf("%f\t%f\n", start, jitter);}
}
```

Taux de perte des paquets :

Lorsque le réseau est congestionné, un processus ce déclenche pour libérer de la bande passante en se débarrassant d'une certaine proportion des paquets entrants, en fonction de seuils prédéfinis. Le destinataire émet des acquittements négatifs indiquant qu'il ne reçoit plus les paquets. Nous avons choisi de le calculer dans un intervalle de 30seconde (pour chaque période).

Le taux de pertes = nombre de paquets non arrivés / le nombre total de paquets transmis.

Le script AWK pour le calcule du taux de perte pour le trafic Audio:

```
BEGIN { fsDrops = 0;
        numFs = 0;
        taux = 0;}
{  action = $1;
   time = $2;
   flow_id = $8;
if ( $2 < 30 ) { time = $2;
        if (flow_id == 123456789 && action == "-"  )
              numFs++;
        if (flow_id == 123456789 && action == "d"  )
              fsDrops++;
     taux = fsDrops/numFs;
printf("%f %f\n", time, fsDrops/numFs);}}
END {}
```

Le Débit (Throughput) :

Le débit binaire ou par abus de langage, la bande passante, entre deux systèmes communicants est le nombre de bits que le réseau est capable d'accepter ou de délivrer par unité de temps. C'est le taux de transfert maximum pouvant être maintenu entre deux nœuds. La bande passante d'un lien réseau représente sa capacité de transport, mesurée en bits par seconde, dans laquelle les données n'incluent pas les bits de paquets retransmis. C'est cette capacité utile qui est le paramètre pertinent pour l'application.

Débit = volume de données reçues / le temps écoulé.

Ce calcul est représenté par 2 script awk :

Le premier script sélectionne la taille de chaque paquet reçue avec le temps de simulation.

```
BEGIN {}
{
if ( $8 == 123456789 && $1 == "r") {
printf("%f %f\n", $2, $6) ;        }
 }
END {}
```

Le second script agie sur le résultat du 1er script pour sommer les tailles des paquets.

```
BEGIN { sum = 0 ;
}
{ if ( $1 > 0 ) {
      sum = sum + $2 ;
      printf("%f %f\n", $1, sum/$1) ; }
}
END {  }
```

Les fichiers textes obtenus après l'exécution de chaque script awk permettent de tracer les graphes en utilisons xgraph ou gnuplot.

IV.7 Résultats de simulation et Discussion

Comme nous l'avons vu dans le paragraphe IV.5, nous avons simulé deux scenarios : l'un avec une forte congestion et l'autre avec une faible congestion ; dans chacun des deux scenarios, nous avons introduit deux cas de figure. Par la suite nous allons présenter les graphes des différents trafics avec les métriques correspondants.

IV.7.1. Scenario : Performances en cas de congestion

1er cas : 1ère Période : le trafic s'écoule sur l'UMTS
2ème Période : le trafic s'écoule sur l'UMTS et le WIFI
3ème Période : le trafic s'écoule sur l'UMTS, WIFI et le WIMAX

A) Flux Data

Interprétation :
Le flux Data à un faible Débit sur la première période, pour la deuxième et la troisième le Débit se stabilise. Puisqu'il reste seul sur l'UMTS.

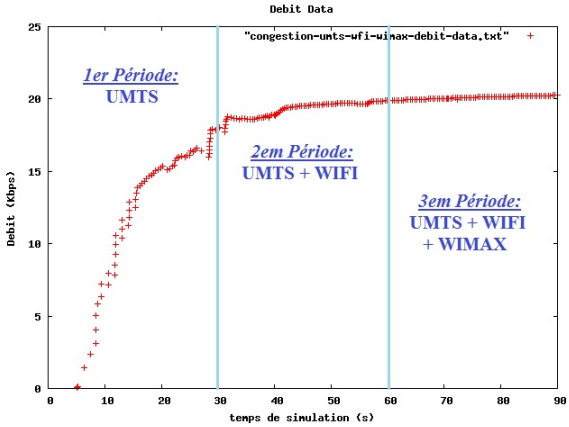

Fig. 4.6 - Débit du flux Données 1er Scenario 1er cas.

Interprétation :
Le taux de perte est considérable dans la première période, la deuxième et la troisième période le taux de perte est nul.

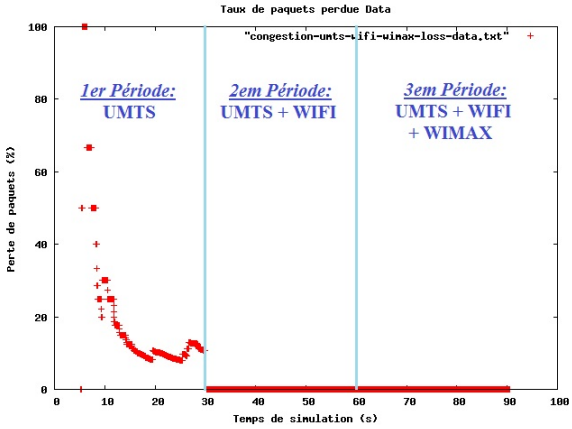

Fig. 4.7 - Taux de perte du flux Données
1er Scenario 1er cas.

B) Flux Audio

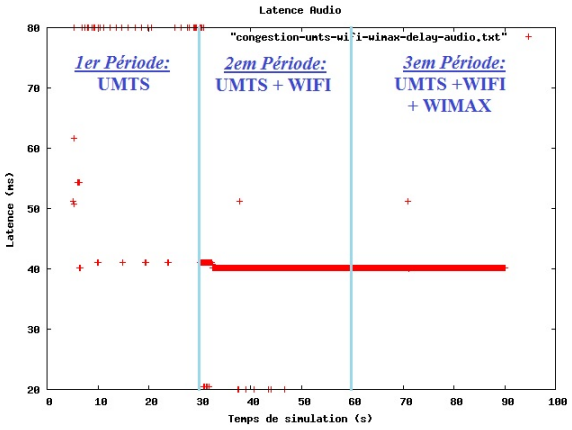

Fig. 4.8 - la Latence du flux Audio
1er Scenario 1er cas.

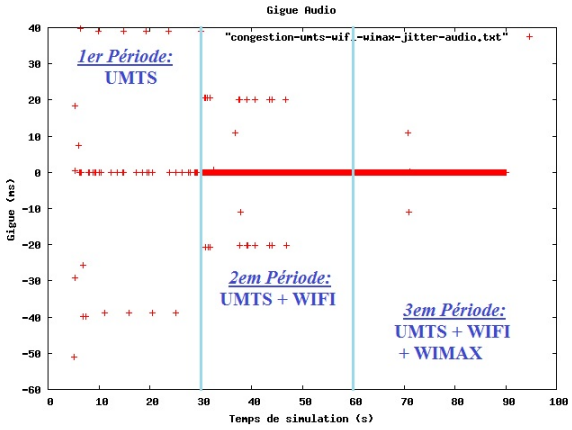

Fig. 4.9 - La gigue du flux Audio
1er Scenario 1er cas.

Interprétation :
La Latence est de 80ms sur la première période, a partir de la deuxième et troisième le flux audio bascule sur le WIFI la latence diminue jusqu'à 40ms.

Interprétation :
La gigue est instable dans la première période, la deuxième elle présente une faible variation et sur la troisième période elle sa nul.

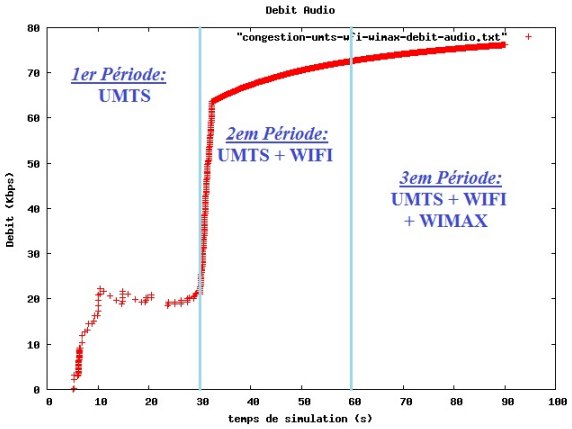

Fig 4.10 - Débit du flux Audio
1er Scenario 1er cas.

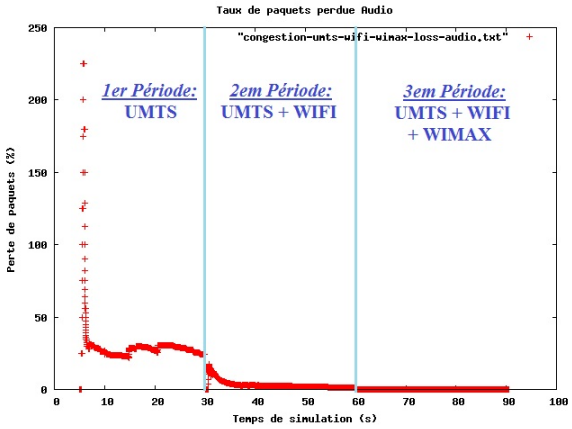

Fig. 4.11 : Taux de perte du flux Audio
1er Scenario 1er cas.

Interprétation :
On remarque que le Débit n'est considérable qu'à partir de la deuxième période.

Interprétation :
Le taux de perte est considérable dans la première période puis il diminue jusqu'à se qu'il s'annule

C) Flux Vidéo

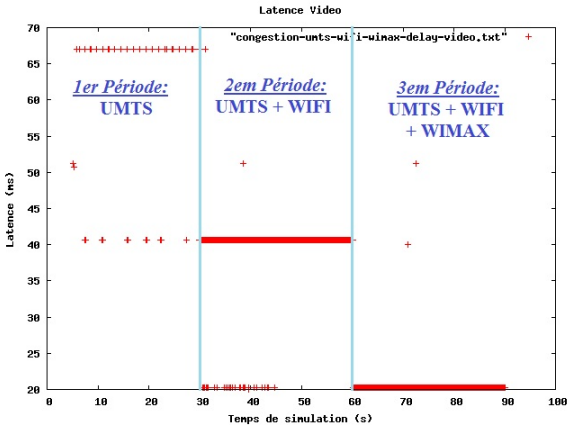

Fig. 4.12 - Latence du flux Vidéo
1er Scenario 1er cas.

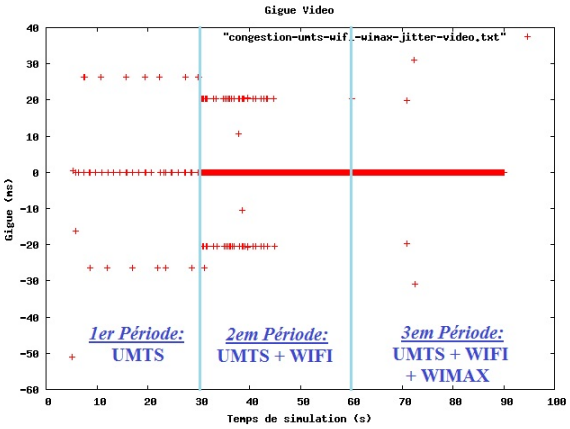

Fig. 4.13 – Gigue du flux Vidéo
1er Scenario 1er cas.

Interprétation :
La latence du flux a voisine 67 ,5 ms dans la première période, durons la deuxième période elle diminue jusqu'à 40ms est diminue encor sur la troisième période à 20ms.

Interprétation :
La gigue est instable sur la première période et sur la moitié de la deuxième, Elle est nul sur le temps restons de simulation.

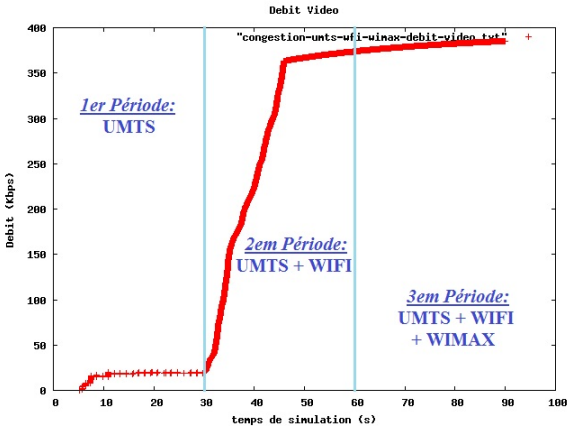

Fig. 4.14 - Débit du flux Vidéo
1er Scenario 1er cas.

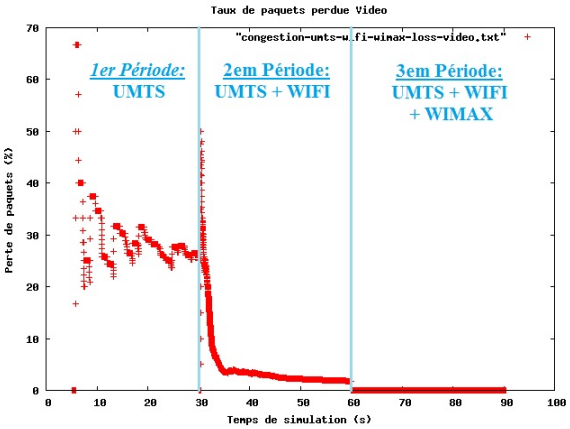

Fig. 4.15 - Taux de perte du flux Vidéo
1er Scenario 1er cas.

Interprétation :
Le flux vidéo est le plus charger on remarque que le Débit n'atteint sons maximum qu'a partir de la moitié de la deuxième période.

Interprétation :
Le taux de perte s'annule quand le trafic s'écoule sur le WIMAX dans la troisième période.

2er cas : 1ère Période : le trafic s'écoule sur WIFI
2ème Période : le trafic s'écoule sur WIFI et WIMAX
3ème Période : le trafic s'écoule sur WIMAX et UMTS

A) Flux Data

Interprétation :
Le Débit du flux Data augmente jusqu'à 18Kbps sur la fin sur la première période, pour la deuxième et la troisième le Débit atteint a maximum de 20Kbps.

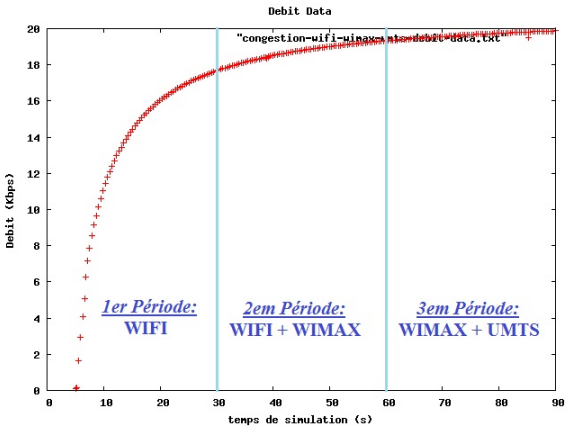

Fig. 4.16 - Débit du flux Données 1er Scenario 2ème cas.

Interprétation :
Dans la première période tous les trafics passent par le WIFI, ce qui explique la perte de paquets, dans la deuxième période le trafic Data bascule sur l'interface WIFI puis dans la troisième période sur l'UMTS.

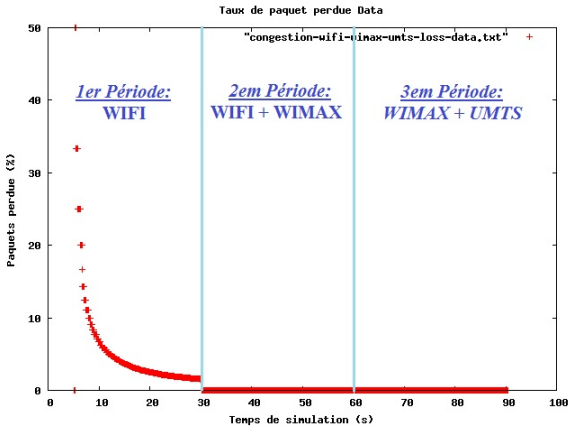

Fig. 4.17 - Taux de perte du flux Données 1er Scenario 2ème cas.

B) Flux Audio

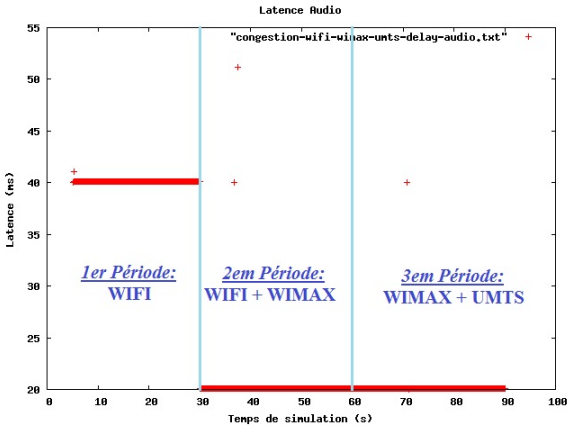

Fig. 4.18 - Latence du flux Audio
1er Scenario 2ème cas.

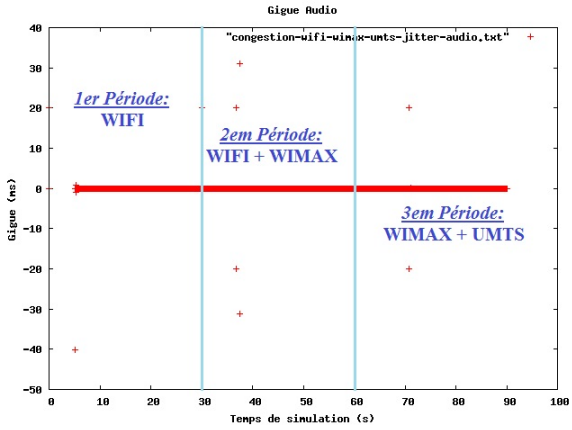

Fig. 4.19 - Gigue du flux Audio
1er Scenario 2ème cas.

Interprétation :
La latence sur la première période est de 40ms puisque tous les flux passent sur le WIFI, la deuxième et la troisième période présentent une latence de 20ms puisque l'Audio bascule sur le WIMAX.

Interprétation :
Puisque la latence est stable, la gigue est nulle durant tout le temps de simulation.

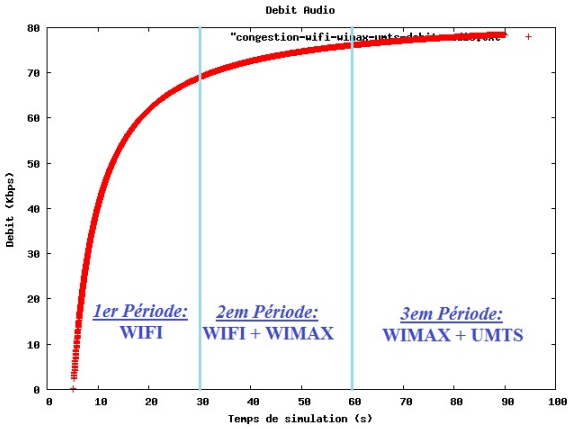

Fig. 4.20 - Débit du flux Audio
1er Scenario 2em cas.

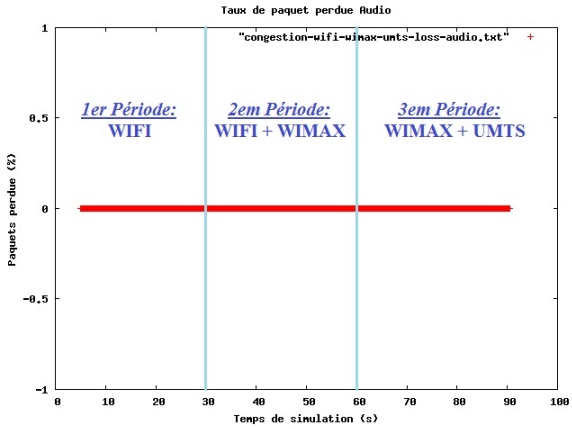

Fig. 4.21 - Taux de perte du flux Audio
1er Scenario 2em cas.

Interprétation :
Le flux vidéo atteint son maximum à la fin de la première période.

Interprétation :
Il n'y a pas de perte de paquet durant tous le temps de simulation.

C) Flux Vidéo

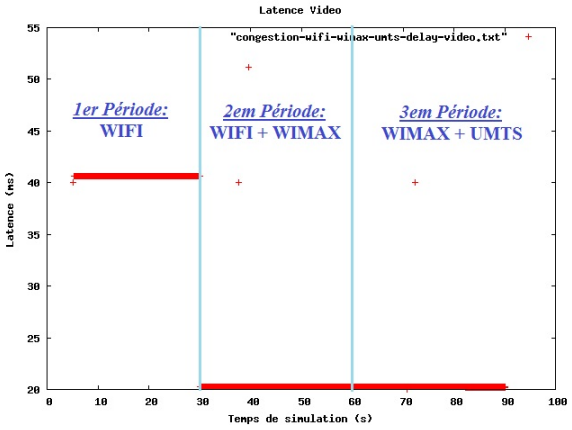

Fig. 4.22 - Latence du flux Vidéo
1er Scenario 2ème cas.

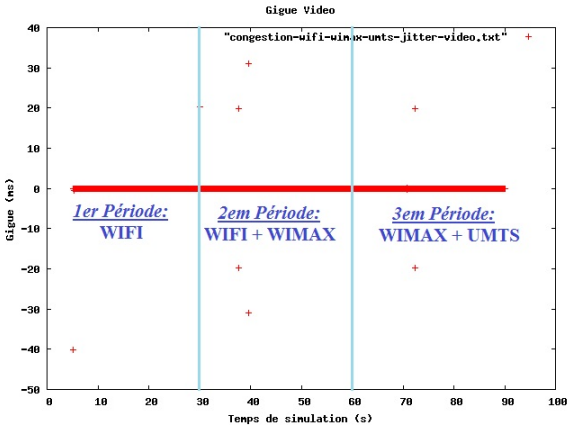

Fig. 4.23 – Gigue du flux Vidéo
1er Scenario 2ème cas.

Interprétation :
La première période présente une latence de 40ms quant le flux Audio bascule sur le WIMAX la latence diminue de 20ms jusqu'à la fin de la simulation.

Interprétation :
La stabilité de la latence annule la gigue.

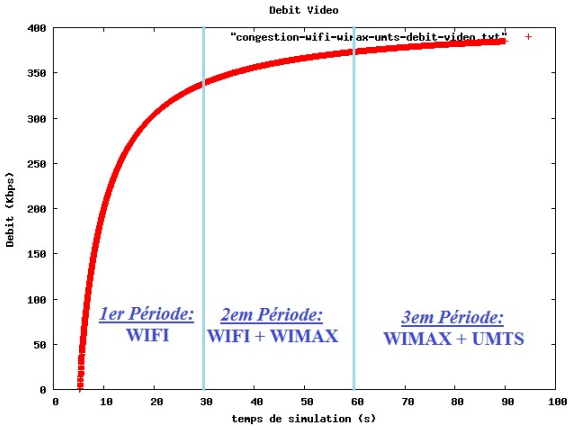

Fig. 4.24 - Débit du flux Vidéo
1er Scenario 2ème cas.

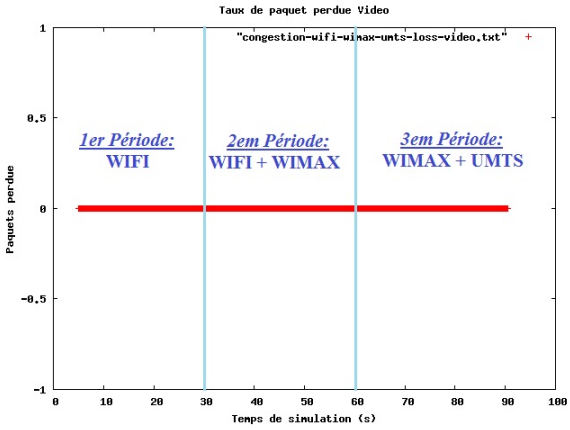

Fig. 4.25 - Taux de perte du flux Vidéo
1er Scenario 2ème cas.

Interprétation :
Le flux Vidéo atteint 347 Kbps à la fin de la première période par la suite dans la deuxième et la troisième période il atteint un maximum de 386Kbps.

Interprétation :
Il n'y a pas de perte de paquet pour tout le temps de simulation.

IV.7.2 Scenario : Performances en absences de congestion

1er cas : 1ère Période : le trafic s'écoule sur l'UMTS
2ème Période : le trafic s'écoule sur l'UMTS et le WIFI
3ème Période : le trafic s'écoule sur l'UMTS, WIFI et le WIMAX

A) Flux Data

Interprétation :
Le Débit du flux Data atteint 18Kbps au début de la deuxième période.

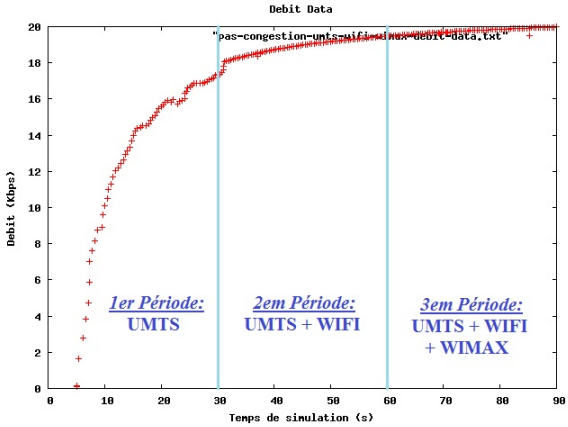

Fig. 4.26 - Débit du flux Données 2ème Scenario 1er cas.

Interprétation :
Le taux de pertes dans la première période est considérable puisque tout le trafic passe sur l'interface de l'UMTS, Dans la deuxième et troisième période le taux de perte est nul.

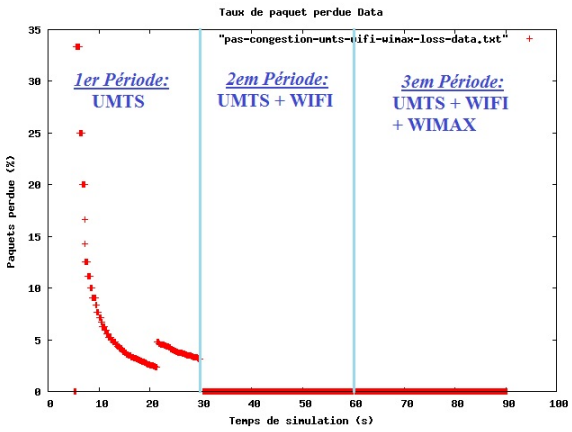

Fig. 4.27 - Taux de perte du flux Données 2ème Scenario 1er cas.

B) Flux Audio

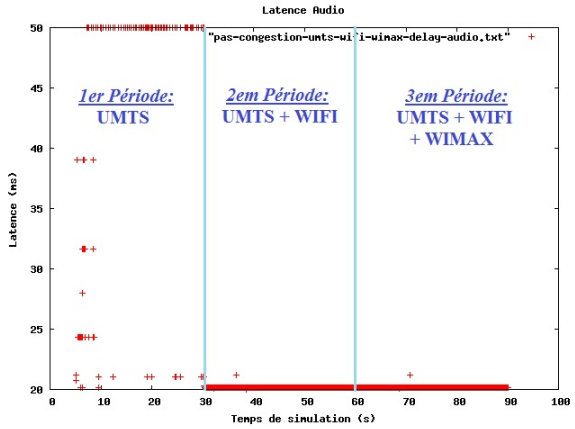

Fig. 4.28 - Latence du flux Audio
2ème Scenario 1er cas.

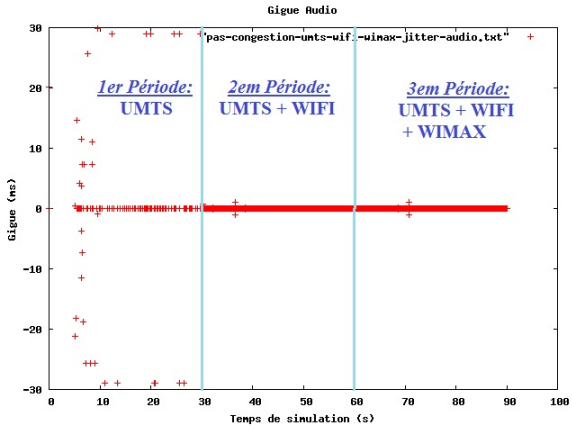

Fig. 4.29 - Gigue du flux Audio
2ème Scenario 1er cas.

Interprétation :
La latence atteint un maximum de 50ms sur la premier période, quand le trafic bascule dans l'interface WIFI la Latence diminue jusqu'à 20ms.

Interprétation :
La première période présente une gigue instable due à la variation de la latence, dans la deuxième et la troisième période la latence se stabilise se qui annule la gigue.

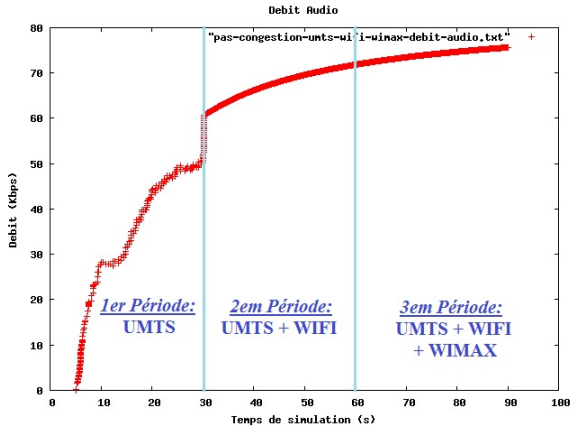

Fig. 4.30 - Débit du flux Audio
2ème Scenario 1er cas.

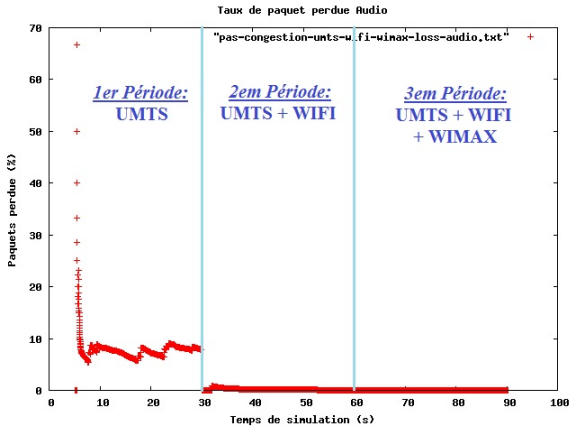

Fig. 4.31 - Taux de perte du flux Audio
2ème Scenario 1er cas.

Interprétation :
Le Début de la deuxième période le flux atteint 60Kbps.

Interprétation :
La première période présente une perte de paquets qui s'annule dans les deux périodes qui suivent.

C) Flux Vidéo

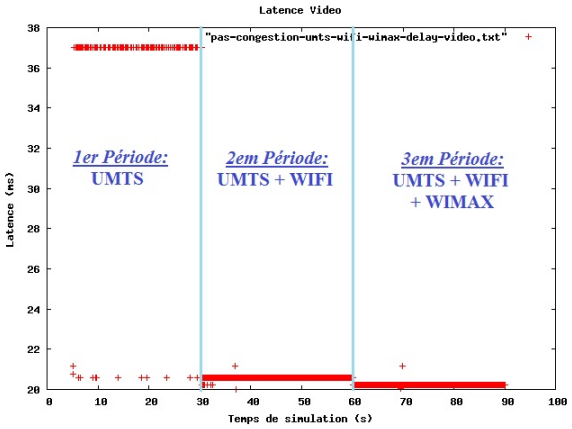

Fig. 4.32 - Latence du flux vidéo
2ème Scenario 1er cas.

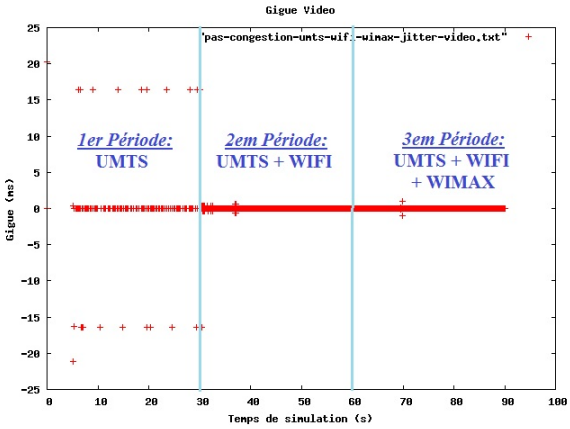

Fig. 4.33 - Gigue du flux vidéo
2ème Scenario 1er cas.

Interprétation :
La Latence atteint 37ms sur la première période, sur la troisième période la latence diminue jusqu'à 20ms.

Interprétation :
La gigue n'est pas stable sur la première période, la deuxième et troisième période la gigue est nulle.

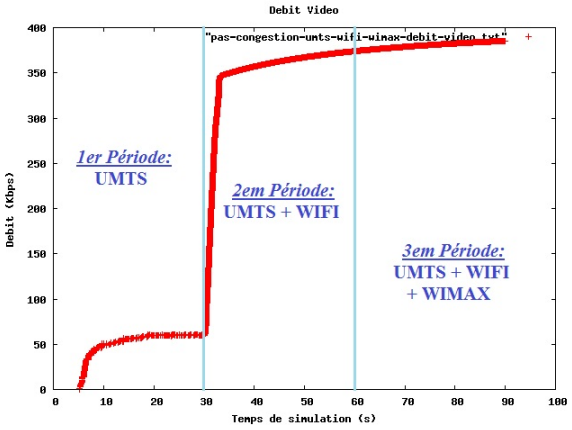

Fig. 4.34 - Débit du flux vidéo
2em Scenario 1er cas.

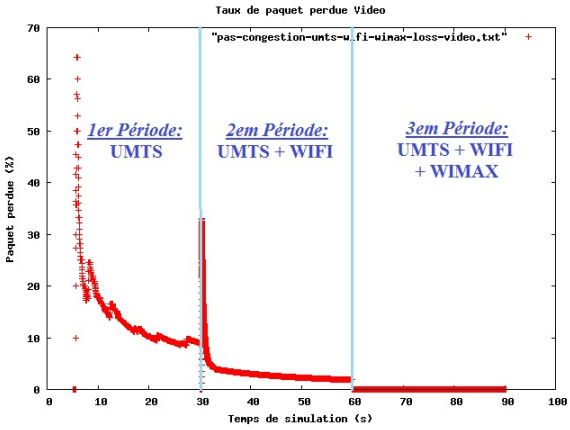

Fig. 4.35 - Taux de perte vidéo
2em Scenario 1er cas.

Interprétation
Le Débit du flux Vidéo augmente à partir de la deuxième période.

Interprétation :
On remarque que la perte de paquet est présente au début de la deuxième période.

2er cas : 1ère Période : le trafic s'écoule sur WIFI
2ème Période: le trafic s'écoule sur WIFI et WIMAX
3ème Période: le trafic s'écoule sur WIMAX et UMTS

A) Flux Data

Interprétation :
Le débit atteint un maximum de 18 Kbps à la fin de la première période.

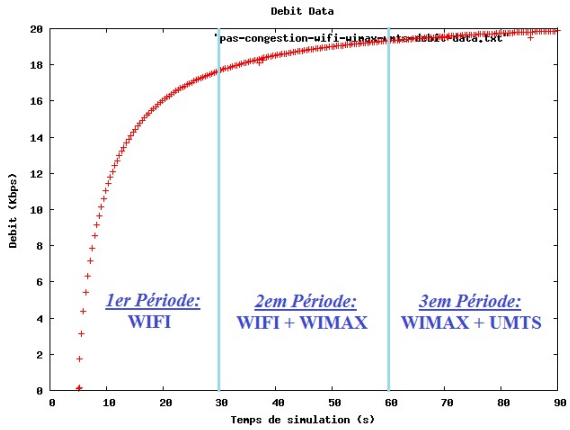

Fig. 4.36 - Débit du flux Données 2ème Scenario 2ème cas.

Interprétation :
Il n'y a pas de perte de paquets sur tout le temps de simulation.

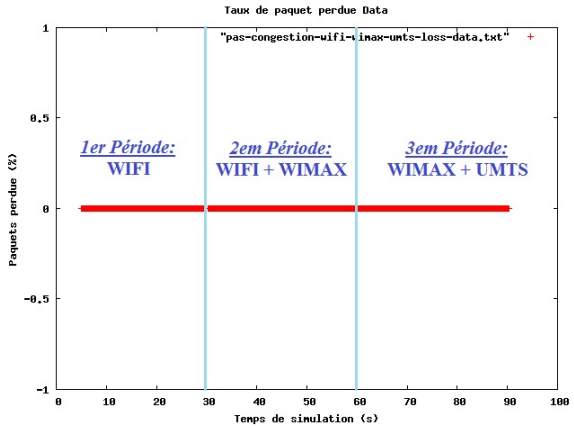

Fig. 4.37 - Taux de perte du flux Données 2ème Scenario 2ème cas.

B) Flux Audio

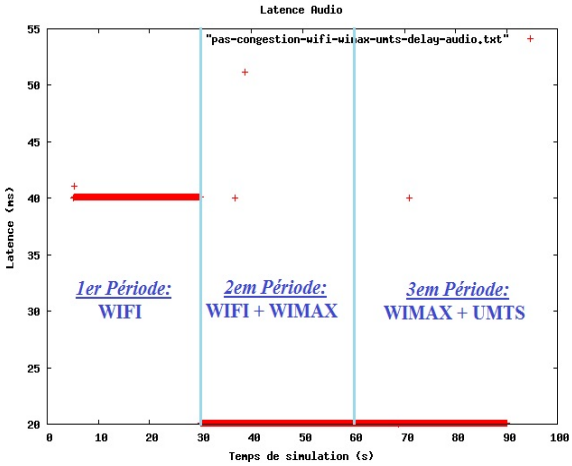

Fig. 4.38 - Latence du flux Audio
2ème Scenario 2ème cas.

Interprétation :
Dans la première période la latence est de 40ms, sur la deuxième période le trafic bascule sur l'interface WIMAX est la latence diminue jusqu'à 20ms.

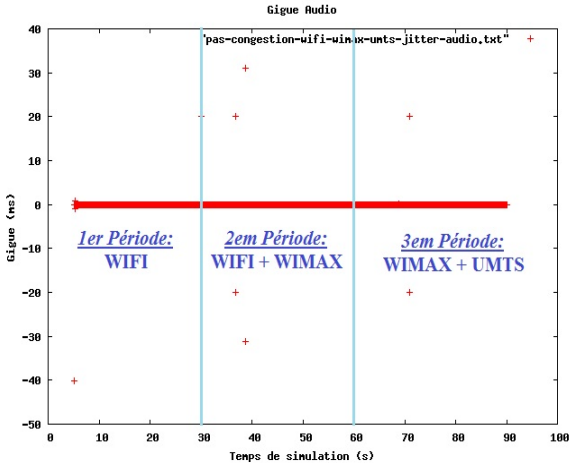

Fig. 4.39 - Gigue flux Audio
2ème Scenario 2ème cas.

Interprétation :
La stabilité de la Latence annule la gigue.

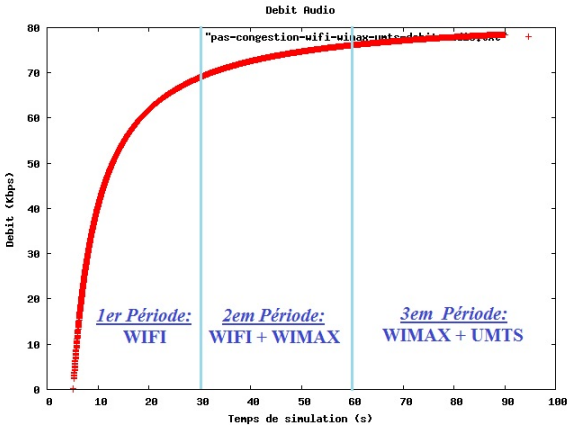

Fig. 4.40 - Débit flux Audio
2ème Scenario 2em cas.

Interprétation :
Le Débit atteint 70Kbps à la fin la première période.

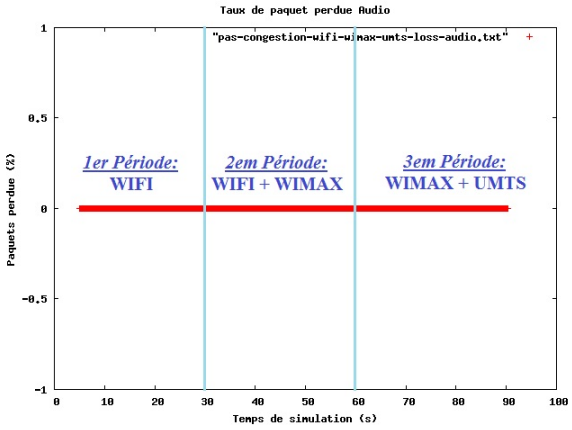

Fig. 4.41 - Taux de perte flux Audio
2ème Scenario 2ème cas.

Interprétation :
Il n'y a pas de perte de paquets sur tout le temps de simulation.

C) Flux Vidéo

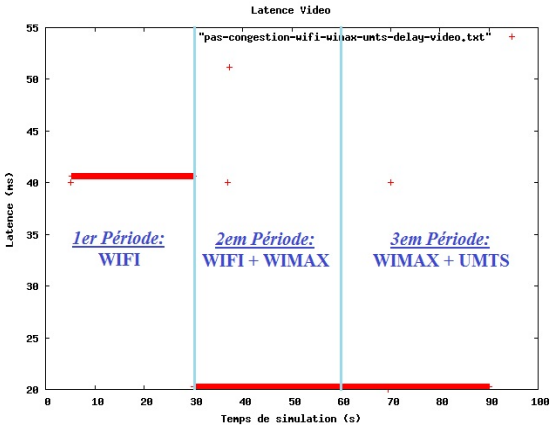

Fig. 4.42 - Latence du flux Vidéo
2ème Scenario 2ème cas.

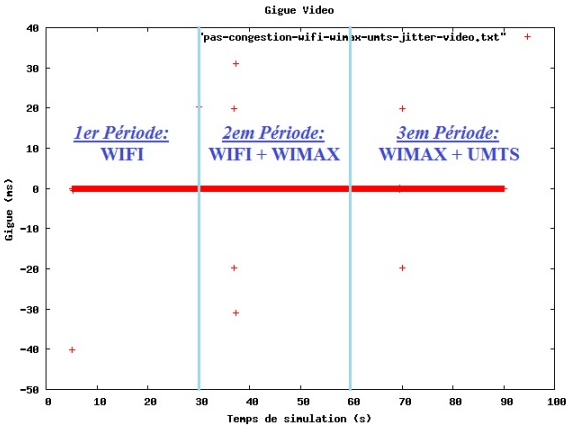

Fig. 4.43 - Gigue flux Vidéo
2ème Scenario 2ème cas.

Interprétation :
Dans la premier période tous le trafic s'écoule sur l'interface WIFI la Latence est de 40ms, la deuxième période le flux de la Vidéo bascule sur WIMAX la latence diminue jusqu'à 20ms.

Interprétation :
La latence est stable et elle agit directement dans la gigue en l'annulant.

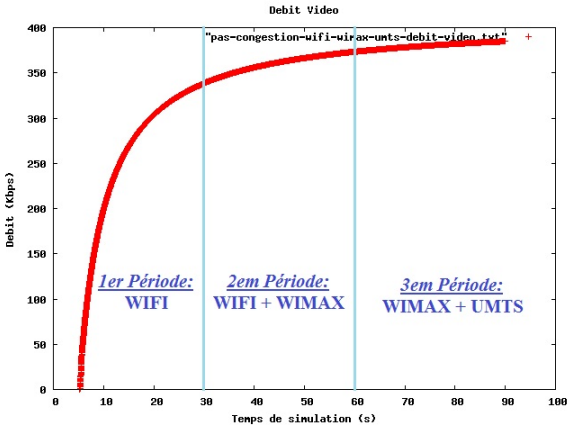

Fig. 4.44 - Débit flux Vidéo
2ème Scenario 2ème cas.

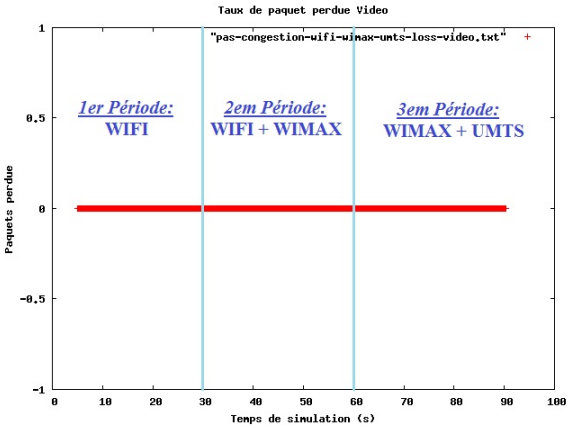

Fig. 4.45 - Taux de perte flux Vidéo
2ème Scenario 2ème cas.

Interprétation :
Le Débit atteint 340 Kbps à la fin de la première simulation jusqu'à atteindre 387 Kbps à la fin de la simulation.

Interprétation :
Il n'y à pas de perte de paquets sur tous le temps de simulation.

IV.8 Conclusion

Au sein d'une association SCTP l'ouverture de plusieurs liens permet d'éviter le problème de congestion. Cet aspect a été démontré par ce chapitre qui s'est focalisé à prouver l'avantage du Multihoming dans une zone ou plusieurs technologies d'accès sont possibles. Les simulations que nous avons menées ont montré que mSCTP peut présenter un service d'équilibrage de charge. En fait, l'équilibrage de charge offre un meilleur écoulement du trafic visualisé par les applications en fonction du taux d'erreur. Cet avantage est la solution la plus approprié pour régler les problèmes de qualité de service dans un cas de congestion. Notre politique de routage a fait office d'un mécanisme qui gère la décision d'aiguillage du trafic sur le plan d'exploitation de la bande passante. Les résultats obtenus sont satisfaisants et permettent d'envisager une amélioration de la proposition avec la prise en charge d'autres paramètres de connexion dans la politique de routage.

Conclusion générale

Dans ce projet nous nous sommes intéressés au protocole SCTP (*Stream Control Transmission Protocol*) qui est un protocole de transport présentant des améliorations par rapport aux autres protocoles de transport tels que TCP et UDP.

Le premier chapitre a été axé sur les différents types de mobilité et à sa gestion dans les réseaux en général pour entourer notre problématique. Nous avons pu discuter de la mobilité d'une session de connexions réseaux, concept qui permet de regrouper la mobilité des terminaux ou des utilisateurs.

Dans le deuxième chapitre, nous avons étudié des solutions de gestion de la mobilité dans les réseaux IP. Nous avons constaté des problèmes en termes de délai et de perte de paquets au niveau du mécanisme de gestion de mobilité dans la couche IP. Plusieurs propositions sont faites pour résoudre ces problèmes au plus haut niveau, telles que l'extension du protocole TCP et le protocole SCTP qui permettent d'utiliser plusieurs adresses IP aux deux extrémités, et cela même simultanément pour les machines multi-domiciliées.

Dans le troisième chapitre nous avons étudié le protocole SCTP en présentant certains de ses caractéristiques fondamentales. L'aspect fonctionnel de SCTP qui nous intéresse particulièrement est le multihoming. Cette étude critique, nous a permis de mettre en évidence les insuffisances des solutions proposées dans la littérature, ce qui nous a aidé à formuler notre approche basée sur plusieurs associations du multihoming instantané à la mobilité. Cette fonctionnalité présente une résolution du problème de congestion. Notre approche a consisté à introduire un mécanisme d'équilibrage de charge.

Notre but est d'améliorer les performances du SCTP particulièrement dans un contexte de réseau mobile à transmission de données et à services multiples dans un environnement hétérogène. Le choix d'un tel environnement est justifié par le fait que nous visons, par cette étude, à assurer une qualité de service optimale dans un cadre exigeant une bande passante importante, des délais importants en exploitant plusieurs technologies d'accès simultanément. Pour le faire, nous avons, d'abord présenté les caractéristiques générales du protocole SCTP et principalement le mécanisme de multihoming, justifiant notre choix pour ce protocole. Ensuite, nous avons étudié et détaillé l'approche de gestion de congestion proposée par le protocole.

Le quatrième chapitre est consacré aux résultats de simulation. Nos simulations ont été réalisées au moyen du simulateur de réseau NS2. Enfin, nous avons procédé à l'évaluation de la technique proposée à la base des simulations. Cette aspect d'équilibrage de charge SCTP performe mieux que le SCTP standard dans un environnement multi-accès que ce soit dans le cas d'augmentation de la taille

des données transmises ou dans le cas de variations du débit sur une interface. Nous avons relevé des améliorations globales dans le comportement du protocole, et ce pour des modèles de trafics variés (Data, Audio et vidéo). Avec l'introduction du mécanisme d'équilibrage de charge, nous avons constaté une réduction des délais de transmission, taux d'erreur et utilisation du débit par les paquets transmis, conformément aux exigences des différentes applications en matière de qualité de service.

Dans ce travail nous n'avons pas considéré l'aspect du multi-adressage (Multistreaming) qui est une autre caractéristique du SCTP et qui contribue de façon importante sur le plan de gestion de mobilité. Vu les limitations de NS, nous n'avons pas pu tester l'aspect multiservice au vrai sens du terme. Delà ce travail peut être amélioré en considérant un stream par classe de service autrement dit un stream pour un trafic de type temps réel (Audio, Vidéo), un autre pour un service non temps réel (Data)…Ce type d'implémentation nécessite une plateforme de test. Tous ces aspects offriront une meilleure adaptation aux nouveaux services offerts aux utilisateurs des réseaux mobiles de nouvelles générations.

Bibliographie

[01] D. Mitton, M. St.Johns, S. Barkley, D. Nelson, B. Patil, M. Stevens, B. Wolff. Authentication, Authorization, and Accounting. RFC 3127, Juin 200.

[02] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, Mars 1997.

[03] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315, July 2003.

[04] P. Mockapetris. DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. RFC 1035, Novembre 1987.

[05] J. Postel. INTERNET CONTROL MESSAGE PROTOCOL. RFC 792, Septembre 1981.

[06] David C. Plummer, An Ethernet Address Resolution Protocol. RFC 826, Novembre 198

[07] Basavaraj Patil, Phil Roberts. Reverse Tunneling for Mobile IP. RFC 3024, Janvier 2001.

[08] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401, Novembre 1998.

[09] D. Johnson, C. Perkins, J. Arkko. Mobility Support in IPv6. RFC 3775, Juin 2004.

[10] H. Soliman, C. Castelluccia, K. El Malki, L. Bellier. Hierarchical Mobile IPv6 Mobility Management (HMIPv6). RFC 4041, Août 2005.

[11] R. Koodli. Mobile IPv6 Fast Handovers. RFC 5268, Juin 2008.

[12] C. Huitema. Multi-homed TCP. IETF Internet Draft draft-huitema-multi-homed-01, Mai 1995.

[13] A. Snoeren and H. Balakrishnan. An End-to-End Approach to Host Mobility. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), pages 155–166, Boston, MA, Août 2000.

[14] W. Diffie and M. E. Hellman. Privacy and Authentication: An Introduction to Cryptography.

In Proceedings of the IEEE, volume 67, pages 397–427, Mars 1979.

[15] NIST (National Institute of Standards and Technology). The Secure Hash Algorithm (SHA-1), Avril 1995. NIST FIPS PUB 180-1.

[16] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), Juin 1999. Mis à jour par RFC 2817.

[17] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), Juin 2002.

[18] H. Schulzrinne and E. Wedlund. Application-layer mobility using SIP. ACM SIGMOBILE Mobile Computing and Communications Review, 4(3):47–57, Juillet 2000.

[19] P. Karn and W. Simpson, «Session-Key Management Protocol», RFC2522, Internet Engineering Task Force, March 1999.

[20] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson, «Stream Control Transmission Protocol», RFC2960, Internet Engineering Task Force, October 2000.

[21] Randall Stewart, Chris Metz, «SCTP New Transport Protocol for TCP/IP», IEEE Internet Computing, vol.5, no.6, pp.64-69, November-December 2001.

[22] R. Stewart, Qiaobing Xie, «Stream Control Transmission Protocol (SCTP) : a reference guide», Addison wesley, London 2002.

[23] Inwhee Joe and Latha Kant, «SCTP with an improved cookie mechanism for wireless networks through modeling and simulation», 58th IEEE VTC Fall, vol.4, pp.2559-2563, October 2003.

[24] Shaojian Fu and Mohammed Atiquzzaman, «SCTP: state of the art in research, products, and technical challenges», IEEE Communications Magazine, vol.42, no. 4, April 2004, pp.64-76.

[25] Shaojian Fu, Mohammed Atiquzzaman and William Ivancic, «Evaluation of SCTP for Space Networks», IEEE Wireless Communications, vol.12, no 5, October 2005, pp. 54-62.

[26] M. Allman, V. Paxson, W. Stevens, «TCP Congestion Control»,RFC2581, Internet Engineering Task Force, April 1999.

[27] Ivan Arias Rodriguez, «Stream Control Transmission Protocol The design of a new reliable transport protocol for IP networks», Helsinki University of Technology, Electrical and Communications Engineering Department Networking Laboratory, Thesis report, Espoo 12 Februray 2002.

[28] J.Iyengar, K.Shah, P.Amer, R.Stewart, «Concurrent Multipath Transfer Using SCTP Multihoming», SPECTS 2004.

[29] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, P. Conrad, «Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration», draft-ietf-tsvwg-addip-sctp-11.doc, Internet Draft, February 2005 (Work in Progress).

[30] M. Riegel, M. Tuexen, «Mobile SCTP»**,** draft-riegel-tuexen-mobile-sctp-04.txt, Internet- Draft , October 2004 (Work in Progress).

[31] C. Perkins, « IP Mobility Support for Ipv4», RFC3344, Internet Engineering Task Force, August 2002.

[32] W Xing, H Karl, A Wolisz, H Mueller, «M-SCTP: Design and prototypical implementation of an end-to-end mobility concept», Proceedings of 5th Intl. Workshop The Internet Challenge: Technology and Applications, October 2002.

[33] JW Jung, YK Kim, HK Kahng, «SCTP Mobility Highly Coupled with Mobile IP», Telecommunications and Networking - ICT 2004 : 11th International Conference on Telecommunications, August 2004.

[34] Seok Joo Koh, Moon Jeong Chang and Meejeong Lee, «mSCTP for Soft Handover in Transport Layer», IEEE Communication Letters, vol.8, no. 3, March 2004.

[35] I Aydin, C Shen, «Cellular SCTP: A Transport-Layer Approach to Internet Mobility», IEEE proceedings the 12th International Conference on Computer Communications and Networks, ICCCN 2003, October 2003.

[36] M Chang, M Lee, S Koh, «A Transport Layer Mobility Support Mechanism», Networking Technologies for Broadband and Mobile Networks International Conference ICOIN 2004, February 2004.

[37] Technical Report, International Organization for Standardization. Quality Management and Quality Assurance Vocabulary. ISO8402 (2000).

[38] Technical Report. International Telecommunication Union (ITU-T-Rec. E.800). Terms and Definitions Related to Quality of Service and Network Performance Including Dependability. 1993.

[39] E. Crawley, R. Nair, B. Rajagopalan and H. Sandick. A Framework for QoS-based Routing in the Internet. RFC 2386. Août 1998.

[40] W. C. Hardy. QoS Measurements and Evaluation of Telecommunication Quality of Service. 2001.

[41] Technical Report. International Telecommunication Union (ITU-T-Rec. G.1010). End-user Multimedia QoS Categories. 2001.

[42] Armando L. Caro Jr. «SCTP patch of ns-2 simulator», NS-2 SCTP module release 3.3, Protocol Engineering Laboratory (PEL), 2003.

[43] Kevin Fall, Kannan Varadhan, «The *ns* Manual (formerly *ns* Notes and Documentation», VINT Project, December 2003.