Architectures pour la mobilité et la qualité de service dans les systèmes satellites DVB-S2/RCS

Baptiste Jacquemin

▶ **To cite this version:**

Baptiste Jacquemin. Architectures pour la mobilité et la qualité de service dans les systèmes satellites DVB-S2/RCS. Informatique [cs]. Université Paul Sabatier - Toulouse III, 2010. Français. NNT : . tel-00509147

HAL Id: tel-00509147
https://theses.hal.science/tel-00509147

Submitted on 10 Aug 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Discipline ou spécialité :

Systèmes Informatiques

Présentée et soutenue par :
Baptiste Jacquemin

le : jeudi 24 juin 2010

Titre :

Architectures pour la Mobilité et la Qualité de Service dans les Systèmes Satellites DVB-S2/RCS

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
LAAS-CNRS

Directeur(s) de Thèse :
Thierry GAYRAUD
Pascal BERTHOU

Rapporteurs :
Francine KRIEF
Jean-Jacques PANSIOT

Autre(s) membre(s) du jury

André-Luc BEYLOT
Olivier ALPHAND
Abdelmadjid BOUABDALLAH
Cédric BAUDOIN (membre invité)

Résumé

Nos travaux de thèse ont pour objectif la conception, la mise en œuvre et l'évaluation d'architectures pour la mobilité et la qualité de service (QoS) dans des systèmes satellites DVB-S2/RCS. Ces systèmes peuvent constituer une solution alternative efficace aux réseaux terrestres dans des zones reculées à faible densité de population mais ils doivent pour cela offrir les mêmes services tout en tenant compte de leurs caractéristiques spécifiques, en particulier leur long délai de transmission qui peut s'avérer problématique dans le cadre d'applications multimédias interactives.

Notre première contribution a donc été de développer une architecture de QoS adaptée à ce type d'applications, utilisant le modèle DiffServ et se basant essentiellement sur l'interaction entre l'architecture liée au protocole d'initiation de session SIP et différentes entités du système satellite. La QoS peut alors être configurée de façon précise au niveau des STs, par le biais de l'outil TC, en analysant les descripteurs de session SDP compris dans les messages SIP et en déduisant leurs caractéristiques (débit, gigue max, délai max, etc…) soit localement si elles sont connues, soit à partir d'un service Web que nous avons développé.

Nous avons ensuite proposé et développé une solution de mobilité basée sur SIP, adaptée au système satellite ainsi qu'à la solution de QoS précédemment décrite. Les performances de cette solution ont alors été comparées, en termes de temps d'interruption et de consommation de ressources, avec celles obtenues par Mobile IPv6 et certaines de ses extensions, démontrant ainsi de réelles améliorations pour le cas des applications multimédias interactives.

Enfin, notre dernière contribution a été de développer deux architectures couplant QoS et mobilité, une spécifiquement conçue pour les applications interactives et basée sur la combinaison de notre solution de mobilité SIP avec notre architecture de QoS SIP et une autre basée sur Mobile IPv6 ou FMIPv6 et sur l'interaction d'un QoS Agent mobile avec les entités de QoS du système satellite. Ces architectures ont été évaluées et comparées sur la plateforme d'émulation PLATINE développée dans le cadre du projet SATSIX.

Mots-clés : Système Satellite DVB-S2/RCS, Mobilité, Qualité de Service (QoS), SIP, IPv6

Abstract

Our thesis work aims at the design, the implementation and the evaluation of architectures for mobility and quality of service (QoS) in DVB-S2/RCS satellite systems. These systems can be an effective alternative to terrestrial networks in remote and sparsely populated areas but, for that, they have to offer the same services while taking into account their specific characteristics, particularly their long transmission delay that can be problematic in the context of interactive multimedia applications.

Our first contribution has been to develop a QoS architecture adapted to such applications, using the DiffServ model and relying heavily on the interaction between the architecture related to the Session Initiation Protocol (SIP) and various entities of the satellite system. The QoS of satellite terminals (STs) can then be configured precisely, by using the TC tool and analyzing the SDP session descriptors included in the SIP messages and deducting their characteristics (throughput, jitter max, delay max, etc. ...) either locally, if they are known, or from a Web service that we have developed.

We then proposed and developed a mobility solution based on SIP, adapted to the satellite system and to the QoS solution described above. The performances of this solution were compared in terms of handover time and resources consumption, with those obtained by Mobile IPv6 and some of its extensions, showing real improvements in the case of interactive multimedia applications.

Finally, our last contribution was to develop two architectures combining QoS and mobility: the first one is specifically designed for interactive applications and based on the combination of our SIP-based mobility solution with our SIP QoS architecture and the another is based on Mobile IPv6 or FMIPv6 for the mobility part and on the interaction of a mobile QoS agent with QoS entities of the satellite system. These architectures have been evaluated and compared on the emulation platform PLATINE developed under the project SATSIX.

Keywords : DVB-S2/RCS satellite system, Mobility, Quality of Service (QoS), SIP, IPv6

Remerciements

Les travaux présentés dans ce mémoire ont été réalisés dans le cadre d'une collaboration entre le Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique (LAAS-CNRS) et Thales Alenia Space au travers du projet européen SATSIX.

Ils ont été effectués au sein du groupe OLC (Outils et Logiciels pour la Communication) du LAAS-CNRS. Je tiens donc tout d'abord à exprimer mes remerciements aux directeurs successifs du LAAS-CNRS, Messieurs Malik Ghallab et Raja Chatila ainsi qu'au responsable du groupe OLC, Mr Francois Vernadat, pour le cadre de travail qu'ils m'ont offert.

Je tiens ensuite à remercier Mr André-Luc Beylot qui a accepté d'assumer la fonction de président de jury de ma thèse, alors qu'il m'avait déjà supporté trois années pendant mes études d'ingénieur. Je remercie aussi les professeurs Francine Krief et Jean-Jacques Pansiot qui ont accepté de rapporter mon mémoire et m'ont ainsi fait profiter de leurs remarques pertinentes sur mon travail.

Je remercie également Thierry Gayraud et Pascal Berthou, mes directeurs de thèse et de stage de fin d'études d'ingénieur. Ils m'ont offert l'opportunité de découvrir le monde de la recherche, tout en restant au contact de l'industrie par le biais de Thales Alenia Space et m'ont témoigné une grande confiance tout au long de mes travaux en me laissant une grande autonomie. Merci à Thierry qui m'a fait profiter de son expérience et de ses critiques avisées pour me permettre de mieux mettre en valeur mes travaux de thèse et merci à Pakal pour m'avoir aidé à résoudre un certain nombre de problème grâce à sa fameuse méthode d' « extrem programming » (entre autres).

Merci aussi à Cédric Baudoin, mon encadrant de Thales Alenia Space, pour les conseils qu'il m'a apportés tout au long de la thèse, pour avoir relu mon manuscrit et participé à mon jury de thèse et aussi, pour sa bonne humeur permanente.

Je souhaite aussi remercier Olivier Alphand qui m'a transmis ses connaissances sur les systèmes satellites et qui a bien voulu participer au jury de ma thèse. Merci aussi pour toutes les discussions que nous avons eu sur le proxy SIP aussi bien sur les plages de Corse que sur les pistes des Alpes.

Un grand merci aussi à Fréderic Nivor (Fred). Travailler dans son bureau a été une expérience très enrichissante tant au niveau professionnel que personnel. Nos longues discussions notamment sur la vie et les femmes m'ont apporté un point de vue différent mais non moins intéressant ; en tout cas, comme il le dit si bien : « vivons !».

Je remercie aussi les membres du groupe OLC, Roxana, Ihsane, Momo, Lionel, Akram, Akhmed, Ion, Johan, Florin, Guillaume, Pedro, Rasha, Yann, Philippe et tous les autres. Ca a été vraiment agréable de travailler, déjeuner, discuter, …, avec vous. Merci aussi à tous les stagiaires avec qui j'ai partagé de bon moment autour de la machine à café. Un grand, grand merci aussi à la « Nano team » du LAAS qui est vraiment super sympa : Toto, Cédric, Laurent, Sabrina, Hélène, Thierry, Sami, Florent, Sven, et tous les autres avec qui j'ai partagé de très bon moments à l'intérieur et à l'extérieur du LAAS. Un merci tout particulier à Toto

pour m'avoir fait rencontrer tous ces gens supers mais surtout merci à lui d'être comme il est, d'avoir refait et encore refait le monde avec moi au labo, pendant nos retours en vélo ou ailleurs, d'avoir partagé des moments inoubliables en Italie et partout en France et tout simplement d'être un de mes amis les plus chers.

Merci aussi à mes compagnons de tests réseaux, Thierry, Pakal, Momo, i2, Fred, Florin et Rado pour les moments de détente partagés pendant la digestion ou le gouter.

Je remercie aussi les membres des différents services techniques et administratifs du LAAS-CNRS, qui m'ont permis de travailler dans d'excellentes conditions.

Je voudrais maintenant remercier ma famille sans qui je n'aurais jamais réussi à arriver jusque là. Merci à mes parents qui ont toujours été là quand j'en avais besoin et qui m'ont offert un cadre de vie de rêve. Merci à mes frères et sœurs, qui sont aussi mes amis les plus chers, avec qui j'ai partagé des moments inoubliables. Et merci à mes grands parents, oncles, tantes, cousins et cousines pour tous les moments que nous avons partagé ensemble. Avoir une famille sympa et soudée a toujours été très important pour moi et j'en suis comblé.

Enfin, je tiens bien sûr à remercier mes amis qui m'ont supporté depuis tant d'années et qui ont toujours été là pour profiter avec moi de la vie en dehors du laboratoire. Merci aux Strasbourgeois, Cousin le grand sage mais aussi le grand dadais, Ptonj l'éternel sans-âme qui nous a tant fait voyager, Alexe l'artiste, toujours aussi généreuse et Rocco le charmeur-poète de m'avoir accueilli au sein de leur communauté alsacienne. Ca a été vraiment mythique de vous rencontrer et qu'on participe tous en même temps à l'exode vers le grand Sud-Ouest. Merci aux potes de Toulouse pour toute ces belles années qu'on a passé ensemble : Ketchouille la fripouille et toute sa bande de bourbonnais ainsi que Laurette la pipelette qui nous ont accueilli et supporté chez eux tant de fois avec bonne humeur, Marianne et Trapos les surfeurs-musicos, Francis le joyeux drille-rugbyman, Giac'omo le lover-mafioso, Steph la meilleure organisatrice de weekend/soirée et bien sûr de pétanque, Tonton, Téton, Jéjé et Nono, les irréductibles loulous-fêtards, Merci aussi à la Comtesse, à Giovani et sa bande à Castel, à Gougou et sa secte du CPP, à Macmanamiche et Eric pour toutes ses sessions musicales au bord de la Garonne, à Boulbacloak le caméraman-footballeur et dédicace toute particulière aux rototomiens pour ses moments paradisiaques qu'on a passé ensemble à Osoppo. Merci à mes amis du Berry, Boubouche mon ami de toujours, Greg, Camille, Marie, Hélène et tous les autres. Malgré la distance, notre amitié est restée intacte et c'est très important pour moi de pouvoir continuer à partager des bons moments avec vous. Enfin, merci à tous mes autres amis éparpillés en France et dans le monde.

Pour finir en beauté, un énorme merci à Chloé avec qui je partage ma vie. Elle m'a aidé et supporté pendant toutes ces années dans les moments difficiles malgré mon caractère imprévisible et m'a toujours apporté réconfort et sérénité. Son naturel et sa joie de vivre ont toujours été une bouffée d'air frais en dehors du travail et pour ça, je ne la remercierai jamais assez.

Liste des acronymes

3G	3rd Generation
3GPP	3rd Generation Partnership Project
AAL5	ATM Adaptation Layer 5
ADSL	Asymmetric Digital Subscriber Line
AF	Assured Forwarding
ANG	Access Network Gateway
AP	Access Point
AR	Access Router
ARC	Access Resource Controller
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
BACK	Binding Acknowledgement
BE	Best Effort
BS	Base Station
BSM	Broadband Satellite Multimedia
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
BU	Binding Update
BWA	Broadband Wireless Access
CAC	Connection Admission Control
CAPWAP	Control and Provisioning of Wireless Access Points
CBQ	Class Based Queuing
CD	Critical Data
CID	Connection Identifier
CL	Controlled-Load
CMIP	Client Mobile IP
CN	Correspondant Node
CoA	Care-of Address
COPS	Common Open Policy Service
CoT	Care-of Test
CoTi	Care-of Test Init
CQD	Classful Queuing Discipline
CRA	Continuous Rate Assignment
CTP	CAPWAP Tunneling Protocol
DAD	Duplicate Address Detection
DAMA	Demand Assignment Multiple Access
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DNS	Domain Name System
DS	Distribution System
DSCP	DiffServ Code Point
DSL	Digital Subscriber Line
DSMIPv6	Dual Stack Mobile IPv6
DULM	Data Unit Labelling Method
DVB	Digital Video Broadcasting
DVB-RCS	DVB Return Channel for Satellite
DVB-S	DVB for Satellite
EF	Expedited Forwarding
ER	Edge Router
ESA	European Space Agency
ESP	Encapsulating Security Payload
ESS	Extended Service Set
ESSID	Extended Service Set Identifier
ETSI	European Telecommunications Standards Institute
EuQoS	End-to-end Quality of Service support over heterogeneous networks
FAI	Fournisseur d'Accès Internet
FBACK	Fast BACK

FBSS	Fast BS Switching
FBU	Fast BU
FCA	Free Capacity Assignment
FLAN	FLow ANalyser
FLOC	FLOw Capturer
FLORE	FLOw REplayer
FMIPv6	Mobile IPv6 Fast Handovers
FTP	File Transfer Protocol
GPRS	General Packet Radio System
GS	Guaranteed Service
GSM	Global System for Mobile
GW	Gateway
HA	Home Agent
HACK	Handover Acknowledge
HI	Handover Initiate
HIP	Host Identity Protocol
HIT	Host Identity Tag
HMIPv6	Hierarchical Mobile IPv6
HNP	Home Network Prefix
HO	Handover
HoA	Home Address
HoT	Home Test
HoTi	Home Test Init
HTB	Hierarchical Token Bucket
HTTP	Hyper Text Transfer Protocol
IAPP	Inter Access Point Protocol
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IntServ	Integrated Services
IP	Internet Protocol
IPv4	IP version 4
IPv6	IP version 6
IPsec	IP security protocol
IST	Information Society Technology
ITU	International Telecommunications Union
JTG	Jugi's Traffic Generator
LAN	Local Area Network
LBU	Local Binding Update
LCoA	On-Link CoA
LMA	Local Mobility Anchor
LWAPP	LightWeight Access Point Protocol
MAC	Medium Access Control
MAG	Mobile Access Gateway
MAP	Mobility Anchor Point
MDHO	Macro Diversity HandOver
MF-TDMA	Multi-Frequency Time Division Multiple Access
MGEN	Multi-GENerator
MIH	Media Independent Handover
MN	Mobile Node
MOS	Mean Opinion Score
MPE	Multi Protocol Encapsulation
MPEG	Motion Picture Expert Group
MPLS	Multi-Protocol Label Switching
MS	Mobile Station
mSCTP	mobile SCTP
MSR	Mobile Support Router
MTR	Media Type Repository
NAR	New Access Router

NAT	Network Address Translation
NCC	Network Control Centre
NCoA	Next CoA
NGN	Next Generation Network
NMC	Network Management Center
NRT	Non Real Time
NSIS	Next Steps in Signaling
NSLP	NSIS Signaling Layer Protocol
NTP	Network Time Protocol
OBP	On-Board-Processing
ORENETA	One way delay REal-time NETwork Analyser
OSI	Open System Interconnection
PAR	Previous Access Router
PBA	Proxy Binding Acknowledgement
PBU	Proxy Binding Update
PCoA	Previous CoA
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PHB	Per-Hop Behavior
PKM	Privacy Key Management
PMIP	Proxy Mobile IP
PMP	Point-to-Multipoint
PQ	Priority Queuing
PVC	Permanent Virtual Circuit
QoS	Quality of Service
RA	Router Advertisement
RADVD	Router ADVertisement Daemon
RAP	Resource Allocation Protocol
RBDC	Rate-Based Dynamic Allocation
RCoA	Regional CoA
RCS	Return Channel for Satellite
RCST	Return Channel Satellite Terminal
RESV	RESerVe
RFC	Request for Comments
RO	Route Optimization
RRT	Return Routability Test
RSVP	Resource reSerVation Protocol
RT	Real Time
RTP	Real Time Protocol
SAP	Service Access Point
SAR	Segmentation and Reassembly
SATIP6	Satellite Broadband Multimedia System for IPv6
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SMTP	Simple Mail Transfert Protocol
SE	Satellite Emulator
SIGMA	Seamless IP diversity based Generalized Mobility Architecture
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SLAPP	Secure Light Access Point Protocol
SLS	Service Level Specification
SOAP	Simple Object Access Protocol
SNMP	Simple Network Management Protocol
SP	Service Provider
SS	Subscriber Station
ST	Satellite Terminal
TBTP	Terminal Burst Time Plan
TC	Traffic Control
TCP	Transmission Control Protocol
TFTP	Trivial File Transfert Protocol

TISPAN	Telecommunications and Internet converged Services and Protocols for Advanced Networking
TOS	Type of Service
TS	Transport Stream
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
ULE	Unidirectionnal Lightweight Encapsulation
UMTS	Universal Mobile Telephone Service
UNA	Unsolicited Neighbor Advertisement
URI	Uniform Resource Identifier
UT	User Terminal
VBDC	Volume Based Dynamic Capacity
VoIP	Voice over IP
WFQ	Weighted Fair Queuing
WiCoP	Wireless LAN Control Protocol
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WWAN	Wireless Wide Area Network
XML	eXtensible Markup Language

Liste des figures

Liste des tableaux

Table des matières

Introduction générale

Contexte

Dans les dernières années, de nombreuses technologies sans fil ont fait leur apparition pour permettre aux utilisateurs de l'Internet de rester connectés quel que soit l'endroit où ils se situent. Leur succès croissant vient essentiellement du peu d'infrastructure qu'elles nécessitent ainsi que de leurs caractéristiques, très variées, qui permettent de s'adapter à un grand nombre de situations. Ainsi, les réseaux sans fil personnels (WPAN, *Wireless Personal Area Network*), tels que les technologies Bluetooth ou ZigBee, sont essentiellement destinés à la connexion à faible portée de petits appareils domestiques sans fil tels que téléphone sans fil, PDA, etc. Les réseaux locaux sans fil (WLAN, *Wireless Local Area Network*), quant à eux, permettent de connecter entre eux différents terminaux par voie radio avec des débits de plusieurs dizaines de mégabits par seconde sur une distance pouvant aller jusqu'à plusieurs centaines de mètres. La technologie Wi-Fi en est l'exemple le plus répandu. Viennent ensuite les réseaux métropolitains sans fil (WMAN, *Wireless Metropolitan Area network*), tels que le WiMAX, qui permettent de couvrir des zones allant jusqu'à plusieurs dizaines de kilomètres, à l'échelle d'une ville par exemple, avec des débits de plusieurs dizaines de mégabits par seconde. Enfin, les réseaux étendus sans fil (WWAN, *Wireless Wide Area Network*) permettent de mettre en place des réseaux à l'échelle d'un pays, d'un continent voire même de notre planète. Deux catégories de système correspondent à ce type de réseaux :

- Les technologies GSM, GPRS, UMTS, 3G etc., plus communément appelées réseaux cellulaires de télécommunications mobiles, qui représentent en fait les réseaux sans fil les plus répandus, chaque téléphone mobile étant connecté à un réseau de ce type. Ces technologies, basées sur la notion de cellules se chevauchant pour couvrir une zone géographique donnée, offrent des débits pouvant aller de quelques kilooctets à plusieurs mégaoctets par seconde.
- Les réseaux satellites qui offrent une couverture très large (jusqu'à un tiers du globe pour les satellites géostationnaires). D'abord utilisés essentiellement comme support de diffusion de la télévision, ces réseaux ont évolués et permettent actuellement un accès bidirectionnel aux services IP classiques mais souffrent principalement de longs délais de propagation.

En constatant ce déploiement croissant de technologies sans fil, on est en droit de penser que les réseaux de nouvelle génération (NGN, *Next Generation Networks*) constitueront une convergence de toutes ces technologies avec le monde du filaire. Ainsi, les futurs équipements mobiles seront équipés de multiples interfaces réseaux, ils devront pouvoir accéder à tous les services depuis n'importe quelle technologie et passer de l'une à l'autre de façon transparente pour l'utilisateur et cela, même en cours de communication.

Cependant, offrir une telle mobilité nécessite des mécanismes complexes qui doivent s'adapter aux différentes caractéristiques des technologies concernées ainsi qu'aux services demandés. Ainsi, maintenir une communication de voix sur IP (VoIP, *Voice over IP*) lorsqu'un utilisateur passe d'un réseau Wi-Fi à un réseau satellite ne nécessitera pas les mêmes exigences que de maintenir un téléchargement FTP lors d'un passage d'un réseau Wi-

Fi au réseau 3G. Effectivement, en plus de devoir gérer la mobilité, une des nouveautés de l'Internet est qu'il doit maintenant transmettre des données interactives (VoIP, vidéoconférence, jeux interactifs, etc.) ayant de fortes contraintes temporelles au niveau du délai ou de la bande passante. Ceci a donc amené la communauté scientifique à définir la notion de qualité de service (QoS, *Quality of Service*) mais Internet n'a été conçu ni dans le but de mettre en place différentes classes de service, ni dans celui de gérer des nœuds mobiles.

C'est dans ce contexte que nos travaux de thèse apportent des éléments pour mettre en place une architecture permettant à un utilisateur mobile de conserver ses communications en cours lors de changements de réseau tout en gardant une qualité de service adaptée aux besoins de ses applications. Nous nous intéresserons plus particulièrement au cas de réseaux de communication impliquant un système satellite géostationnaire bidirectionnel de type DVB-S2/RCS et nous essaierons avant tout de mettre en œuvre dans ce type de réseau une qualité de service pour les applications ayant de fortes contraintes temporelles telles que la VoIP et la vidéoconférence dont l'utilisation est de plus en plus répandue actuellement.

Problématique

Pendant longtemps, les systèmes de télécommunication par satellite, essentiellement dédiés aux services de diffusion, ont souffert de leur caractère propriétaire qui rendait difficile et coûteux leur interopérabilité ainsi que leur intégration à l'Internet.

Cependant, la récente démocratisation des terminaux DVB-S, puis DVB-S2, la standardisation d'une voie de retour par satellite avec la norme DVB-RCS et les efforts de la communauté satellite en terme d'interopérabilité ont permis à ces systèmes de se positionner comme solution alternative pour les zones géographiques à couverture difficile ou nulle.

Parallèlement, les innovations réalisées dans le domaine des applications interactives telles que l'apparition de la VoIP ou de la visioconférence et l'important développement du sans fil ont fait apparaître de nouveaux besoins chez l'utilisateur en terme de mobilité et de QoS.

Dans ce contexte, si les systèmes satellites géostationnaires veulent réellement devenir compétitifs, ils doivent permettre à leurs utilisateurs d'accéder à ces services malgré les inconvénients majeurs dont ils souffrent : des délais de transmission très longs et une faible bande passante. Les mécanismes de mobilité et de QoS mis en place doivent donc tenir compte de ces contraintes tout en restant compatible avec les solutions « terrestres ». Par ailleurs, ces solutions doivent être cohérentes avec les mécanismes actuellement développés dans le cadre des architectures NGN pour rendre possible leur future intégration.

Organisation du mémoire

Ce mémoire est organisé de la façon suivante.

Le premier chapitre présente la notion de mobilité de façon générale, sans être spécifique aux systèmes satellites. Nous préciserons tout d'abord la terminologie employée dans le domaine de la mobilité pour définir clairement dans quel cadre nous nous positionnons dans

la suite du mémoire. Ensuite, nous établirons un état de l'art des différentes solutions proposées pour la gestion de la mobilité en parcourant les différentes couches.

Le deuxième chapitre est consacré essentiellement à la notion de QoS. Dans un premier temps, nous introduirons les différentes métriques liées à la QoS ainsi que les besoins associés aux principaux types d'applications actuellement existantes. Ensuite, nous présenterons les modèles précurseurs de gestion de la QoS ainsi qu'une sélection de protocoles (COPS, SIP, NSIS) pouvant participer à ce processus. Enfin, après avoir brièvement décrit la notion de réseau de nouvelle génération (NGN), nous parcourerons les différentes propositions faites pour la gestion couplée de la mobilité et de la QoS.

Dans le troisième chapitre, après avoir rappelé l'architecture d'un système satellite DVB-S2/RCS, nous décrirons tout d'abord le projet dans le cadre duquel la majeure partie de nos travaux a été réalisée : le projet SATSIX. Nous présenterons alors les mécanismes de QoS spécifiques aux réseaux DVB-S2/RCS pour ensuite spécifier les outils que nous avons participés à mettre en œuvre pour la partie QoS de notre architecture. Après cela, nous détaillerons nos choix concernant la mise en œuvre d'une mobilité SIP adaptée au système satellite et comparerons ses performances théoriques dans un système satellite à celles des protocoles de mobilité basés sur Mobile IPv6 présentées dans le premier chapitre. Cela nous permettra d'établir des premières conclusions et recommandations quant à leur utilisation dans des réseaux de type DVB-S2/RCS. Finalement, sur la base des spécifications de QoS et de mobilité réalisées auparavant, nous présenterons nos propositions d'architectures complètes permettant la gestion couplée de la QoS et de la mobilité.

Le chapitre 4 présentera finalement les différentes implémentations et expérimentations réalisées au cours du doctorat permettant de mesurer les performances de nos architectures de QoS et de mobilité. Pour cela, nous expliciterons tout d'abord la méthode et les outils utilisés et développés pour nos évaluations puis présenterons les résultats obtenus sur la plateforme d'émulation que nous avons participé à mettre en œuvre pour le projet SATSIX.

En conclusion générale de ce mémoire, nous rappellerons les principales contributions réalisées au cours de nos travaux et énonçerons leurs principales perspectives.

I. La Mobilité dans les Réseaux de Communication

Le terme de mobilité peut être défini comme la capacité d'un utilisateur d'accéder, quelle que soit sa localisation, à l'ensemble des services auxquels il a accès habituellement en environnement fixe et câblé. Cette définition générique traduit l'idée principale de mobilité mais englobe un grand nombre de concepts qu'il convient de préciser.

Ainsi, pour éviter certaines confusions, la première partie de ce chapitre est donc consacrée à présenter les différents concepts qu'englobe le terme de mobilité, ce qui nous permet de positionner précisément notre étude pour la suite du mémoire. La deuxième partie de ce chapitre présente ensuite un état de l'art des solutions de gestion de la mobilité de terminal.

I.1. Terminologie de la mobilité

Dans cette partie, nous nous efforçons de différencier les principaux types de mobilité tels qu'ils ont été majoritairement définis au cours des dernières années. Pour les trois premiers types de mobilité, la mobilité personnelle, la mobilité de session et la mobilité de service, nous nous appuierons en particulier sur les définitions données dans [1].

I.1.1. La mobilité personnelle

La notion de mobilité personnelle décrit la possibilité de joindre un utilisateur par le même identifiant logique où qu'il soit, quel que soit l'appareil de communication ou terminal (PC, laptop, téléphone portable, etc.) qu'il utilise et cela indépendamment de la technologie d'accès (GSM, Wi-Fi, Ethernet, etc.). Ainsi, un même identifiant peut être associé à différents terminaux et plusieurs identifiants peuvent être associées au même terminal comme le montre la Figure 1.

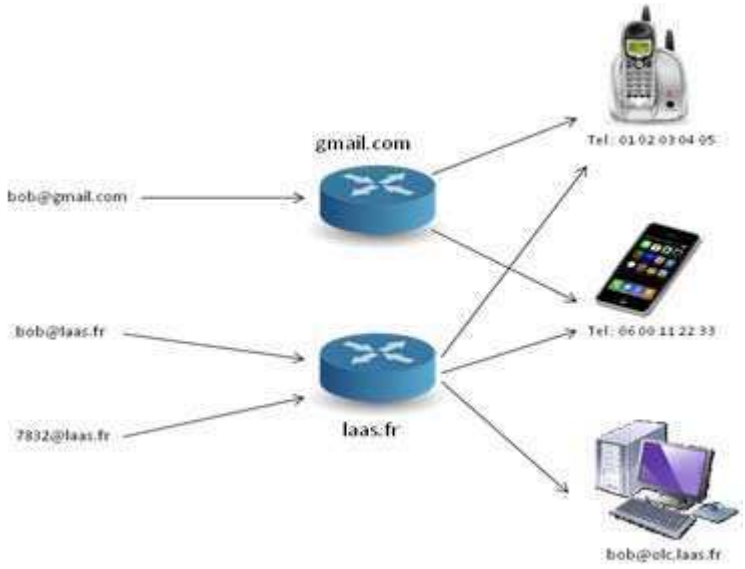


Figure 1 – Exemple de mobilité personnelle

Dans cet exemple, Bob peut être joint depuis l'extérieur sur n'importe lequel de ces terminaux via différents identifiants (adresse mail par exemple) et cela de façon transparente pour le correspondant. Il peut ainsi décider que son identifiant professionnel (bob@laas.fr ou 7832@laas.fr) soit redirigé vers son téléphone de maison mais aussi que son identifiant privé ne soit redirigé que vers son téléphone de maison et son smartphone.

I.1.2. La mobilité de session

La mobilité de session doit permettre à un utilisateur de maintenir ses sessions actives tout en changeant de terminal. Ainsi, un utilisateur ayant une communication de VoIP sur son smartphone peut, en arrivant à son bureau de travail, choisir de continuer sa conversation sur son PC sans qu'il y ait rupture de communication.

I.1.3. La mobilité de service

La mobilité de service doit permettre à un utilisateur d'accéder aux services auxquels il a souscrit auprès de son fournisseur d'accès où qu'il se situe (éventuellement chez un autre fournisseur d'accès) et quel que soit le type de terminal ou de technologie qu'il utilise. Ainsi, en prenant l'exemple de la VoIP, l'utilisateur, où qu'il se trouve, aura accès à la liste de ses contacts, au blocage d'appel, aux préférences de médias et plus généralement à toutes les options souscrites pour ce service.

I.1.4. La mobilité de terminal

La mobilité de terminal doit permettre à un utilisateur de maintenir ses sessions actives et de rester joignable depuis l'extérieur tout en changeant de réseau ou de sous réseau IP (et donc de point d'accès ou point de rattachement). Dans ce cas là, l'utilisateur ne change pas de terminal au cours de sa session. Différentes terminologies sont alors utilisées pour désigner les sous-catégories de ce type de mobilité. Tout d'abord, on peut distinguer:

- La mobilité **horizontale** ou **intra-technologie**, dans laquelle un utilisateur se déplace de cellule en cellule à l'intérieur d'une même technologie d'accès. C'est le cas, par exemple, d'un utilisateur qui se déplace à l'intérieur du réseau Wi-Fi de son entreprise. Dans ce cas, la mobilité est généralement gérée par le système Wi-Fi en lui-même par l'intermédiaire de messages échangés par les différentes bornes d'accès qui le composent.
- La mobilité **verticale** ou **inter-technologie**, dans laquelle un utilisateur va changer de technologie d'accès. Il peut ainsi passer du GSM au Wi-Fi, au WiMAX, au satellite ou encore au Bluetooth, etc. Ce changement peut intervenir lorsque l'utilisateur se déplace (on peut alors parler de mobilité diagonale car l'utilisateur change à la fois de technologie et de cellule) mais aussi pour des raisons de qualité de service ou de sécurité par exemple.

Ces deux définitions se révèlent assez vagues ; en effet, il est difficile de dire dans quelle catégorie placer un cas de mobilité entre un point d'accès 802.11b et un autre 802.11e ou encore un cas de mobilité entre un point d'accès 802.11b public (d'une ville par exemple) et un autre 802.11b privé (de sa maison par exemple).

Ensuite, on peut séparer deux autres catégories de mobilité de terminal qui dépendent du fait que l'utilisateur change ou pas de domaine :

- La mobilité **intra-domaine**, dans laquelle un utilisateur se déplace à l'intérieur d'un même domaine administratif, c'est-à-dire un domaine sous la responsabilité d'une seule et même autorité. Typiquement, un opérateur constitue un domaine administratif. Ce domaine peut être constitué ou non de plusieurs technologies.
- La mobilité **inter-domaine** qui intervient lorsque l'utilisateur change de domaine administratif au cours de ses déplacements.

Ces termes peuvent par exemple être adaptés dans le cas de différents opérateurs voulant s'accorder sur les politiques à mettre en place lorsqu'un utilisateur se déplace entre deux réseaux d'accès appartenant à deux opérateurs différents.

D'autres termes sont aussi utilisés pour définir différents niveaux de mobilité. Ainsi, en se basant sur l'architecture présentée par la Figure 2 comprenant des passerelles de réseau d'accès (ANG, *Access Network Gateway*), des routeurs d'accès (AR, *Access Router*) et des points d'accès (AP, *Access Point*), l'IETF a définit, au travers de [2] puis [3], trois catégories distinctes de mobilité qu'un nœud mobile (MN, *Mobile Node*) peut réaliser:

- La **mobilité intra-lien** ou **mobilité de niveau 2** qui définit une mobilité entre deux points d'accès sans fil d'un même réseau d'accès. Typiquement, ce cas de mobilité n'implique que des mécanismes de niveau 2 ou du moins, aucune reconfiguration de réseau ou sous réseau IP n'est nécessaire. Cependant, de la signalisation de niveau réseau ou supérieur peut être requise pour les échanges de messages entre points d'accès. Ce type de mobilité est en fait une sous-catégorie de la mobilité horizontale ou intra-technologie.
- La **micro-mobilité** ou **mobilité locale** qui désigne une mobilité à l'intérieur d'un même réseau d'accès mais qui implique des mécanismes de reconfiguration de réseau ou sous réseau IP. Cependant, bien que les termes « micro » ou « local » soient utilisés, cela n'implique pas que la zone géographique couverte par le réseau d'accès en lui-même ne puisse pas être importante. On peut alors parler de gestion localisée de la mobilité lorsque la signalisation permettant de maintenir la connectivité IP est restreinte au réseau d'accès.
- La **macro-mobilité** ou **mobilité globale** qui désigne une mobilité entre différents réseaux d'accès impliquant une reconfiguration IP, sans tenir compte du type de technologie. La gestion de la mobilité ne peut pas se faire localement et peut impliquer d'importants changements au niveau du routage des paquets de bout-en-bout.

Ce dernier type de différenciation de la mobilité semble le plus approprié pour la suite de nos travaux. En effet, nous nous intéressons plus particulièrement à savoir si un changement d'adresse IP est nécessaire lors d'un épisode de mobilité, ce qu'il est difficile de distinguer dans le cas horizontal/vertical et intra-domaine/inter-domaine. En effet, dans les cas horizontal et intra-domaine, un changement d'adresse IP peut être nécessaire ou non, selon les cas pris en considération.

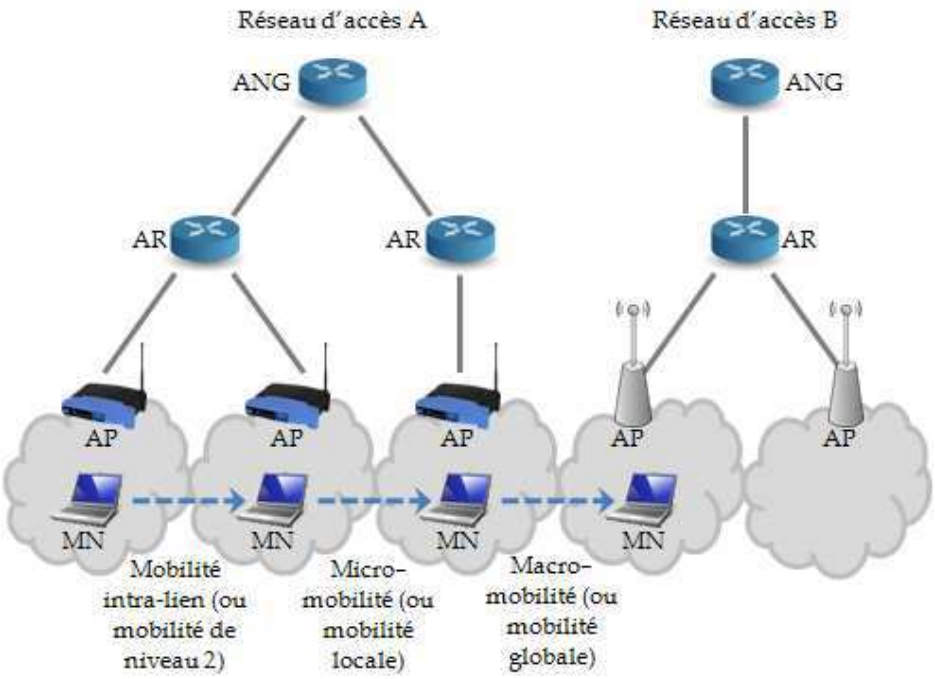


Figure 2 – Terminologie IETF de la mobilité

Enfin, la mobilité de terminal peut être divisée selon sa « granularité » ; se distinguent alors trois catégories :

- La mobilité **discrète** ou « **nomade** » qui désigne un utilisateur qui se déplace et donc qui change de point de rattachement sans avoir de communication en cours.
- La mobilité **continue** qui désigne un utilisateur qui se déplace en ayant des communications en cours. Il reste joignable durant son déplacement mais ses communications peuvent subir une interruption due aux différentes étapes de rattachement au nouveau point d'accès, d'attribution d'une nouvelle adresse IP, etc.
- La mobilité **sans interruption** ou « **sans couture** » qui désigne un utilisateur qui se déplace en ayant des communications en cours sans que celles-ci ne subissent la moindre interruption.

I.1.5. La mobilité de réseau

On parle de mobilité de réseau lorsque un ensemble de réseaux ou sous réseaux connectés à l'Internet par l'intermédiaire d'un ou plusieurs routeurs mobiles changent leur point de rattachement à l'Internet. Ce type de mobilité peut par exemple intervenir lorsque un réseau, mis en place à l'intérieur d'un véhicule (train, bateau, voiture, bus, avion, etc.) souhaite accéder aux services IP en cours de déplacement. Ces réseaux embarqués à bord de véhicules peuvent être de différentes natures : par exemple, un réseau de capteurs déployé dans un bateau qui échange des données nécessaires à la navigation, ou encore un réseau d'accès déployé à l'intérieur d'un train pour permettre aux usagers de se connecter lors de leur trajet.

I.1.6. Positionnement de notre étude sur la mobilité

Nous avons donc pu constater que le terme « mobilité » couvre un grand nombre de significations ce qui en fait un domaine complexe. Ainsi, et vu que notre thèse englobe d'autres thématiques importantes que sont la QoS et les réseaux satellites, dans la suite du mémoire, nous n'étudierons pas tous les types de mobilité et nous nous focaliserons sur la mobilité de terminal qui nous semble être la plus importante. En effet, c'est le type de mobilité qui fait actuellement l'objet du plus grand nombre de travaux et qui soulève le plus de questions. Cela étant dit, cela ne nous empêchera pas, lorsque l'occasion s'y prêtera, d'expliciter comment telle ou telle solution pourrait aussi permettre la gestion d'un autre type de mobilité. Mais, si aucune précision n'est faite, le terme mobilité désignera, dans la suite de ce mémoire, la **mobilité de terminal**. De plus, nous préférerons utiliser la distinction {mobilité de niveau 2/micro-mobilité ou mobilité locale/macro-mobilité ou mobilité globale} plutôt que d'utiliser la distinction horizontale/verticale.

I.2. Les protocoles de gestion de la mobilité

Dans cette partie, nous allons présenter un état de l'art des solutions de gestion de la mobilité en parcourant les différentes couches du modèle présenté en Figure 3.

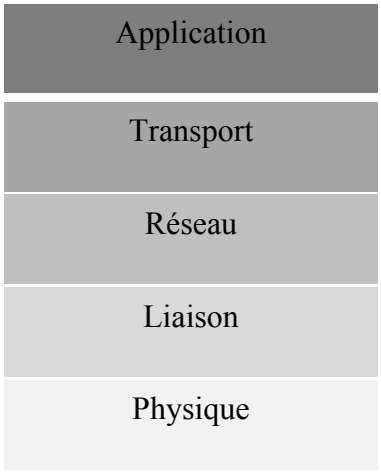


Figure 3 – Modèle en couches

Le modèle proposé est un modèle hybride entre le modèle OSI (*Open System Interconnection*) [4] et le modèle TCP/IP dans le sens où il reprend les 4 couches basses du modèle OSI, mais considère comme dans le modèle TCP/IP que la couche application rassemble les fonctionnalités des trois couches supérieures du modèle OSI, à savoir des couches session, présentation et application.

Dans cette partie, étant donné la quantité de solutions qui ont été proposées pour gérer la mobilité d'un terminal, nous nous appliquerons à présenter les solutions les plus représentatives de leur couche respective ou qui ont fait l'objet du plus grand nombre de recherches et nous ne présenterons pas les solutions de téléphonie mobile type GSM, GPRS, etc.

I.2.1. La gestion de la mobilité dans les technologies de niveau 1 et 2

Dans le cadre de notre étude, nous présentons dans cette partie les technologies IEEE 802.11 et IEEE 802.16 qui définissent toutes les deux des couches physique et liaison spécifiques. Le but de notre étude n'étant pas d'expliquer la nature de ces couches, nous nous attachons plus particulièrement à comprendre les mécanismes mis au point par ces technologies pour permettre la gestion de la mobilité, c'est-à-dire du mécanisme qui permet de se déplacer d'une cellule à une autre sans que la communication soit coupée. De plus, dans cette partie, nous n'analysons précisément que le cas dans lequel ces technologies sont basées sur un élément central : la station de base (BS, *Base Station*) ou point d'accès (AP, *Access Point*) auquel un mobile doit s'associer pour pouvoir communiquer. Ces points d'accès sont en charge de la couverture d'une cellule et sont reliés à l'Internet par voie filaire (xDSL par exemple) ou sans fil (technologie satellite par exemple).

I.2.1.1. La technologie IEEE 802.11 (Wi-Fi)

Le terme IEEE 802.11 [5] désigne un ensemble de normes concernant les réseaux locaux sans fil qui ont été mises au point par le groupe de travail 11 du comité de normalisation LAN/MAN de l'IEEE (IEEE 802). Aussi appelée Wi-Fi par abus de langage, car le terme Wi-Fi correspond en fait à une certification délivrée par la WECA (*Wireless Ethernet Compatibility Alliance),* cette norme permet à différents terminaux de communiquer en haut débit (jusqu'à 54 Mbps pour le 802.11g [6] et même jusqu'à plusieurs centaines de Mbps pour le 802.11n [7]) sur un rayon de plusieurs dizaines de mètres en intérieur et jusqu'à plusieurs centaines de mètres en extérieur. D'après le standard IEEE 802.11, deux modes d'utilisation sont définis :

- le mode Ad-Hoc dans lequel les clients sont connectés les uns aux autres sans point d'accès. Chaque terminal joue alors à la fois le rôle de client et de point d'accès.
- le mode infrastructure dans lequel les clients sont associés à un point d'accès.

Comme nous l'avons précisé précédemment, nous ne nous intéresserons ici qu'au mode infrastructure, dans lequel les terminaux mobiles, lorsqu'ils changent de cellule, peuvent continuer à communiquer au travers des points d'accès. La Figure 4 présente l'architecture d'un réseau IEEE 802.11 en mode infrastructure.

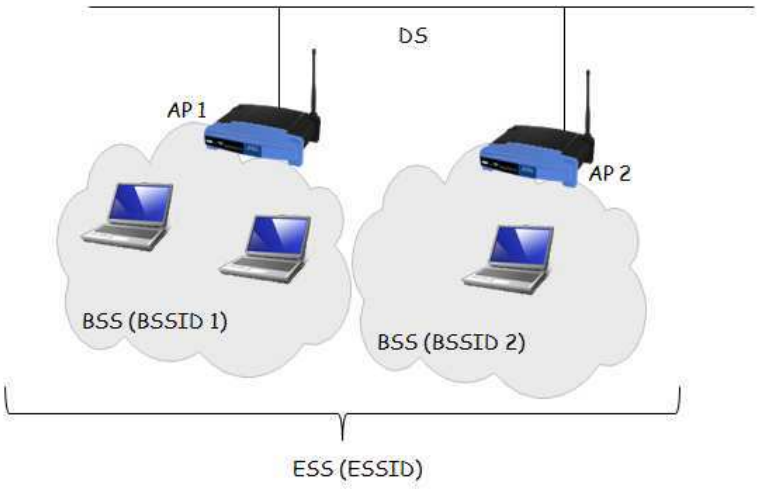


Figure 4 – Architecture d'un réseau 802.11 en mode infrastructure

L'ensemble des terminaux à portée d'un même point d'accès forment un ensemble de service de base (BSS, *Basic Service Set*) et communiquent via ce point d'accès. Ce BSS, qui correspond à une cellule, a pour identifiant un BSSID (*BSS Identifier*) qui correspond à l'adresse MAC de son point d'accès. Un réseau IEEE 802.11 peut alors être composé de plusieurs BSS pour former un ensemble de service étendu (ESS, *Extended Service Set*). Les différents BSS sont reliés entre eux par un système de distribution (DS) qui relie leur point d'accès respectif. Ce DS peut être un réseau filaire, un simple câble ou encore une liaison sans fil. Un ESS est alors identifié par son ESSID (*ESS Identifier*) souvent abrégé en SSID qui représente en fait le nom du réseau IEEE 802.11.

I.2.1.1.a Gestion des associations dans IEEE 802.11

Lorsqu'un terminal IEEE 802.11 se trouve à portée d'un ou plusieurs points d'accès, il va en sélectionner un en fonction de caractéristiques telles que la puissance du signal. Le processus d'association se déroule selon différentes étapes récapitulées sur la Figure 5 :

- **L'écoute du support** qui peut être réalisée de deux façons :
  - L'écoute active dans laquelle le terminal a déjà, par le passé, été configuré pour se connecter à un réseau sans fil et donc envoie sur chaque canal une requête de type Probe Request contenant des informations telles que son ESSID, les débits qu'il supporte, etc. Si l'ESSID correspond à celui d'un ou plusieurs des points d'accès qui sont à portée, ils répondent par une requête de type Probe Response en donnant leurs caractéristiques (débit, charge, etc.). La station peut alors choisir son point d'accès. Si aucune réponse n'est reçue, le terminal passe en mode d'écoute passive.
  - L'écoute passive dans laquelle le terminal scanne tous les canaux et attend de recevoir une trame balise (ou *beacon*) que les points d'accès envoient environ toutes les 0.1s.
- **L'authentification** qui peut aussi être réalisée de deux manières différentes :
  - L'*Open System Authentication* qui est le mode par défaut mais dans lequel il n'y pas réelle authentification puisque tous les terminaux se connectant sont authentifiés.
  - La *Shared Key Authentication* dont l'authentification est basée sur le partage d'une clef secrète entre le terminal et le point d'accès. Si le terminal n'utilise pas la même clef que le point d'accès, son authentification sera rejetée.
- **L'association** en elle-même, dans laquelle le terminal envoie une requête de type *Association Request* à laquelle le point d'accès va répondre par une requête de type *Association Response*. Lors de cet échange, les deux entités négocient les différents paramètres qui seront utilisés pour la communication.

Une fois cette association réalisée, le terminal continue à écouter les différents canaux et peut éventuellement changer de point d'accès si ses performances (qualité du signal, type de réseau IEEE 802.11, etc…) sont meilleures.

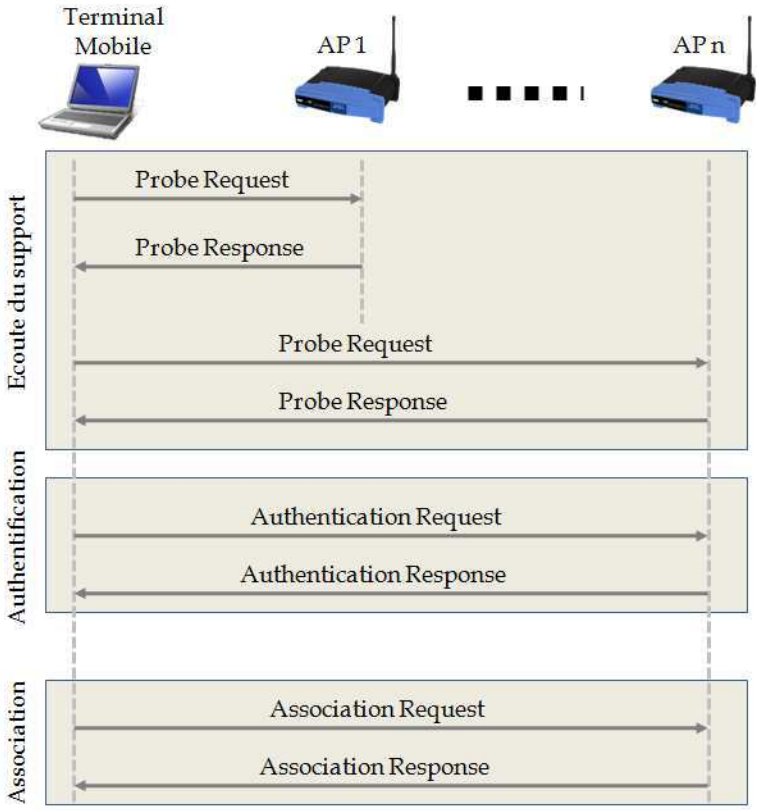


Figure 5 – Processus d'association en mode d'écoute active dans un réseau IEEE 802.11

I.2.1.1.b Les prémices de l'itinérance (ou roaming) dans IEEE 802.11

La notion d'itinérance ou de changement de point d'accès sans perte de connectivité réseau, correspondant à une mobilité de niveau 2, n'a pas été initialement prévue dans la norme IEEE 802.11. Ce n'est qu'en 2003, sur la base d'un protocole propriétaire nommé IAPP (*Inter Access Point Protocol*) que l'IEEE a aboutit à la rédaction d'un « trial and use » qui a donné naissance à la norme IEEE 802.11f [8]. Ce protocole, par l'intermédiaire du DS, permet aux différents AP d'un ESS d'échanger des informations entre eux. Il définit aussi la possibilité d'utiliser un ou plusieurs serveurs RADIUS pour permettre dans un premier temps d'associer chaque AP avec son BSSID et de consulter cette information et ensuite de donner des clefs de cryptage aux AP pour améliorer la confidentialité des échanges. Cette norme se décompose alors essentiellement selon trois étapes :

- L'arrivée d'un nouvel AP dans l'ESS : lors de cette étape, le nouvel AP envoie un message IAPP-INITIATE aux AP déjà présents dans l'ESS. Cette étape permet de définir les paramètres des futurs échanges entre les AP. Le message IAPP-TERMINATE, à l'inverse indiquera le départ d'un AP.
- L'association d'un nouveau terminal à un AP : une fois que l'AP concerné reçoit la requête de type *Association Request*, il va alors envoyer un message IAPP-ADD aux autres AP pour leur indiquer cette nouvelle association.
- Le déplacement d'un terminal vers un autre AP : à l'approche de la nouvelle cellule, le terminal va envoyer une trame de réassociation. L'AP concerné qui avait déjà reçu un message IAPP-ADD de l'ancien AP pour prévenir l'arrivée dans l'ESS de ce terminal,

envoie alors un message IAPP-MOVE à l'ancien AP pour le prévenir de cette réassociation. Une fois celle-ci achevée, le nouveau AP peut prévenir les autres AP que le terminal lui est maintenant associé.

Cependant, la mise en place de cette norme fut un échec puisqu'en 2006, elle fut abandonnée. En effet, les temps d'interruption, qui variaient de la centaine de millisecondes jusqu'à plusieurs secondes, étaient considérés comme trop importants.

Par la suite deux normes additionnelles, la 802.11k [9] et la 802.11r [10] devaient à leur tour permettre d'améliorer les mécanismes de roaming, mais elles ont tardé à être mises au point et validées par l'IEEE, ce qui a impliqué un désintérêt de la part des constructeurs. Cependant, la norme 802.11r (aussi connue sous le nom de *Fast Basic Service Set Transition*), qui permettrait un temps de ré-association inférieur à 50 ms, semble bien intéresser des constructeurs comme Cisco pour offrir aux entreprises la VoIP sans fil mais à l'heure actuelle, cette norme reste peu répandue et doit encore faire ses preuves. Les constructeurs ont donc développé parallèlement des solutions propriétaires adaptées à leurs besoins spécifiques.

I.2.1.1.c L'alternative actuelle: des solutions propriétaires

Malgré l'échec de la norme 802.11f, le besoin en itinérance reste très important. Des solutions propriétaires vont donc progressivement voir le jour mais l'interopérabilité entre les différents constructeurs n'est toujours pas d'actualité. En effet, contrairement aux préconisations données dans 802.11f, les constructeurs vont principalement développer des solutions basées sur un point central. Ainsi, Cisco utilise essentiellement le protocole LWAPP (*LightWeight Access Point Protocol*) initialement développé par Airspace (racheté plus tard par Cisco) et basé sur le principe d' « intelligence centralisée ». Ce point central, appelé contrôleur WLAN, est en charge de la gestion de l'itinérance mais aussi de la sécurité ou encore de la qualité de service. Il gère donc les communications entre les différents points d'accès qui sont dits « légers ».

Ce protocole, censé apporter une solution pour la mise en place de réseaux WLAN multi-constructeurs dans lequel un utilisateur peut se déplacer, va progressivement intéresser l'IETF (*Internet Engineering Task Force*) qui travaille sur un projet similaire par l'intermédiaire du groupe CAPWAP (*Control and Provisioning of Wireless Access Points*). En Juillet 2006, la [11], issue de ce groupe de travail et dans laquelle LWAPP est comparé à trois autres protocoles, SLAPP (*Secure Light Access Point Protocol*), CTP (*CAPWAP Tunneling Protocol*) et WiCoP (*Wireless LAN Control Protocol*), retient le modèle LWAPP comme base du standard.

Trois années plus tard, en Mars 2009, le protocole CAPWAP est finalisé et spécifié dans [12]. L'itinérance devient donc théoriquement possible et cela, même entre différents points d'accès IEEE 802.11 appartenant à différents constructeurs. Il est même prévu de pouvoir interopérer avec d'autres technologies telles que la norme IEEE 802.16.

Cependant, à l'heure actuelle, aucun constructeur ne supporte officiellement le protocole CAPWAP, bien qu'un projet open-source soit disponible [13]. De plus, la plupart des fournisseurs et intégrateurs restent sceptiques quant à la réelle utilité de faire interopérer des réseaux IEEE 802.11 de différents constructeurs. En effet, du point de vue d'une entreprise, cela rendrait plus compliqué l'administration et le diagnostic en cas de panne, par exemple.

Ceci dit, le protocole CAPWAP reste intéressant dans le cadre de l'utilisation d'un système IEEE 802.11 issu d'un même fournisseur car il permet une gestion optimisée de l'itinérance ainsi que de la sécurité ou encore de la qualité de service.

I.2.1.1.d Travaux concernant la mobilité et IEEE 802.11

Quelques travaux ont été réalisés pour améliorer les mécanismes de mobilité dans les réseaux IEEE 802.11. Ainsi, la RFC 4260 [14] présente une solution qui permet la gestion de la mobilité locale ou globale par FMIPv6 [15] (voir I.2.2.2) entre un ou plusieurs réseaux IEEE 802.11. En effet, ce type de mobilité impliquant des mécanismes de niveau IP ou supérieur ne sont pas traités par les différentes normes IEEE 802.11. Une évaluation de cette proposition a été réalisée dans [16].

I.2.1.2. La technologie IEEE 802.16 (WiMAX)

Le standard IEEE 802.16 [17] définit un système d'accès sans fil à large bande (BWA, *Broadband Wireless Access*) orienté connexion permettant d'offrir à des utilisateurs fixes des débits théoriques allant jusqu'à 70 Mbps (pour des utilisateurs à faible portée). D'autre part, la portée maximale de cette technologie peut aller jusqu'à 50 km (mais le débit est alors fortement réduit).

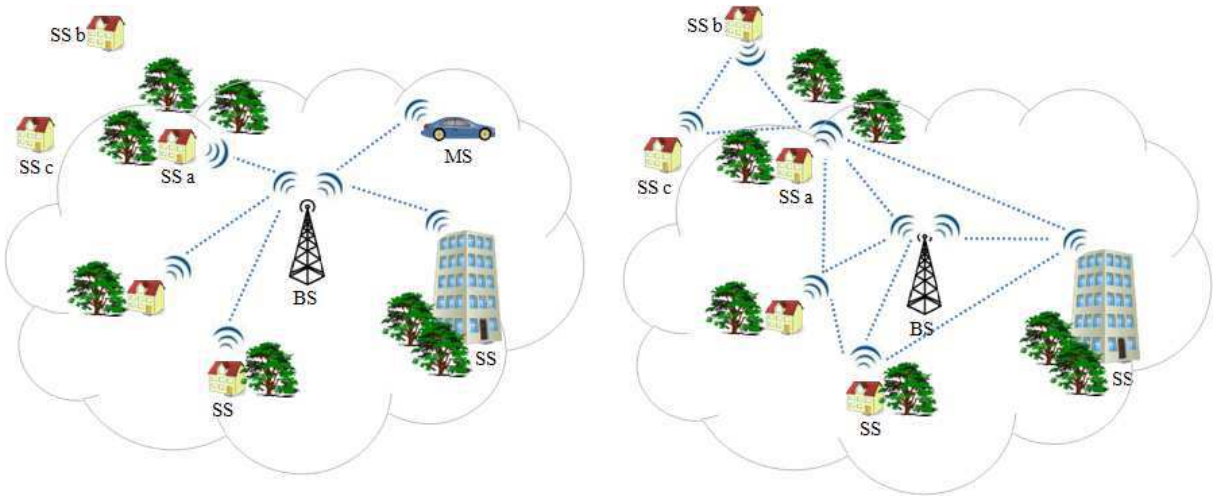
Aussi appelée WiMAX (*Worldwide Interoperability for Microwave Access*) par abus de langage puisque le WiMAX en est seulement une application possible (en Corée du Sud, il existe le WiBro par exemple), cette technologie a été conçue essentiellement pour permettre à des usagers d'accéder à l'Internet haut débit par l'intermédiaire de la boucle locale radio. Cependant, elle peut aussi être utilisée comme moyen d'interconnexion entre différents réseaux locaux sans fil tels que le Wi-Fi. Ces deux applications de l'IEEE 802.16 ont donné naissance à deux modes d'utilisations :

- Le mode point-à-multipoint (PMP, *point-to-multipoint*) dans lequel une station de base (BS, *Base Station*) joue le rôle central de point d'accès. Toutes les stations utilisateurs (SS, *Subscriber Station*) communiquent via ce point central. Ce mode correspond à celui décrit Figure 6.a.
- Le mode maillé (*mesh*) dans lequel les communications directes sont possibles entre utilisateurs (Figure 6.b). Dans ce cas, il est possible que le réseau WMAN ne comporte aucune BS.

On remarque sur la Figure 6 que les deux stations SS b et SS c du réseau WMAN ne peuvent pas être connectées dans le cas PMP alors que dans le cas maillé, vu que les SSs jouent aussi le rôle de relais, la connexion devient possible par l'intermédiaire de la SS a. Cependant, dans ce dernier cas, la connectivité des deux stations situées en dehors de la zone couverte par la BS dépend de la connectivité de SS a. Si cette dernière décide d'éteindre son équipement WiMAX, SS b et SS c perdront leur lien.

Dans la suite de cette partie, nous focaliserons notre étude sur l'architecture point-à-multipoint puisque notre objectif est toujours d'étudier les mécanismes permettant de passer d'une cellule à une autre (aussi appelé *handover*).

En Décembre 2005, une nouvelle version du standard 802.16 est approuvée et fait son apparition sous le nom de IEEE 802.16e [18], plus communément appelé « WiMAX mobile » du fait qu'il apporte principalement une nouvelle fonctionnalité : la mobilité. Cette norme permet théoriquement à des véhicules se déplaçant à une vitesse allant jusqu'à 120 km/h (mais en réalité, la limite est plus proche des 60 km/h) de passer d'une BS à une autre sans perte de connectivité. Cependant, la portée d'une BS est réduite à quelques kilomètres.



(a) **(b)**

Figure 6 – Les différents modes d'utilisations de la norme IEEE 802.16
(a) mode PMP, (b) mode maillé

Ainsi, naissent entre autres, les concepts de station mobile (MS, *Mobile Station*) et de gestion de handover au niveau MAC. La suite de notre analyse sera basée sur ce standard mais avant de présenter les mécanismes de handover, nous allons tout d'abord décrire comment est gérée l'initialisation et l'entrée d'une SS (ou d'une MS) dans un réseau IEEE 802.16e.

I.2.1.2.a Gestion de l'initialisation et de l'entrée d'une SS dans un réseau IEEE 802.16e

Chaque SS est identifiée par une adresse MAC unique qui sera utilisée pour l'initialisation et l'entrée d'une SS dans un réseau IEEE 802.16e.

Nous allons maintenant en décrire les différentes phases (voir Figure 7) :

- **L'écoute du canal descendant (BS vers SS) et la synchronisation avec une BS** : si la SS s'est déjà connectée par le passé à un réseau IEEE 802.16e, elle utilise ces paramètres pour tenter de réacquérir ce canal descendant. Si cela échoue ou si la SS ne possède pas de tels paramètres, elle scanne les différents canaux descendants jusqu'à l'obtention d'un signal descendant valide qui va permettre à la SS et à la BS de se synchroniser au niveau physique.
- **L'obtention des paramètres de transmission sur les canaux montants (SS vers BS) et descendants** : la couche MAC peut alors se synchroniser en obtenant au moins un message DL-MAP (*Downlink map*). Elle restera synchronisée tant qu'elle continuera à recevoir les messages DL-MAP et DCD (*Downlink Channel Descriptor*) correspondant à ce canal. La SS doit alors attendre la réception d'un message UCD

(*Uplink Channel Descriptor*) décrivant les paramètres de transmission d'un canal montant adaptés à ses besoins. Si elle n'en reçoit pas ou s'ils ne conviennent pas, elle cherchera un autre canal descendant. Si les paramètres conviennent, la couche MAC considérera le canal montant valide tant qu'elle recevra des messages UL-MAP (*Uplink map*) et UCD.

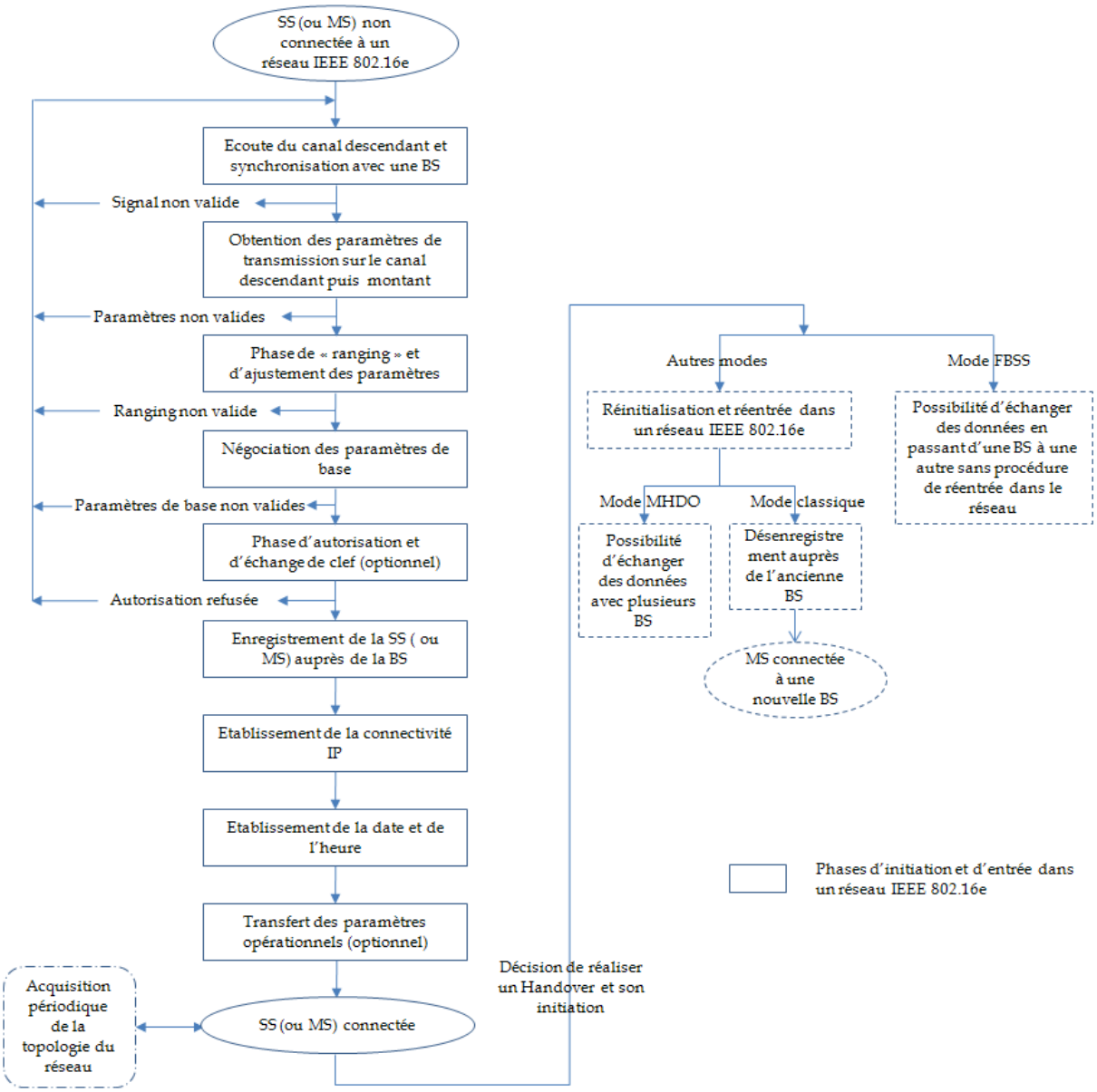


Figure 7 – Le processus d'initialisation et d'entrée dans un réseau IEEE 802.16e ainsi que les différents modes de handover

- **La phase de « *ranging* » et d'ajustement des paramètres** : la SS doit adapter ses paramètres de transmission (la puissance par exemple) en fonction de la BS. Pour cela, après réception d'un message UL-MAP, la SS va envoyer un message RNG-REQ (*Ranging Request*) auquel la BS va répondre par un RNG-RSP (*Ranging Response*). Dans ce message, la BS va préciser les CIDs (*Connection Identifiers*) de deux paires de connexions dites « de gestion » (un CID pour chaque paire) associés à cette SS :

  - Une « basique » utilisée pour échanger des messages MAC de gestion, courts et urgents.
  - Une « primaire » utilisée pour échanger des messages MAC de gestion plus longs et plus tolérants en termes de délai.
  - De plus, à chaque réception d'un message RNG-RSP, la SS doit ajuster ses paramètres de transmission et renvoyer un RNG-REQ jusqu'à ce que le RNG-RSP contienne la notification Ranging Successful.

- **La négociation des paramètres de base** : la SS envoie à la BS un message SBC-REQ (*SS Basic Capability Request*) qui contient d'une part les propriétés de la SS dont la BS a besoin pour l'allocation de bande passante et d'autre part des informations telles que la puissance maximale de transmission et la puissance actuelle de transmission. La BS répond en validant ou non ces paramètres.
- **La phase d'autorisation et d'échange de clef (optionnelle)** : si l'option PKM (*Privacy Key Management*) est activée, la SS et la BS doivent réaliser cette phase avant l'enregistrement de la SS. Cette étape consiste donc à la réalisation du protocole PKM version 1 ou 2 défini dans [18] qui, en résumé, est composé des étapes suivantes :
  - La BS authentifie l'identité de la SS
  - La BS fournit à la SS une AK (*Authentication Key*)
  - La BS fournit à la SS des SAIDs (*Service Association Identifier*) qui correspondent à chacun des services auxquels la SS a souscrit.
  - Pour chaque SAID, la SS lance une procédure d'échange de TEK (*Traffic Encryption Key*) pour valider l'utilisation de ce service.
- **L'enregistrement de la SS** : la SS a envoyé un message REG-REQ (*Registration Request*) à la BS en précisant si l'option PKM était activée. Une fois cette phase achevée, la BS répond par un message REG-RSP (*Registration Response*). Pour une SS ayant indiqué qu'elle est une SS autogérée (option *managed SS*), un nouveau CID lui sera attribué pour une paire de connexions de gestion « secondaire » utilisée pour échanger des messages standards du type DHCP (*Dynamic Host Configuration Protocol*), TFTP (*Trivial File Transfert Protocol*), SNMP (*Simple Network Management Protocol*), etc. La SS est alors autorisée à entrer dans le réseau. Une SS autogérée peut aussi indiquer dans le REG-REQ quelle version d'IP (IPv4 [19] ou IPv6 [20]) elle veut utiliser pour sa connexion de gestion « secondaire ». Les étapes suivantes ne concernent plus que les SS autogérées.
- **L'établissement de la connectivité IP** : cette étape se fait au travers de la connexion de gestion « secondaire ». Le protocole utilisé sera DHCP [21] dans le cas d'une SS utilisant IPv4 et DHCPv6 [22] ou IPv6 Stateless Address Autoconfiguration [23] dans le cas de l'utilisation d'IPv6.
- **L'établissement de la date et de l'heure** : cette étape permet à la SS et la BS de synchroniser leur date et heure au travers du Time Protocol [24] en utilisant la connexion de gestion « secondaire ».
- **Le transfert des paramètres opérationnels (optionnel)** : cette étape permet à la SS d'obtenir un fichier de configuration contenant des informations telles que l'adresse IP

du serveur de logiciels ainsi que des informations spécifiques au constructeur de l'équipement. Elle est réalisée par le protocole TFTP [25] en utilisant la connexion de gestion « secondaire ».

Les services contractés par la SS peuvent ensuite être associés à des connexions de transport (et ne peuvent être transmis à travers les connexions de gestion). De même, les messages de gestion ne peuvent être transmis dans les connexions de transport. Chaque connexion de transport, correspondant à un flux de service donné, est associée à un ensemble de paramètres de QoS qui pourra contenir tout ou partie de ces paramètres : *Traffic Priority*, *Maximum sustained traffic rate*, *Maximum traffic burst*, *Minimum Reserved traffic rate*, *Vendor specific QoS parameters*, *Tolerated jitter*, *Maximum latency*, *Unsolicited grant interval* and *Unsolicited polling interval*. Cependant, il est possible de n'utiliser qu'une seule connexion de transport pour tous les flux de données.

Dans la suite de cette partie, nous allons nous intéresser aux mécanismes définis dans IEEE 802.16e pour permettre la gestion d'un handover, c'est-à-dire le transfert des communications d'une BS à une autre.

I.2.1.2.b La gestion de la mobilité de niveau 2 dans IEEE 802.16e

Dans cette partie, nous parlerons de MS (*Mobile Station*) plutôt que de SS. Une MS, pour s'initialiser et rentrer dans un réseau IEEE 802.16e doit réaliser les étapes décrites précédemment.

La procédure de handover (HO) peut se produire dans le cas d'une MS qui se déplace et qui a besoin de changer de BS parce que son signal faiblit mais aussi dans le cas où la MS peut obtenir des services avec une meilleure QoS en s'associant à une autre BS.

Cette procédure est réalisée selon différentes étapes (voir Figure 7) :

- **L'acquisition de la topologie du réseau** : Une BS envoie périodiquement des messages MOB_NBR-ADV (*Neighbour Advertisement*) qui contiennent des informations sur les canaux des BSs voisines.
- **La sélection d'une nouvelle cellule (associée à une BS)** : Grâce aux informations précédemment acquises, une MS peut alors décider de successivement se synchroniser avec une ou plusieurs BSs voisines pour les « scanner » (cela correspond aux échanges réalisés lors de la phase de « ranging » définie dans le paragraphe précédent). Elle peut ainsi obtenir des informations concernant les paramètres de portée ou les services disponibles et les enregistrer dans une liste d'association qu'elle pourra utiliser lors d'une procédure de handover.
- **La décision de réaliser un handover et son initiation** : Cette décision peut être prise soit par la BS soit par la MS mais dans les deux cas, une ou plusieurs potentielles futures BSs peuvent être contenues dans le message d'initiation:
  - Initiation par la MS : elle envoie une requête de type MOB_MSHO-REQ (*MS HO Request*) à la BS. La BS obtient alors des informations auprès des potentielles futures BS pour évaluer les performances qu'obtiendrait la MS en s'associant avec. Elle répond à la MS en incluant ces informations. La MS prend alors sa décision mais n'est pas obligée de choisir une des BSs contenues dans la réponse.

- Initiation par la BS courante : elle envoie une requête de type MOB_BSHO-REQ (*BS HO Request*) à la MS. Si l'option « Network Assisted » est activée, la MS peut choisir une future BS sans la notifier à sa BS courante. La BS peut aussi forcer la MS à entreprendre un handover en activant l'option « HO operation mode » mais dans tous les cas, le choix de la future BS revient à la MS.

- La BS courante peut alors notifier à une ou plusieurs potentielles futures BS, l'intention de la MS de réaliser un handover mais à tout moment, lors de cette procédure, le handover peut être annulé par la MS et les communications reprendront normalement (si possible).
- Il est à noter que les protocoles de communication entre les différentes BS ne sont pas spécifiés dans le standard IEEE 802.16e.
- **La réentrée dans le réseau** : Une fois le choix d'une BS réalisé, la MS doit réaliser les différentes étapes de l'initialisation et d'entrée d'une SS dans un réseau IEEE 802.16e avec la BS choisie.
- **Le désenregistrement auprès de l'ancienne BS** : Au cours de la procédure de réentrée, la MS indique à son ancienne BS qu'elle peut libérer les services et ressources qui lui sont associés. Les services continueront à être délivrés par cette BS jusqu'à l'expiration du timer « Resource Retain ».
- **La coordination assistée par la MS des transmissions descendantes au niveau de la nouvelle BS** : Cette étape peut être réalisée si l'ancienne et la nouvelle BS impliquées dans le handover peuvent supporter la continuité des connexions pour lesquelles soit l'option ARQ (*Automatic Repeat Request*), soit l'option SDU_SN (*Service Data Unit Sequence Number*) a été activée. La nouvelle BS indique à la MS son intention de réaliser cette procédure en activant l'option « *HO Process Optimization* » lors de l'envoi du message RNG-RSP. Lorsque la MS a achevé avec succès son handover, elle peut grâce à cette option maintenir la continuité des transmissions qui lui sont destinées entre l'ancienne et la nouvelle BS en utilisant le numéro de séquence du bloc ARQ ou de la MAC SDU. La commutation des transmissions de l'une à l'autre BS se fait dès lors que la MS a achevé sa réentrée dans le réseau et qu'elle a transmis un « *Sequence Number Report MAC Header* » à sa nouvelle BS. Cette dernière se met à transmettre les flux de service correspondants à partir de ce numéro de séquence.

Cependant, cette procédure de handover, définie comme étant du « hard handover » [26], prend du temps et peut ne pas être assez rapide si la MS se déplace trop rapidement. Et étant donné que la technologie IEEE 802.16e a été développée pour pouvoir gérer le handover pour des véhicules se déplaçant jusqu'à environ 60 km/h, deux modes supplémentaires ont été définis et peuvent optionnellement être activés :

- Le mode MDHO (*Macro Diversity Handover*) avec lequel une MS est capable d'échanger ses données avec plusieurs BS en même temps.
- Le mode FBSS (*Fast BS Switching*) avec lequel une MS peut échanger ses données en passant d'une BS de rattachement (*Anchor BS*) à une autre sans avoir besoin de réaliser la procédure habituelle de réentrée dans le réseau. Chaque BS contenue dans la

liste doit être prête à échanger des données avec la MS ; elles peuvent se préparer grâce à des mécanismes de Timer envoyés par la MS.

Ces modes sont soumis à plusieurs conditions dont les plus importantes sont :

- Les différentes BSs impliquées dans ces deux modes doivent être contenues dans une liste appelée « Diversity set ».
- Les différentes BSs impliquées doivent être synchronisées sur les mêmes dates et heures.
- Les différentes BSs impliquées doivent utiliser les mêmes fréquences.
- Les différentes BSs impliquées doivent utiliser les mêmes CIDs pour les connexions établies avec la MS.
- Les différentes BSs impliquées doivent partager le contexte MAC qui inclut des informations telles que l'authentification ou encore les différents flux de services en cours et leurs connexions associées.

I.2.1.2.c La gestion de la mobilité locale ou globale dans IEEE 802.16e

La norme IEEE 802.16e a aussi défini la possibilité d'utiliser le protocole Mobile IPv4 [27] dans le cas de handovers impliquant des mécanismes de niveau supérieur en plus du handover de niveau 2 précédemment décrit. La MS doit alors préciser qu'elle utilise Mobile IPv4 lors des messages d'enregistrement. De plus, lors de l'étape « établissement de la connectivité IP », la MS doit sécuriser son adresse à travers la connexion de gestion « secondaire » en utilisant les mécanismes spécifiques à Mobile IPv4. Les différentes BS du réseau IEEE 802.16e joueraient alors le rôle de home agent ou de foreign agent selon la situation.

I.2.1.2.d Travaux concernant la mobilité et IEEE 802.16e

Différents travaux ont été réalisés pour étudier les mécanismes de mobilité pour les réseaux IEEE 802.16e. Ainsi, la RFC 5270 [28] définit comment FMIPv6 [15] (voir I.2.2.2) pourrait améliorer la gestion des handover impliquant un changement d'adresse IPv6 dans IEEE 802.16e. Dans [29] et [30], le WiMAX Forum Network Working Group propose aussi comme alternative à Mobile IPv4 (aussi appelé Client MIPv4 ou CMIPv4), l'utilisation de PMIPv4 (la version v4 n'est pas présentée dans ce rapport mais la version PMIPv6 est détaillée dans le paragraphe I.2.2.4), ce dernier ayant l'avantage de ne nécessiter aucune implémentation spécifique de Mobile IPv4 au niveau de la MS. De plus, ces mêmes documents définissent l'utilisation de Mobile IPv6 [31] dans les réseaux IEEE 802.16e. Enfin, [32] propose aussi une amélioration des mécanismes de gestion de la mobilité dans les réseaux IEEE 802.16e en utilisant mSCTP [33] (détaillé dans le paragraphe I.2.4.3.a).

I.2.2. La gestion de la mobilité au niveau de la couche réseau

L'implémentation la plus largement répandue de la couche réseau correspond au protocole IP (*Internet Protocol*) [19]. Il est principalement en charge de l'acheminement des paquets de la source jusqu'à la destination. Actuellement, ce protocole est déployé sous

l'appellation IPv4 et permet l'utilisation d'un peu plus de quatre milliards d'adresses différentes. Mais le succès grandissant d'Internet laisse à penser que, dans les années à venir, ce nombre ne sera pas suffisant malgré l'utilisation de solutions telles que la traduction d'adresse réseau (NAT, *Network Address Translation*) [34] ou encore le découpage en classe d'adresse [35]. C'est donc principalement pour cette raison de pénurie d'adresse qu'un nouveau protocole est définit : IPv6 [20].

La mobilité n'étant pas incluse dans IPv4, il semble peu probable que les modifications nécessaires pour sa mise en place, définies dans [27], soient réalisées sur ce protocole, en tout cas, pas à grande échelle mais par contre, on est en droit de penser que, si IPv6 est réellement mis en place, des mécanismes de mobilité y seront directement intégrés puisqu'ils auront déjà fait l'objet de grand nombre de recherches d'ici là. C'est pourquoi, dans la suite de cette partie, nous nous concentrerons sur les protocoles de mobilité fonctionnant avec IPv6, où la mobilité est native.

I.2.2.1. Mobile IPv6

En 1994, profitant de l'émergence du nouveau protocole IPv6 et améliorant les mécanismes mis au point dans Mobile IPv4 [27], trois chercheurs (C. Perkins, D. Johnson et A. Myles) soumettent à l'IETF une proposition de protocole de mobilité sur IPv6 appelé Mobile IPv6 (ainsi que MIPv6 ou encore CMIPv6 pour Client MIPv6), qui décrit un moyen de gérer la mobilité de terminaux IPv6. Cependant, des désaccords concernant la sécurisation de Mobile IPv6 ainsi que les différentes optimisations possibles ont rendu sa standardisation longue et laborieuse et ce n'est qu'en juin 2004 que la RFC 3775 [31] fut publiée.

I.2.2.1.a Terminologie

Mobile IPv6 décrit un mécanisme dans lequel trois entités interviennent : le terminal IPv6 mobile (MN, *Mobile Node*) qui peut se déplacer d'un réseau à l'autre et donc changer son point d'attachement d'un réseau ou sous réseau à un autre, l'agent mère (HA, *Home Agent*) en charge de rediriger les paquets à destination du MN lorsqu'il se trouve dans un réseau visité et le terminal correspondant (CN, *Correspondant Node*) qui communique avec le MN. Trois types de réseau sont alors distingués:

- Le réseau mère, qui est le réseau d'origine du MN.
- Le réseau correspondant, qui est le réseau du CN (peut être le même que le réseau mère)
- Les réseaux visités, qui sont les réseaux (autres que le réseau mère) dans lesquels le MN se déplace.

Pour permettre au MN de toujours rester adressable, une adresse permanente dite adresse mère (HoA, *Home Address*) lui est assignée. De plus, lorsqu'il se trouve dans un réseau visité, il se voit attribuer une adresse temporaire (CoA, *Care-of Address*).

I.2.2.1.b Principe de base

Lorsque le MN se trouve dans son réseau mère (cf Figure 8.a), le routage s'effectue de manière standard en se basant sur les tables de routage, puisque, dans son réseau mère, le MN

se comporte comme un terminal IPv6 « fixe ». Lorsque le MN se déplace dans un réseau visité (cf Figure 8.b), il récupère une CoA dont le préfixe sera évidemment celui du réseau visité. Pour cela, à son arrivée dans le réseau visité, soit il reçoit directement un *Router Advertisement* (RA) non sollicité, soit il envoie un *Router Sollicitation* pour forcer l'envoi d'un RA. Grâce à ce message, il connaît le préfixe du réseau et peut donc se construire une adresse grâce au mécanisme d'autoconfiguration IPv6 [23] (concaténation du préfixe et de l'adresse MAC). Ensuite, une fois que les mécanismes de DAD (*Duplicate Address Detection*) [23] ont été effectués pour garantir l'unicité des adresses IPv6 (lien-local et unicast) sur le lien, le MN enregistre sa CoA auprès de son HA en lui envoyant un message de mise à jour d'association : un *Binding Update* (BU) qui comprend à la fois son adresse mère et son adresse temporaire. Le MN attend alors la réponse de son HA par l'intermédiaire d'un *Binding Acknowledgement* (BACK). L'agent mère joue ensuite le rôle de proxy (cf Figure 8.b) : les paquets envoyés par le CN (pour qui la mobilité du MN est transparente) à destination du MN sont interceptés par le HA qui les encapsule et les « tunnèle » (transmet à travers le tunnel mis en place entre le HA et le MN) à destination de la CoA du MN tandis que le MN transmet ses paquets au CN par l'intermédiaire du HA.

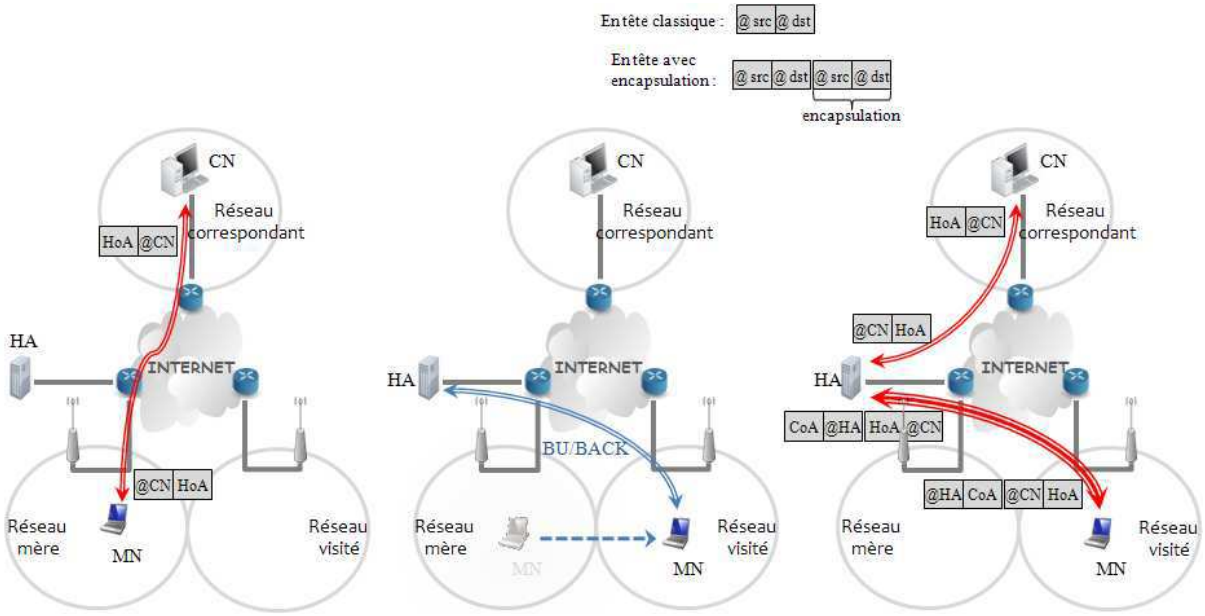


Figure 8 – Mise en place du tunnel bidirectionnel dans Mobile IPv6
(a) Communication directe. (b) Mise à jour de l'association avec le HA. (c) Communication en mode tunnel bidirectionnel.

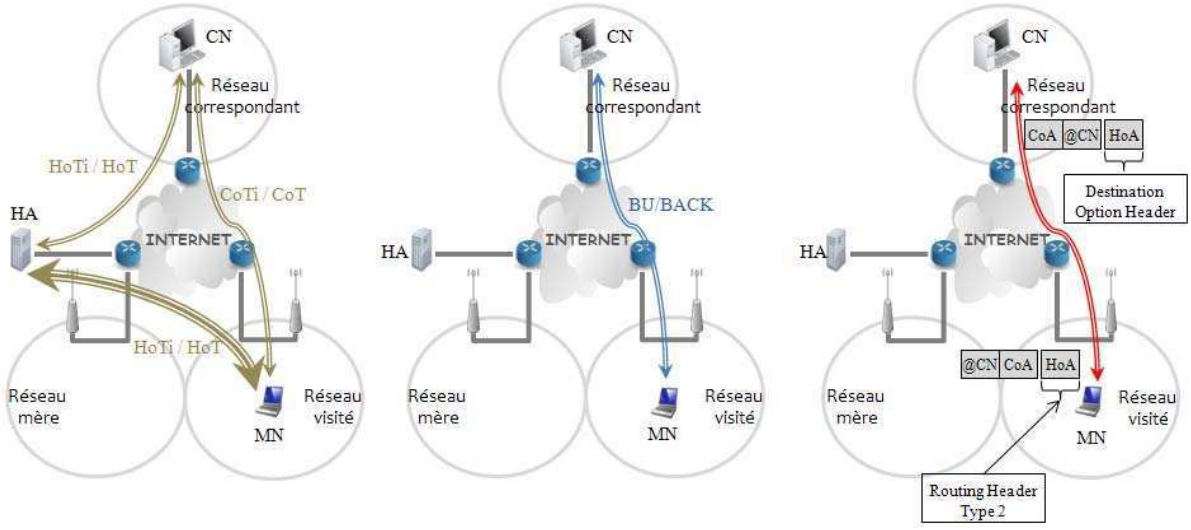
I.2.2.1.c Optimisation de routage et procédure de Return Routability Test (RRT)

Une des principales améliorations de Mobile IPv6 tient dans le fait que le MN a la possibilité d'avertir son correspondant de son adresse temporaire en échangeant directement avec lui les messages BU/BACK. En effet, le routage systématique par l'agent mère du mobile reste particulièrement inefficace au niveau du routage, bien que simple à mettre en œuvre et très sûr puisque la communication entre l'agent mère et le mobile peut être sécurisée par IPsec ([36], [37]) qui utilise des méthodes de sécurité cryptographiques. Par exemple, si le mobile

est en déplacement loin de son réseau mère et qu'il communique avec un serveur proche de lui, il est plus efficace de communiquer directement que de passer par l'agent mère. Cela permet d'économiser des ressources dans l'Internet et surtout au niveau du réseau mère. De plus, si un agent mère doit rediriger les paquets d'un grand nombre de nœuds mobiles, il peut ne pas supporter la charge.

Lorsque le MN reçoit un paquet encapsulé par le HA en provenance initiale du CN, il peut décider de signaler au CN sa CoA par l'échange de messages BU/BACK (cf Figure 9.b), de la même façon qu'avec le HA. Cela permet au CN de communiquer directement avec le MN grâce à l'utilisation de deux options d'IPv6, le *Routing Header, Type 2* et le *Destination Option header,* qui sont ajoutées à chaque paquet IPv6 pour indiquer la HoA du MN (cf Figure 9.c). Les paquets sont alors directement routés du CN au MN (et inversement), mais le MN lorsqu'il reçoit le paquet adressé à sa CoA, récupère dans le *Routing Header* son HoA qui sera utilisée en tant qu'adresse de destination finale du paquet. C'est ce qui permet à Mobile IPv6 d'être transparent pour les applications en mode « optimisée ». Cependant, Mobile IPv6 ainsi que l'optimisation de routage doivent alors être implémentés au niveau du CN.

En effet, tous les correspondants IPv6 potentiels ne supportent pas forcément l'optimisation de route. Dans ce cas, un correspondant répond qu'il ne comprend pas la mise à jour d'association et les communications continuent à passer à travers l'agent mère.



(a) (b) (c)

Figure 9 – Procédure d'optimisation de routage dans Mobile IPv6
(a) Procédure de *Return Routability Test.* (b) Mise à jour de l'association avec le CN.
(c) Communication directe avec options de routage spécifiques.

Un autre problème a été soulevé lors de l'élaboration de la phase d'optimisation de routage du protocole Mobile IPv6 concernant le mécanisme de mise à jour d'association qui pose d'importants problèmes de sécurité. En effet, il est aisé de protéger les échanges de signalisation entre le MN et le HA du fait de la relation administrative qui permet par exemple l'utilisation d'un secret partagé. C'est beaucoup plus compliqué en ce qui concerne les correspondants et pourtant la sécurité des mises à jour d'association est vitale. Sans protection, il serait possible de détourner les communications d'un mobile en redirigeant le trafic pour

l'espionner ou de mener une attaque par déni de service. C'est pourquoi une procédure appelée *Return Routability Test* (RRT) est spécifiée pour protéger la signalisation entre le nœud mobile et ses correspondants.

Les mises à jour d'association pouvant être fréquentes, il est important que cette procédure soit la plus légère possible. Un nœud mobile et un nœud correspondant ne se connaissent pas a priori ; ils ne partagent donc pas de secret susceptible de chiffrer leurs échanges lors des différentes mises à jour d'association nécessaires pendant toute la durée de la communication. L'utilisation d'IPsec [37] et d'une procédure d'échange sécurisé des clefs aurait été trop lourde. La procédure choisie a pour premier but de générer ce secret partagé.

Afin d'éviter l'attaque en déni de service à l'encontre d'un nœud tiers, la procédure de RRT garantit au nœud correspondant que, pour une certaine adresse temporaire et pour une certaine adresse mère, il y a effectivement un nœud mobile prêt à recevoir un paquet.

La procédure est constituée de deux phases préliminaires (cf Figure 9.a), dont l'une teste la HoA (HoTi puis HoT) et l'autre teste la CoA (CoTi puis CoT). Ensuite toute demande de mise à jour ou de destruction d'association sera assujettie à l'exécution correcte de ces deux phases préliminaires.

Les deux phases sont menées parallèlement l'une et l'autre, à l'initiative du nœud mobile. Le nœud correspondant répond aux deux requêtes indépendamment l'une de l'autre en y ajoutant respectivement un *home keygen token* et un *care-of keygen token* à partir desquels le MN peut générer une clef, la *Binding Management Key* (Kbm), qui lui permet de s'authentifier lors de l'échange BU/BACK effectué avec le CN.

I.2.2.1.d Améliorations de l'optimisation de routage

L'optimisation de routage, dont fait partie la procédure de RRT, peut s'avérer longue et pénalisante, particulièrement dans le cadre d'applications temps-réel. Différentes propositions ont alors été faites pour améliorer cette procédure. Ainsi, [38] propose qu'un échange de clef soit réalisé entre le MN et le CN avant qu'un handover n'ait lieu, en considérant que le CN et le MN appartiennent au même domaine administratif. La procédure de RRT n'est alors plus utile mais cette solution ne permet pas de résoudre globalement le problème. [39] et [40] proposent et étudient alors des mécanismes se basant entre autres sur des mécanismes de *Early Binding Update* en réalisant simultanément l'échange des messages HoTi/HoT, CoTi/CoT et BU/BACK avec le HA avant d'échanger les BU/BACK avec le CN ou encore d'échanger les messages HoTi/Hot avant qu'un handover ait lieu.

I.2.2.1.e Conclusion sur Mobile IPv6

Cependant, malgré les améliorations apportées aux mécanismes d'optimisation de routage, Mobile IPv6 souffre toujours d'une latence pouvant entrainer des pertes de paquets importantes qui peuvent être pénalisantes, surtout pour les applications « temps réel ». Cette latence est essentiellement due à trois facteurs :

- Le délai de handover de niveau 2 (détection de changement de réseau, association au nouveau point d'accès).

- La configuration d'une nouvelle CoA [23] (obtention d'une adresse, mécanismes de détection d'adresse dupliquée ou DAD).
- Les échanges de BU/BACK et des messages nécessaires à la procédure de RRT avec le HA et le CN.

Plusieurs solutions présentées dans les paragraphes suivants ont été proposées pour pallier à ces problèmes de délai.

Mobile IPv6 souffre aussi d'un autre problème dû au rôle central que joue le HA dans les mécanismes de mobilité. En effet, un dysfonctionnement du HA (crash, attaque de déni de service, aucune route disponible jusqu'au HA, etc…) impliquerait une interruption totale des communications du MN dans le cas où l'optimisation de route n'a pas été réalisée et, dans tous les cas, rendrait impossible la conservation des communications dans le cas d'un changement de réseau. Différents mécanismes de fiabilité ont donc été proposés dans [41] en se basant sur le principe de redondance du HA. Ce document décrit alors les mécanismes de détection de défaillance d'un HA ou encore de passage d'un HA à un autre. Ceci a cependant comme inconvénient de rendre Mobile IPv6 encore plus complexe et plus lourd en termes d'infrastructure nécessaire.

Enfin, des problèmes de congestions issus de ce même rôle central joué par le HA peuvent intervenir lorsqu'un grand nombre de MN sont associés au même HA.

I.2.2.2. FMIPv6 : Mobile IPv6 Fast Handovers

FMIPv6 [15] est certainement une des améliorations les plus prometteuses qui ait été apportée à la mobilité IPv6. Le but de ce protocole est de réduire le délai du handover en améliorant d'une part le temps de détection du mouvement du MN et d'autre part le temps d'enregistrement de la nouvelle CoA. FMIPv6 définit donc de nouveaux mécanismes, indépendants de la technologie de niveau 2 pour permettre au MN :

- De configurer une adresse IPv6 pour le prochain réseau avant de s'être effectivement déplacé.
- D'envoyer des paquets dès qu'il a détecté un nouveau lien (justement grâce à la pré-configuration d'une adresse IPv6).
- De recevoir des paquets dès que son nouveau routeur d'accès a détecté son attachement (par des mécanismes de tunnel et de bufferisation).

Ces mécanismes sont entièrement compatibles avec Mobile IPv6, (ou, comme le précise la norme, avec d'autres protocoles gérant la mobilité IPv6 mais dans cette partie, nous considérerons que c'est Mobile IPv6 qui est utilisé).

Dans le meilleur des cas, le temps d'interruption global peut être réduit au temps d'interruption dû à la ré-association de niveau 2.

Pour cela, de nouvelles entités apparaissent : le précédent routeur d'accès (PAR, *Previous Access Router*) et le nouveau routeur d'accès (NAR, *New Access Router*) qui correspondent respectivement aux routeurs d'accès avant et après la procédure de handover. De plus, deux adresses temporaires sont définies : la PCoA (*Previous CoA*) et la NCoA (*New CoA*) correspondants respectivement aux adresses temporaires obtenues auprès du PAR et du NAR. L'architecture de référence pour FMIPv6 est illustrée Figure 10.

Lorsque le MN est encore connecté au PAR, il peut obtenir des informations sur d'éventuels futurs points d'accès auxquels il pourrait s'associer grâce à des mécanismes de niveau 2 (par exemple grâce à la phase d'écoute définie dans le paragraphe I.2.1.1.a pour 802.11). Le principe de FMIPv6 est alors de permettre au MN de demander à son PAR des informations concernant les routeurs d'accès correspondants à un ou plusieurs des points d'accès qu'il a découvert en envoyant au PAR un message ICMP du type RtSolPr (*Router Solicitation for Proxy Advertisement*) dans lequel il indique les identifiants (adresse MAC) des points d'accès découverts. En réponse à ce message, le PAR envoie un autre message ICMP de type PrRtAdv (*Proxy Router Advertisment*) dans lequel il indique les informations concernant les routeurs d'accès (AR) environnants correspondants aux points d'accès indiqués. Ces informations sont de la forme (AP-ID, AR-Info) et les AR-Info peuvent contenir :

- L'adresse MAC du nouveau routeur
- L'adresse IP du nouveau routeur
- Le préfixe IPv6 annoncé par ce nouveau routeur
- Eventuellement une nouvelle CoA (NCoA) si le PrRtAdv est envoyé par le PAR sans sollicitation de la part du MN. Si cette option est présente, le MN doit immédiatement envoyer un FBU (*Fast Binding Update*) s'il ne veut pas risquer une perte de connectivité.

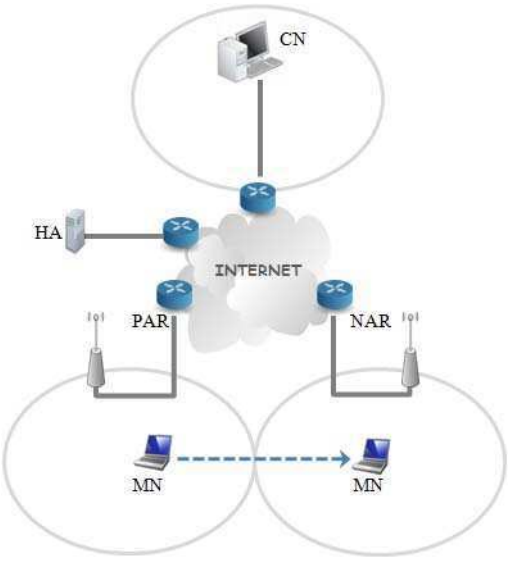


Figure 10 – Architecture de FMIPv6

Une fois que le message PrRtAdv est reçu par le MN, deux modes opératoires sont définis par FMIPv6 : les modes prédictif et réactif.

I.2.2.2.a Le mode prédictif de FMIPv6

Lorsque cela est possible, le MN, dès qu'il reçoit le PrRtAdv (et s'est configuré une NCoA si nécessaire), envoie un FBU à son PAR directement par le lien de niveau 2 correspondant. Le mode prédictif (voir Figure 11) intervient lorsque le MN reçoit un FBACK (*Fast Binding Acknowledgement*) depuis ce même lien. Sinon, le MN considère le FBU perdu et en enverra un autre depuis le nouveau lien, ce qui correspond au mode réactif.

Lorsque le PAR reçoit le FBU dans lequel sont indiqués la PCoA du MN ainsi que la NCoA en tant que *Alternate CoA Option*, il envoie au NAR un message HI (*Handover Initiate*) contenant l'adresse MAC du MN, sa PCoA et la NCoA que le MN souhaite utiliser. Le NAR doit alors réaliser la procédure de DAD avant de répondre par un message HACK (*Handover Acknowledge*) qui précise si le handover est accepté ou non et, s'il est accepté, il contient aussi la NCoA que le MN devra utiliser. Le PAR envoie alors un FBACK au MN en lui précisant cette NCoA. Un tunnel entre le PAR et le NAR est alors mis en place, le PAR intercepte les paquets à destination de la PCoA du MN et les transmet au NAR qui peut, soit les transmettre à la NCoA du MN s'il est déjà attaché, soit les bufférisers. Le MN, une fois le FBACK reçu, peut changer de lien et s'attacher au NAR ; une fois attaché, le MN doit immédiatement envoyer un message UNA (*Unsolicited Neighbor Advertisement*) au NAR qui peut alors lui transmettre ses paquets. Le MN peut alors aussi recommencer à transmettre des paquets par le tunnel inversé (MN→NAR→PAR→CN par exemple si la communication se faisait directement entre le MN et le CN avant le déplacement). Ce tunnel bidirectionnel est maintenu suffisamment de temps pour que le MN puisse échanger des BU/BACK avec son HA et éventuellement ses CNs.

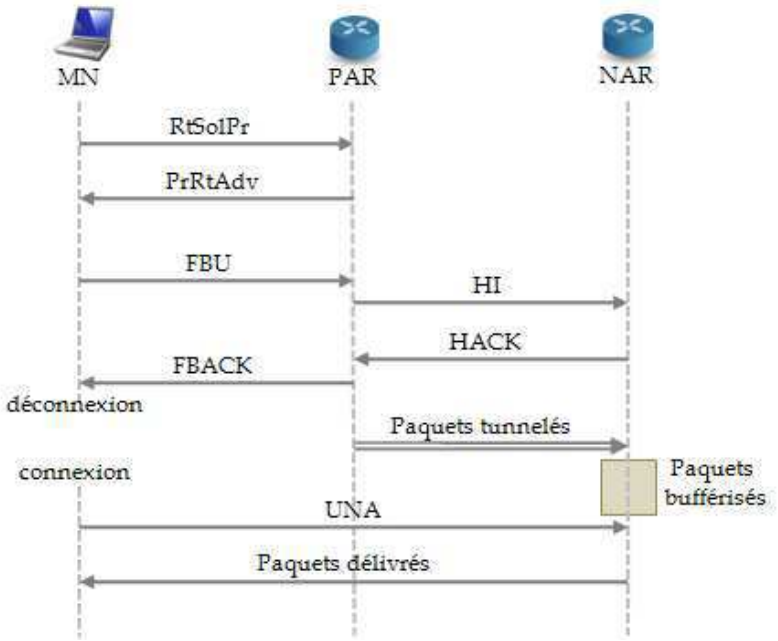


Figure 11 – FMIPv6 en mode prédictif

I.2.2.2.b Le mode réactif de FMIPv6

Le mode opératoire de FMIPv6 est dit réactif lorsque le MN n'a pas la possibilité d'envoyer le FBU depuis le lien correspondant à son PAR ou encore dans le cas où le MN envoie le FBU depuis ce lien mais ne reçoit pas le FBACK avant d'avoir changé de réseau de rattachement. Le MN, une fois rattaché au NAR, envoie un message UNA au NAR immédiatement suivi d'un FBU (même si le MN en a déjà envoyé un depuis l'ancien réseau ; en effet, il n'a aucun moyen de savoir si le PAR l'a reçu) à destination du PAR, le NAR ne faisant que le lui transmettre. Ensuite, les messages HI/HACK sont échangés de la même manière que dans le mode prédictif. Le PAR peut alors tunneler le FBACK ainsi que les paquets à destination du MN en passant par le NAR jusqu'à ce que l'échange des messages

spécifiques à Mobile IPv6 soit terminé. Dans le sens inverse, dès que le MN a reçu le FBACK, il peut recommencer à émettre des paquets par le tunnel bidirectionnel pendant l'échange des messages Mobile IPv6. La Figure 12 résume les échanges de messages réalisés par FMIPv6 en mode réactif.

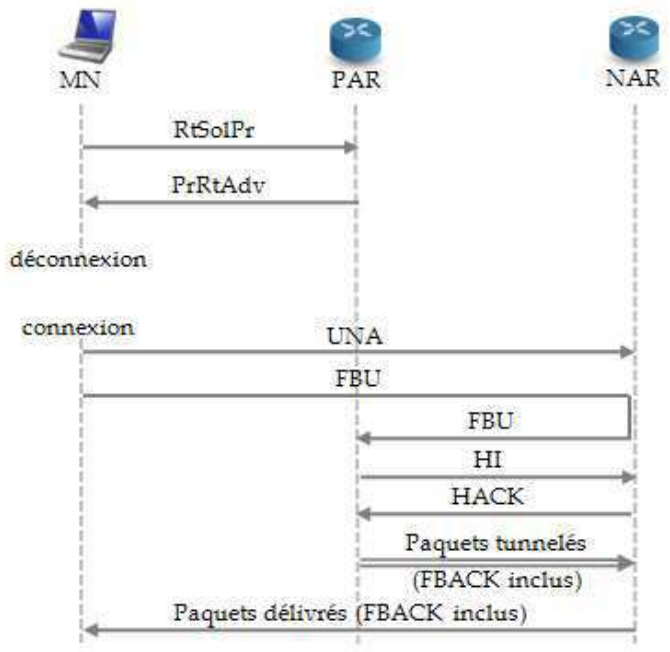


Figure 12 – FMIPv6 en mode réactif

I.2.2.2.c Handover initié par le réseau

Dans certaines technologies sans fil, le handover peut être initié par le réseau et non plus par le mobile. Dans ce cas là, le PAR va envoyer au MN un message PrRtAdv non sollicité contenant l'adresse MAC, l'adresse IP et le préfixe de sous réseau du NAR. Le MN doit alors configurer une NCoA et envoyer un FBU au PAR, la suite des opérations étant l'un ou l'autre des deux modes décrits précédemment selon si le MN reçoit le FBACK depuis le lien correspondant au PAR ou non.

I.2.2.3. HMIPv6 : Hierarchical Mobile IPv6

Les mécanismes de Mobile IPv6 se révèlent assez inefficaces lorsque le mobile change de point de rattachement à l'Internet à l'intérieur d'un même domaine et ceci particulièrement lorsque la distance de déplacement du MN est faible en comparaison avec les distances MN/HA et MN/CN. En effet, chaque déplacement nécessite au minimum un aller retour entre le réseau visité et l'agent mère (BU/BACK). De plus, si le MN est en communication avec plusieurs CNs et qu'il veut rétablir une communication directe avec eux, il doit échanger un nombre de messages importants qui, dans le cadre de technologies sans fil par exemple, peuvent se révéler pénalisants en termes de charge du système.

HMIPv6 [42] a alors été proposé pour permettre une meilleure gestion de ces déplacements à l'intérieur d'un domaine en se basant sur un modèle hiérarchique de gestion

de la mobilité. Pour cela, HMIPv6 propose l'utilisation d'une nouvelle entité : le MAP (*Mobility Anchor Point*), un routeur localisé dans un domaine composé de plusieurs réseaux visités et utilisé comme agent mère local par le MN. Si le MN se déplace à l'intérieur d'un domaine MAP (domaine local dans lequel la mobilité est gérée par un MAP), la signalisation nécessaire à la mobilité IPv6 se limite à ce domaine et devient alors transparente pour le HA et le (ou les) CN. HMIPv6 définit aussi deux nouvelles adresses temporaires : la RCoA (*Regional CoA*) allouée par le MAP au MN et la LCoA (*On-Link CoA*) allouée par le routeur d'accès courant auquel le MN est rattaché.

Il est important de préciser que HMIPv6, tout comme FMIPv6 est totalement compatible avec Mobile IPv6, le MN peut ainsi choisir ou non d'utiliser HMIPv6 pour gérer sa mobilité (par exemple, s'il se trouve dans son réseau mère ou dans un réseau proche de son réseau mère, il peut choisir d'utiliser son HA plutôt qu'un MAP mais le but de cette partie n'est pas de préciser quand ce choix doit être fait). Cependant, comme précisé dans [42], HMIPv6 peut être utilisé indépendamment de Mobile IPv6 (sans HA), les MAPs successifs jouant le rôle de HA. L'architecture de référence de HMIPv6 est illustrée par la Figure 13.

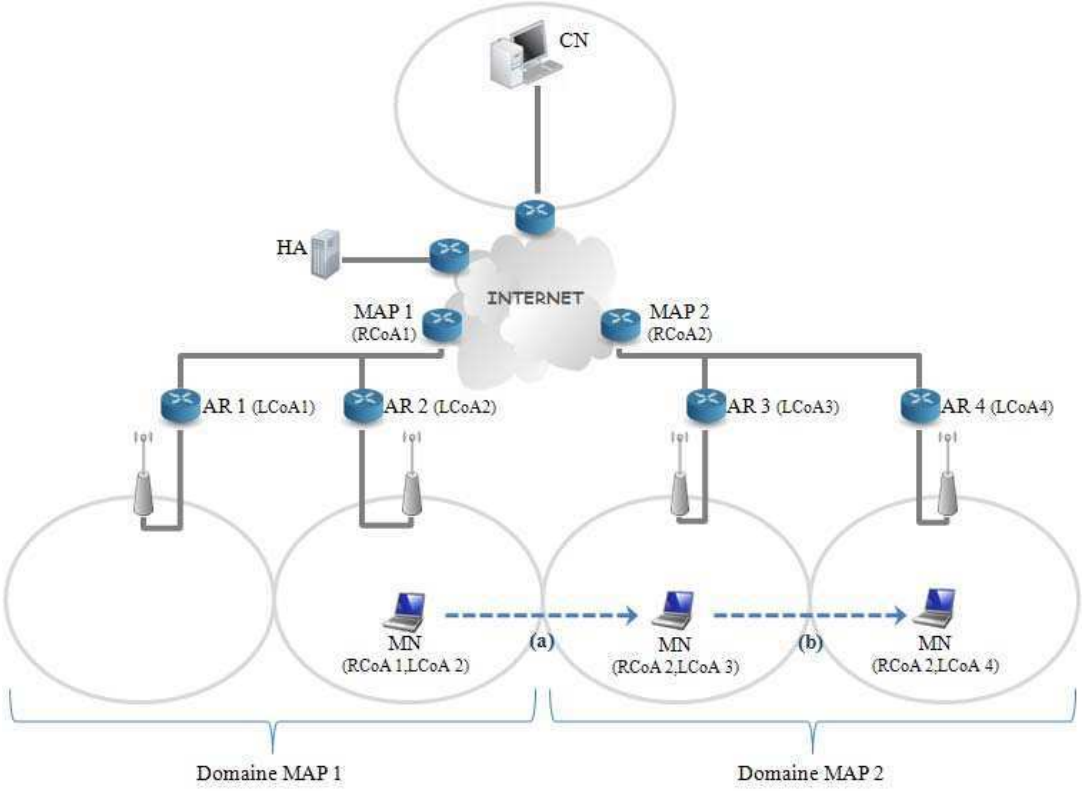


Figure 13 – Architecture de HMIPv6

I.2.2.3.a Mobilité avec changement de domaine MAP

Bien que l'utilisation de HMIPv6 présente un intérêt essentiellement dans le cas d'une mobilité à l'intérieur d'un domaine MAP, il permet aussi sa gestion lors de changement de domaine. Ainsi, lorsque le MN arrive dans le nouveau domaine MAP2 (déplacement (a) sur les Figures 13 et 14), il doit configurer ses nouvelles RCoA et LCoA. Pour cela, lorsqu'il reçoit un *Router Advertisement* de l'AR3, il cherche l'Option MAP (cette option doit pour l'instant être manuellement configurée par l'administrateur réseau au niveau des AR et des

messages qu'ils envoient) qui contient une ou plusieurs adresses IP de MAP. Le MN choisit alors le MAP ayant la valeur de préférence la plus haute. Dans notre exemple, il obtient l'adresse IP du MAP2 ce qui lui permet de configurer sa RCoA2 comme décrit dans [43]. Il construit aussi sa LCoA3 par les mécanismes classiques d'autoconfiguration IPv6.

Le MN prévient alors le MAP2 de sa nouvelle association entre la RCoA2 et la LCoA3 par un message LBU (*Local Binding Update*). Le MAP2 répond alors par un BACK contenant un *Routing Header, Type 2* qui inclut la RCoA2. Après s'être enregistré auprès de son MAP, le MN doit prévenir son HA et éventuellement son CN en échangeant des BU/BACK spécifiant l'association entre sa RCoA2 et sa HoA, de la même manière que pour Mobile IPv6. Une fois que ces opérations ont été réalisées, le MAP2 joue le rôle de proxy en interceptant les paquets à destination de la RCoA2 et en les encapsulant à destination de la LCoA3 du MN de la même manière que le fait le HA avec la HoA et la CoA.

Le MN a enfin la possibilité de communiquer directement avec son CN (sans passer par le MAP, si par exemple le CN se situe dans le même sous-réseau) en lui envoyant un message BU spécifiant l'association LCoA3 – HoA mais cette possibilité n'est pas présentée sur la Figure 14.

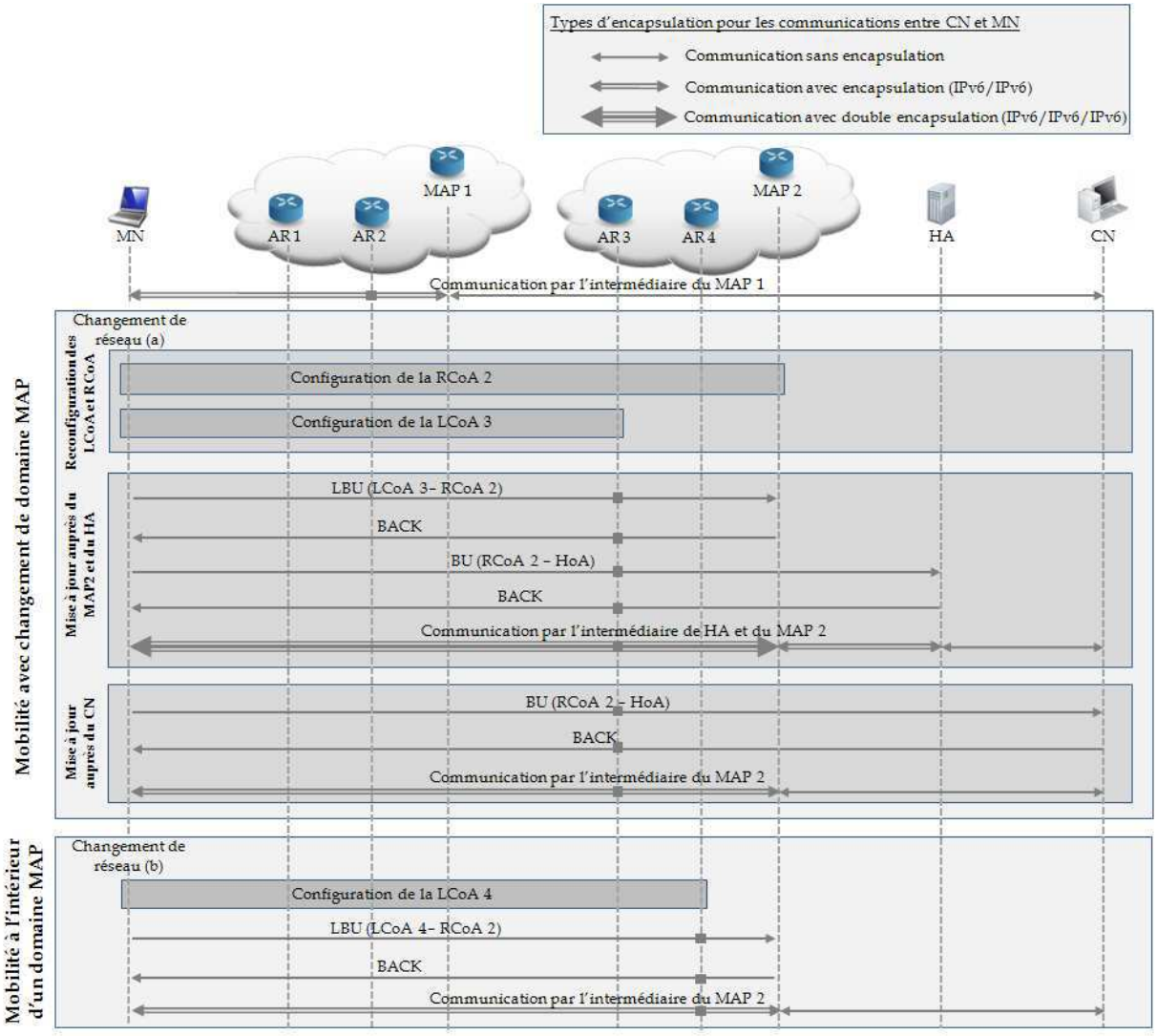


Figure 14 – Gestion de la mobilité par HMIPv6

I.2.2.3.b Mobilité à l'intérieur d'un domaine MAP

L'avantage de l'utilisation de HMIPv6 réside dans les déplacements à l'intérieur d'un même domaine MAP. En effet, dans ce cas là (déplacement (b) sur les Figures 13 et 14), le MN a juste besoin d'envoyer un LBU au MAP2 pour lui dire qu'il a maintenant l'adresse locale LCoA4. Il n'est alors pas nécessaire d'envoyer de message aux HA et CN. HMIPv6 peut alors typiquement être utilisé comme solution de gestion localisée de la mobilité à l'intérieur d'un même réseau d'accès. Cependant, cela implique une encapsulation IPv6/IPv6 entre le MAP et le MN qui peut être pénalisante sur le lien sans fil utilisé par le MN.

I.2.2.3.c Combinaison de HMIPv6 et de FMIPv6

Bien qu'efficace dans le cadre de déplacement intra-domaine MAP, HMIPv6 se révèle encore plus long que Mobile IPv6 dans sa gestion des déplacements inter-domaine MAP. En effet, il ajoute un échange supplémentaire de LBU/BACK entre le MN et le MAP.

Pour réduire l'impact de ce type de déplacement, HMIPv6 conseille l'envoi d'un LBU à destination de l'ancien MAP (MAP1 dans notre exemple) en lui spécifiant sa nouvelle LCoA. Ce dernier pourra alors transmettre les paquets vers la nouvelle localisation du MN, si cela est autorisé (à l'intérieur d'un même domaine administratif par exemple), mais cette étape n'est pas spécifiée et finalement rejoint le principe de FMIPv6.

L'idée de combiner HMIPv6 avec FMIPv6 a donc fait son apparition. Cette combinaison, aussi connue sous le nom de F-HMIPv6 peut être réalisée de plusieurs manières comme le spécifie l'ancienne RFC correspondant à HMIPv6 dans son appendice [44]. Une première solution consiste à mettre en place un tunnel, comme dans FMIPv6, entre le PAR et le NAR mais cela entraine un double passage par le lien MAP-PAR. Pour éviter cela, une deuxième solution a donc été de proposer la mise en place d'un tunnel entre le MAP et le NAR.

En appliquant cette dernière combinaison à notre exemple de la Figure 13, dans le cas d'un déplacement (a), le gain en termes de temps de handover est important pour HMIPv6 puisque les paquets sont retransmis depuis la PAR ou le MAP pendant le handover de niveau 2 tandis que FMIPv6 ne gagne que le double passage par le lien MAP-PAR. Par contre, pour le déplacement (b), les deux protocoles tirent réellement profit l'un de l'autre puisque, en plus d'éviter le double passage inutile, le temps de latence de niveau 2 est supprimé par FMIPv6 et le MN n'a besoin de mettre à jour ses associations qu'avec le MAP plutôt qu'avec le HA et le CN.

I.2.2.4. PMIPv6 : Proxy Mobile IPv6

Une autre proposition concernant la mobilité IPv6 a été faite pour que sa gestion soit entièrement réalisée par le réseau. Après plusieurs années de recherches, l'IETF NetLMM Working Group sort donc un standard sous le nom de Proxy Mobile IPv6 [45]. Le besoin pour une solution de mobilité basée uniquement sur le réseau a tout d'abord été adressée par [3] en expliquant que cela permettait à n'importe quel nœud mobile de se déplacer entre différents réseaux d'accès sans avoir besoin d'implémenter lui-même des solutions de mobilité spécifiques.

I.2.2.4.a Principes de base

PMIPv6 définit alors la notion de domaine PMIPv6 dans lequel la mobilité est gérée par ce protocole. Un MN localisé à l'intérieur d'un domaine PMIPv6 est lié à un LMA (*Local Mobility Anchor*) qui joue le rôle de son HA. De plus, lorsqu'il se déplace, le MN s'attache à des MAG (*Mobile Access Gateway*) successifs qui jouent le rôle de routeurs d'accès responsables de la signalisation liée à sa mobilité. Le principe de PMIPv6 est alors d'émuler le fait que le MN se trouve toujours dans son réseau mère en lui assignant un HNP (*Home Network Prefix*), celui-ci étant uniquement assigné à ce MN. Le MN considère donc le domaine PMIPv6 comme un seul lien. PMIPv6 précise aussi que le LMA a la possibilité d'assigner plusieurs HNP à un même MN (de la même manière, ils ne seront assignés qu'à ce MN) mais dans la suite, nous considérerons qu'un seul HNP est assigné (le principe étant le même). L'architecture de référence de PMIPv6 est illustrée par la Figure 15.

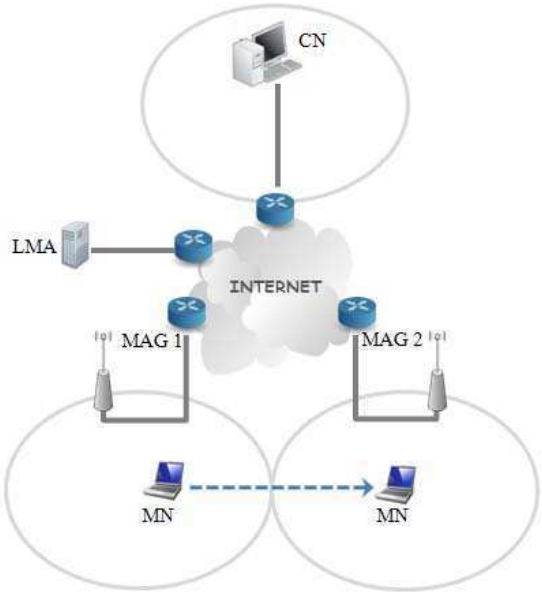


Figure 15 – Architecture de PMIPv6

Lorsqu'un MN arrive dans un domaine PMIPv6 (le domaine géré par le MAG1 sur la Figure 16), il doit tout d'abord se rattacher au réseau d'accès dans lequel il se trouve. Le MAG1 peut alors détecter l'arrivée de ce MN et déterminer à partir de son identifiant ou MN-Id (qui peut être le *Network Access Identifier* [46] ou encore l'adresse MAC) si celui-ci est autorisé à utiliser les services de mobilité ou pas. Si oui, le MAG1 envoie un PBU (*Proxy Binding Update*) au LMA, incluant l'identifiant du MN. Ce dernier alloue alors au MN un HNP, met à jour sa table d'association et met en place un tunnel bidirectionnel jusqu'à la Proxy-CoA1 du MAG1 qu'il voit comme étant la CoA courante du MN. Le LMA répond alors au MAG1 par un PBA (*Proxy Binding Acknowledgement*) incluant le HNP du MN. Le MAG1 réalise lui aussi les mécanismes nécessaires à la mise en place du tunnel vers l'adresse du LMA (LMAA, *LMA address*) et au transfert des paquets vers le MN. Le MAG1 envoie alors un RA (*Router Advertisement*) dans lequel il indique au MN son HNP, le préfixe IPv6 que le MN doit utiliser pour configurer sa HoA. Dès lors, tous les paquets à destination de ce préfixe seront routés par le LMA vers le MAG1 qui, à son tour, les transmettra au MN. Les paquets envoyés par le MN suivront le même trajet.

Lorsque le MN change de réseau et donc de MAG de rattachement, le MAG1 détecte la déconnexion et lance une procédure de désenregistrement auprès du LMA par l'échange de PBU/PBA. Le LMA lance alors un timer à la fin duquel il supprime l'association entre le HNP et la Proxy-CoA1. Une fois que le MN s'est attaché au MAG2, de la même manière que précédemment, un tunnel bidirectionnel entre le LMA et le MAG2 est mis en place et toutes les communications depuis/vers le MN passent par ce nouveau tunnel.

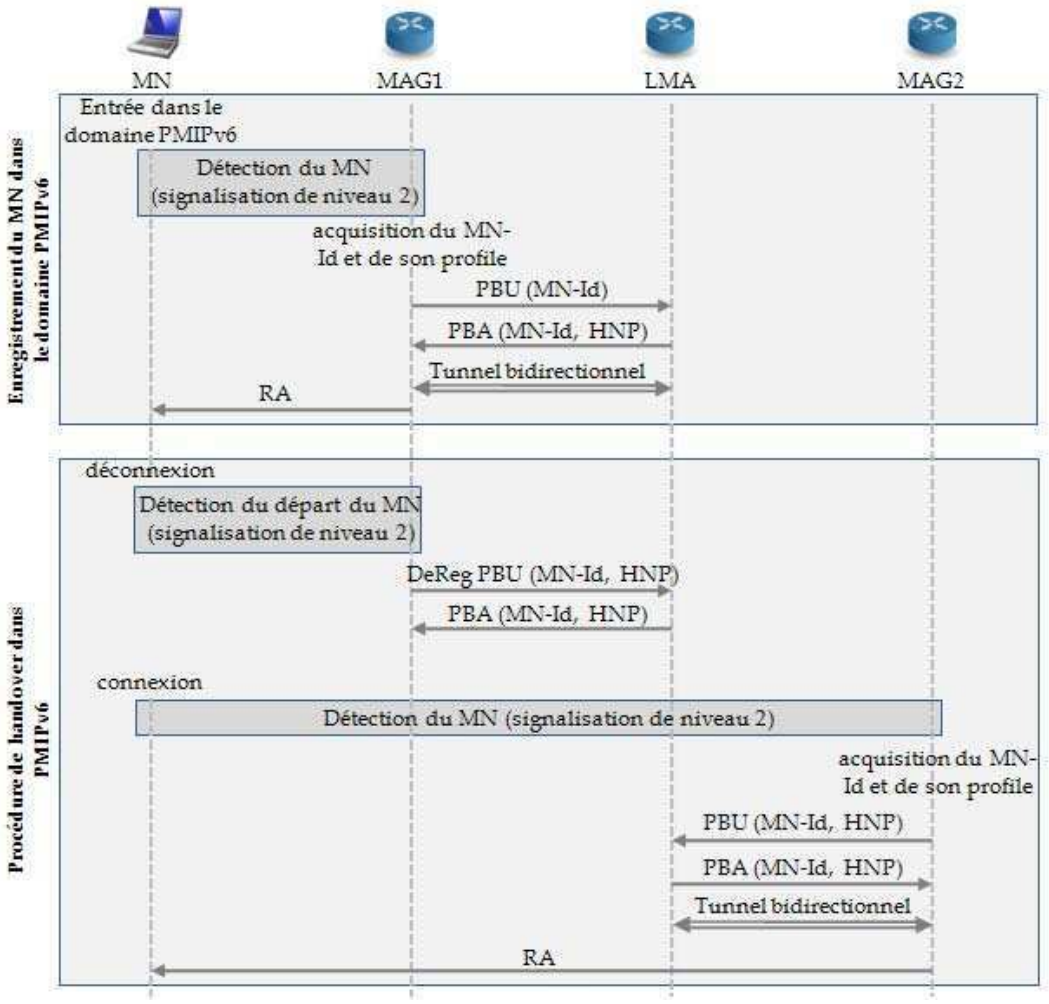


Figure 16 – Entrée d'un MN dans un domaine PMIPv6 et procédure de handover

Notons que sur la Figure 16, le désenregistrement auprès du MAG1 est réalisé avant l'enregistrement auprès du MAG2 mais ces deux procédures sont décorrélées et peuvent se produire en même temps ou bien dans l'autre sens selon le moment où les MAGs détectent les connexion/déconnexion.

I.2.2.4.b Gestion d'interfaces multiples

Le standard PMIPv6 prévoit la possibilité pour le MN d'utiliser plusieurs interfaces simultanément. Dans ce cas là, le LMA doit allouer une session de mobilité (et un ou plusieurs HNP) à chacune des interfaces. Dans le cas d'un handover entre deux interfaces, le LMA décide de quand il est nécessaire de créer une nouvelle session et quand il est nécessaire

de mettre à jour une session selon les paramètres présents dans les PBU (technologie d'accès utilisée, indicateur de handoff).

I.2.2.4.c Optimisation de routage

PMIPv6 prévoit aussi la possibilité d'éviter le passage par le tunnel MAG-LMA lorsque le MAG détecte que le CN est situé sur le même lien que le MN. Cependant, la communication directe entre le MN et le CN n'est pas prévue dans le standard, ce qui peut entrainer de longs délais supplémentaires tout comme dans Mobile IPv6.

Pour éviter cela, de nombreuses propositions ont été faites. Ainsi, [47] propose des mécanismes basés sur ceux de Mobile IPv6 en permettant de ne faire qu'une fois le *Home Test Init* tandis que [48] propose un mécanisme d'optimisation basé sur une nouvelle entité : le *Route Optimization Controller*. Enfin [49] propose quatre mécanismes d'optimisation pour gérer aussi bien la mobilité intra ou inter-LMA en considérant que le MN et le CN sont tous deux rattachés à un couple MAG/LMA :

- L'optimisation « BCE-based » : deux nouveaux messages (CBU/CBA pour Correspondant BU/BA) sont alors échangés entre LMAs (cas inter-LMA) et entre LMA et MAGs pour permettre aux MAGs correspondants au MN et au CN de communiquer directement entre eux.
- L'optimisation « Binding query-based » : un MAG associé au MN et recevant des paquets à destination d'un CN va interroger ses MAGs voisins pour savoir si le CN est attaché à eux. Si oui, les messages seront directement transmis entre les MAGs concernés.
- L'optimisation « Binding multicast-based » : lorsque un LMA reçoit un nouveau PBU, il envoie en multicast cette association aux MAGs qu'il a en charge pour leur permettre de savoir que pour communiquer avec tel nœud, ils peuvent directement transmettre les messages vers tel MAG.
- L'optimisation « Ring, bus and mesh-based » qui tient compte de la topologie du réseau pour faire communiquer directement les MAG entre eux.

I.2.2.5. Conclusion sur la mobilité de niveau réseau

Comme on a pu le constater, un grand nombre de protocoles de gestion de la mobilité ont été proposés au niveau de la couche réseau. En effet, sa position dans le modèle en couche permet d'une part de conserver la connectivité d'un utilisateur mobile indépendamment de la technologie qu'il utilise et d'autre part de faire basculer les communications de façon transparente pour les couches supérieures. Les différentes améliorations permettent aussi de limiter le temps de handover en améliorant les mécanismes de détection de mouvement et d'enregistrement de la nouvelle adresse (FMIPv6) et en limitant la signalisation de mobilité à des domaines réduits (HMIPv6). Une gestion orientée réseau (PMIPv6) est aussi proposée pour permettre à des nœuds mobiles de ne pas implémenter eux-mêmes les mécanismes de mobilité. Enfin, un récent standard, DSMIPv6 [50] (*Dual Stack Mobile IPv6*), détaille les mécanismes permettant d'adapter l'utilisation de Mobile IPv6 à des réseaux IPv4. Toutefois, cette spécification précise que l'optimisation de route n'est pas possible dans le cas où le MN

se déplace dans un réseau uniquement IPv4 ou lorsqu'il doit utiliser une HoA IPv4 pour communiquer avec un CN non équipé d'IPv6.

Cependant, ces solutions souffrent de désavantages tels que la nécessité de modifier l'infrastructure réseau (HA, PAR, NAR, MAP, MAG, LMA), l'ajout d'en-têtes supplémentaires à chaque paquet IP dès que le nœud mobile se situe dans un réseau visité et le passage par une phase de tunnel bidirectionnel pénalisante en termes de délai d'acheminement de paquet et de temps d'interruption.

I.2.3. La gestion de la mobilité entre les couches réseau et transport : le protocole HIP

I.2.3.1. Principe de base

L'architecture HIP (*Host Identity Protocol*) [51] propose la mise en place d'une couche intermédiaire entre les couches réseau et transport permettant de séparer l'identification et la localisation d'un hôte, rôles que jouent actuellement les adresses IP. Ce protocole est essentiellement basé sur un nouvel espace de nom constitué d'identifiants d'hôte (HIs, *Host Identifiers*). Chaque hôte possède alors un ou plusieurs HIs, chacun d'eux étant la clef publique d'une paire asymétrique de clef. Un HI peut être soit un HIT (*HI Tag*, identifiant sur 128 bits), soit un LSI (*Local Scope Identifier*, identifiant sur 32 bits) mais nous utiliserons les HITs. Les HITs peuvent alors être publiés ou non par exemple dans la base DNS (*Domaine Name Server*).

L'utilisation de HIP permet donc d'attacher une socket à un HIT plutôt qu'à une adresse IP, le HIT étant lui-même attaché (dynamiquement) à une ou plusieurs adresses. Les adresses IP ne sont plus alors utilisées que pour l'expédition des paquets vers la destination.

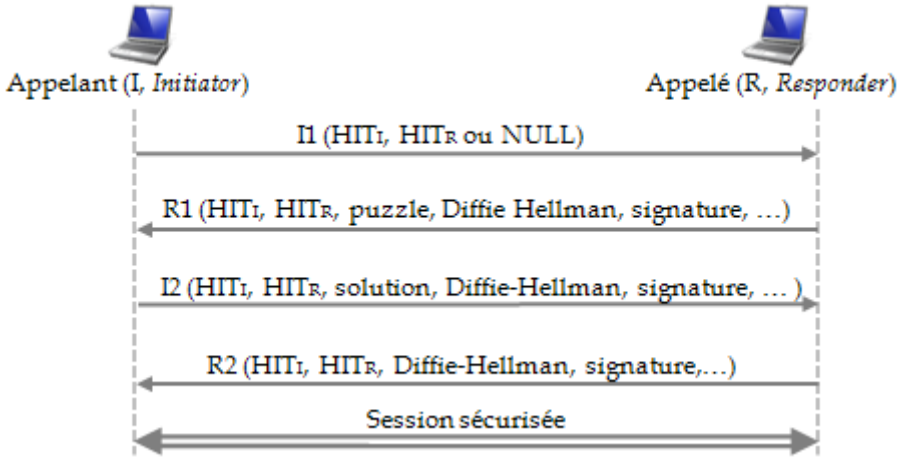


Figure 17 – Session HIP

Pour établir et authentifier une connexion, un échange de base en quatre temps avec échange de clef Diffie-Hellman [52] est préconisé dans [53] entre l'appelant et l'appelé (voir Figure 17), identifiés respectivement par l'HIT_I et l'HIT_R. L'appelant envoie un premier paquet I1 qui contient son HIT (et éventuellement celui de l'appelé s'il le connaît) pour déclencher l'échange de clef. Les trois messages suivants (R1-I2-R2) permettent l'échange

sécurisé des clefs respectives. Le paquet R1 contient, en plus des HITs respectifs de l'appelant et de l'appelé, un problème cryptographique (dit « puzzle ») que l'appelant devra résoudre ainsi que le début d'un échange Diffie-Hellman. De plus, il contient aussi les informations permettant de créer une association de sécurité (SA, *Security Association*) IPsec [37]. Enfin, ce paquet HIP est signé. Une fois que le puzzle est résolu et que la signature est vérifiée, l'appelant peut envoyer un paquet I2 contenant les HITs de l'appelant et de l'appelé et la solution au puzzle. Il permet aussi de poursuivre l'échange Diffie-Hellman et contient les informations permettant la création d'une SA IPsec. Ce paquet est, lui aussi, signé. L'appelé valide alors la réponse, la solution du puzzle ainsi que la signature et peut finaliser l'échange Diffie-Hellman en envoyant un paquet R2 signé et contenant les HITs des deux entités. L'appelant et l'appelé connaissent maintenant mutuellement le HIT et l'adresse IP l'un de l'autre ; le HIT pour l'identification et l'adresse IP pour la localisation.

I.2.3.2. Gestion de la mobilité avec HIP

Profitant de cette séparation entre adresse IP et HIT, une proposition de la gestion de la mobilité a alors été faite [54]. Lorsqu'un nœud se déplace et obtient une nouvelle adresse, il prévient directement son correspondant (voir Figure 18) en lui envoyant un message HIP UPDATE contenant un *locator* ou paramètre de localisation (principalement l'adresse IP mais il peut contenir un identifiant tel que le ESP *Security Parameters Index* si ESP (*Encapsulating Security Payload*) est utilisé ainsi que les numéros de port, etc.). Le correspondant reçoit le message, met à jour l'adresse du nœud mobile et lui renvoie un message UPDATE contenant un paramètre *ECHO_REQUEST* dans lequel est placée n'importe quelle suite de caractère pour vérifier cette adresse. Le nœud mobile répond alors à ce message par un UPDATE ACK en répétant cette suite de caractère dans le paramètre *ECHO_RESPONSE*. La communication peut alors reprendre en toute sécurité entre les deux nœuds.

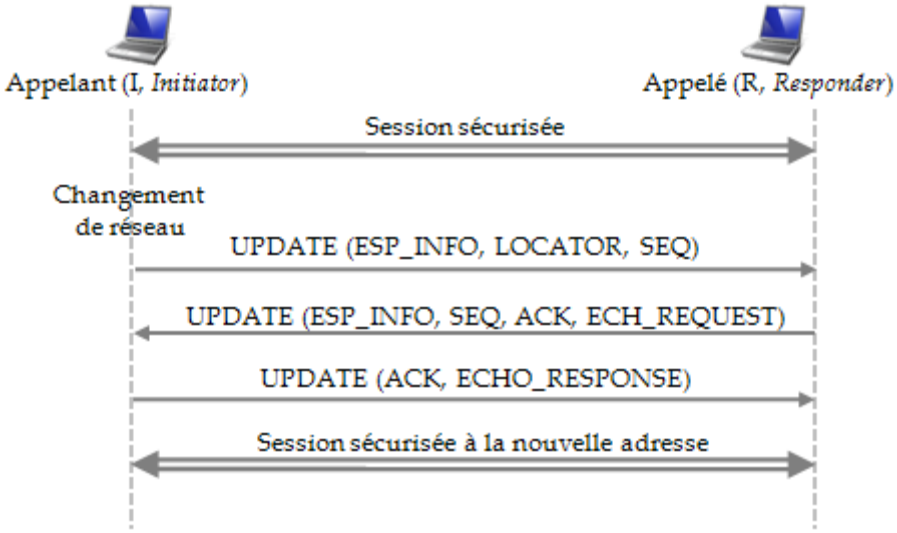


Figure 18 – Mobilité HIP

HIP offre aussi la possibilité de gérer le cas d'un utilisateur possédant plusieurs interfaces ou plusieurs adresses IP pour une même interface. Lors de la mise en place d'une connexion, le nœud concerné doit indiquer à son correspondant l'adresse qu'il préfère utiliser.

Cependant, deux problèmes de taille restent à régler : actuellement, la plupart des applications Internet qui veulent communiquer avec un correspondant traduisent d'abord un nom de domaine en une ou plusieurs adresses IP en interrogeant un serveur DNS tandis que, avec HIP, les applications utilisent des HITs. Il est donc nécessaire de trouver un moyen pour traduire un nom de domaine en HIT. De plus, pour permettre aux utilisateurs HIP mobiles d'être joignables après un changement de réseau, [51] a préconisé l'utilisation d'un mécanisme de « rendez vous » sans en préciser les mécanismes. Pour ces deux raisons, deux standards ont donc été mis en place pour pallier à ce problème : [55] qui introduit la notion de serveur de « rendez-vous » et [56] qui propose une extension aux mécanismes de DNS.

I.2.3.3. Conclusion sur l'architecture HIP

L'architecture HIP permet donc une gestion efficace de la mobilité d'un terminal en séparant localisation et identification. Cependant, la mise en place d'une telle solution nécessiterait des changements importants au niveau du noyau des systèmes d'exploitation, des modifications au niveau de chaque application utilisant l'Internet de près ou de loin ainsi que l'introduction de serveurs de « rendez vous ». Ceci semble donc difficilement réalisable à l'échelle mondiale dans les années à venir.

I.2.4. La gestion de la mobilité au niveau de la couche transport

La couche transport est principalement en charge de la gestion de bout-en-bout des communications. Ses fonctionnalités de base sont : au niveau de la source, elle s'occupe d'encapsuler les données provenant de la couche application (et éventuellement de les segmenter) pour les transmettre à la couche réseau et au niveau du destinataire, elle est en charge de réassembler les paquets en provenance de la couche réseau et de les transmettre à la couche application. Les principaux protocoles utilisés sur l'Internet sont TCP (*Transmission Control Protocol*) [57] qui fournit un transport fiable, orienté connexion et UDP (*User Datagram Protocol*) [58] qui fournit un transport non fiable en mode sans connexion. De plus, d'autres ont été développés par la suite pour s'adapter aux nouvelles applications.

Dans le cadre de la gestion de la mobilité, les solutions de niveau transport doivent aussi être informées par les couches inférieures de la détection d'un changement de réseau et de la reconfiguration d'une adresse IP. Une fois ces informations obtenues, elles doivent alors mettre en place des mécanismes pour maintenir la connectivité. Nous étudions donc, dans la suite de cette partie, les différentes solutions qui sont proposées pour résoudre ces différents problèmes.

I.2.4.1. Les solutions basées sur un proxy.

Les premières solutions, basées sur le protocole TCP proposent de gérer la mobilité d'un utilisateur en séparant la connexion TCP de bout-en-bout en deux connexions.

Ainsi, MSOCKS [59] se base sur une technique nommée *TCP Splice* [60] qui permet à un proxy de joindre deux connexions TCP en une seule de façon transparente. Le nœud mobile peut se connecter à un serveur distant en établissant une connexion avec le proxy de façon classique puis en lui indiquant l'adresse et le port du serveur auquel il veut se connecter. Le

proxy se connecte alors au serveur distant de manière classique. De même, si le MN veut accepter une connexion provenant d'un serveur distant, elle s'attache d'abord au proxy et lui indique quelle adresse et quel port il doit écouter. Dès que le serveur en question tente de se connecter, le proxy intercepte et met en place la connexion avec le serveur, puis prévient le MN que la communication va commencer. Lorsque le nœud mobile change d'adresse IP ou souhaite changer d'interface, il envoie un message RECONNECT au proxy pour mettre à jour sa localisation et dès que la connexion est réalisée, l'échange de donnée reprend (et les données qui ont été perdues sont retransmises).

De même, I-TCP [61] utilise une entité intermédiaire, le MSR (*Mobile Support Router*) localisée au niveau des points d'accès, pour diviser en deux la connexion TCP, sauf que cette entité varie avec les déplacements du MN. Ainsi, si le MN veut mettre en place une connexion avec un CN, deux connexions TCP sont mises en place, une entre le MN et le MSR1 et une autre entre le MSR1 et le CN. Lorsque le MN se déplace vers le MSR2 et reçoit un « beacon » de ce dernier, il s'associe à ce nouveau point d'accès et lui transmet les informations concernant ces connexions I-TCP actives ainsi que l'adresse du MSR1. Le MSR2 prépare les sockets nécessaires aux futures connexions et envoie un message au MSR1 pour le prévenir que le MN lui est associé. Le MSR1 gèle alors toutes les connexions correspondant au MN et envoie au MSR2 l'état de ces connexions. Le MSR2 peut alors relancer chaque connexion et les communications reprennent.

Une autre solution basée sur le même principe de proxy a aussi été proposée pour UDP : M-UDP [62].

Cependant, le même problème que pour HIP se pose : la localisation du MN, une fois qu'il s'est déplacé. Dans MSOCKS, le proxy permet cette localisation à partir du moment où le MN s'attache à ce dernier où qu'il se trouve. Mais, si le MN se déplace sur de longues distances, cela va conduire à un routage triangulaire pénalisant. En effet, même si le MN se trouve dans le même réseau d'accès que son correspondant, toutes les connexions TCP doivent passer par le proxy. Dans le cas de I-TCP et de M-UDP, aucun mécanisme de localisation n'est mis en place. Si le MN se déplace, seules ses communications en cours peuvent reprendre mais aucune nouvelle n'est possible. Cependant, ils préconisent l'utilisation couplée de leur solution avec Mobile IP qui peut jouer ce rôle.

I.2.4.2. TCP-Migrate

L'idée principale de TCP-Migrate [63] est basée sur la migration de connexions TCP lors d'un changement d'adresse IP. Cette migration est rendue possible par deux nouvelles options. La première, nommée option *Migrate-Permitted*, permet de négocier un identifiant de connexion (ou *token*) à l'établissement d'une connexion TCP, cette négociation d'identifiant pouvant être sécurisée par un échange de clef de type Diffie-Hellman [52]. La connexion peut alors être identifiée soit classiquement par {adresse source, port source, adresse destination, port destination} soit par {adresse source, port source, *token*} pour chacun des deux correspondants. Un MN peut alors relancer une connexion TCP précédemment établie après avoir changé de réseau en envoyant un paquet SYN contenant la deuxième option mise en place dans cette solution : l'option *Migrate*, qui contient le token de la connexion. Le CN doit alors resynchroniser la connexion avec le MN à sa nouvelle adresse. L'état de la connexion

étant maintenue, toute retransmission nécessaire est alors effectuée de façon standard. De plus, toute option ayant été négociée à l'établissement de la connexion reste en vigueur et n'a pas besoin d'être renvoyée dans le SYN *Migrate*.

En ce qui concerne la localisation d'un MN, TCP-Migrate préconise l'utilisation des mécanismes de DNS dynamique. Ainsi, lorsque le MN change d'adresse IP, il doit le détecter et changer l'association « nom d'hôte/adresse » (A-record pour IPv4) au niveau du DNS grâce aux mécanismes décrits dans [64] et [65].

Cette proposition permet donc une gestion de la mobilité des connexions TCP d'un utilisateur en nécessitant tout de même des modifications au niveau du protocole TCP. Mais aucune nouvelle infrastructure réseau n'est nécessaire pour déployer cette solution.

I.2.4.3. Les solutions basées sur SCTP

Plusieurs solutions ont plus récemment proposé une gestion de la mobilité en utilisant le protocole SCTP (*Stream Control Transmission Protocol*) [67] et plus particulièrement la possibilité qu'il offre de gestion du *multi-homing*, c'est-à-dire la possibilité d'établir une connexion avec plusieurs adresses au niveau de chaque extrémité (éventuellement répartie sur plusieurs interfaces) et de *multi-streaming*, qui permet d'associer plusieurs flux à une seule connexion SCTP. Ainsi dans le cas d'une connexion multiple (*multi-homed*) qui signifie donc qu'un utilisateur peut être atteint par plusieurs adresses IP, l'association SCTP établie entre deux points terminaux pourra être de la forme {[@IP1, @IP2 : Port1] : [@IPx : Port x]} ce qui signifie que le point terminal 1 peut communiquer avec le point terminal 2 par l'intermédiaire de l'une ou l'autre de ces adresses @IP1 et @IP2 en utilisant le port 1. Pour établir une association SCTP, une procédure en 4 temps est définie : les messages INIT, INIT-ACK, COOKIE-ECHO et COOKIE-ACK sont ainsi successivement échangés entre les deux points terminaux qui peuvent ensuite échanger des données en utilisant les différentes adresses qui ont été précisées pendant la procédure d'établissement.

I.2.4.3.a mSCTP ou mobile SCTP

Une première proposition de gestion de la mobilité par SCTP, nommée mSCTP, a été faite par [33] en utilisant l'extension permettant la reconfiguration dynamique d'adresse [68]. Cette extension permet à une extrémité SCTP d'ajouter ou de supprimer une adresse à l'association SCTP ainsi que de changer l'adresse primaire utilisée pour l'association SCTP alors même que celle-ci est active, ce qui était impossible auparavant. Ces opérations sont respectivement amorcées par les événements ADD, DELETE et CHANGE au cours d'une association. L'extrémité concernée envoie alors un message SCTP ASCONF (Address Configration Change) à son correspondant pour la prévenir du changement. La Figure 19 présente le principe de base de mSCTP. Il considère qu'une association entre l'adresse IP1 du MN et l'adresse IP3 du CN a été initiée (cf Figure 19.a). Le MN se déplace alors de l'AR1 vers l'AR2 et, lorsqu'il se trouve dans la zone de chevauchement entre les deux zones de couverture (cf Figure 19.b), la procédure de handover se déroule en quatre parties :

- L'obtention d'une nouvelle adresse IP auprès de l'AR2 par les mécanismes d'autoconfiguration IPv6, si l'on considère l'utilisation d'IPv6.

- L'ajout de l'adresse IP2 à l'association SCTP : l'extrémité SCTP du MN prévient alors l'extrémité SCTP du CN par le chemin 1 qu'il va ajouter une nouvelle adresse IP, l'adresse IP2, à l'association SCTP. Ceci est réalisé par l'envoi d'un message ASCONF (avec l'option « Add IP Address ») auquel le CN répond par un message ASCONF-ACK.
- Le changement d'adresse IP primaire : le MN doit utiliser son adresse IP2 comme adresse IP primaire et prévient donc le CN du changement par un nouvel échange de ASCONF/ASCONF-ACK (avec l'option « Set Primary IP Address »), par l'un ou l'autre des chemins.
- La suppression de l'ancienne adresse à l'association SCTP : le MN échange avec le CN les messages ASCONF/ASCONF-ACK (avec l'option « Delete IP Address ») à travers le chemin 2.

Le MN et le CN communiquent alors à travers le chemin 2 (cf Figure 19.c).

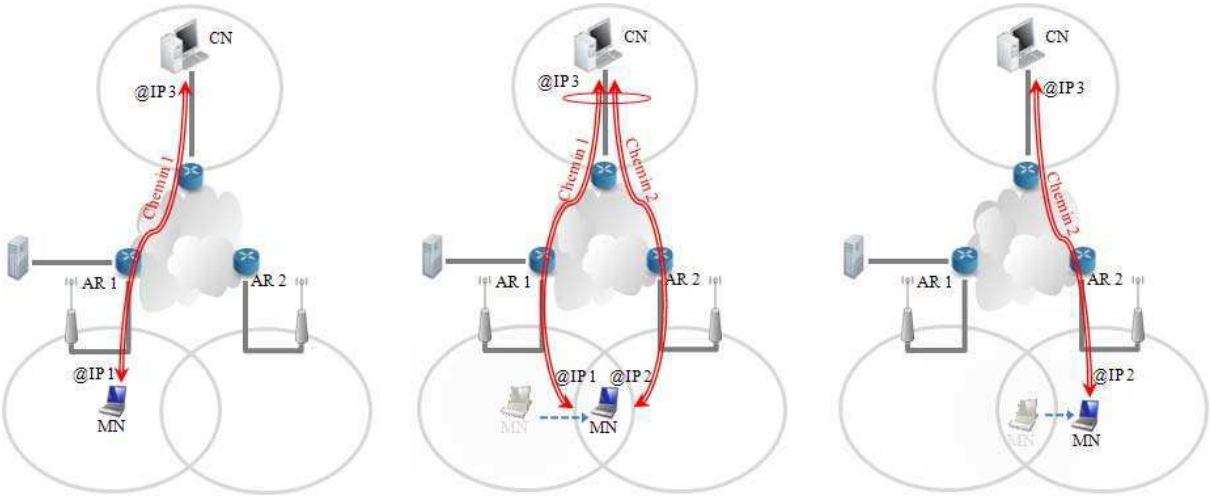


Figure 19 – Principe de mSCTP

Cependant, bien qu'elle offre théoriquement une mobilité sans interruption, cette solution ne prévoit pas de gestion de la localisation du MN, une fois qu'il s'est déplacé.

I.2.4.3.b SIGMA

Une solution proposée par [66] sous le nom de SIGMA (*Seamless IP diversity based Generalized Mobility Architecture*) permet alors de résoudre le problème de localisation du MN dont souffre mSCTP. Cette solution propose aussi une architecture capable de gérer la mobilité en utilisant les principes du *multi-homing* de SCTP ainsi que son extension de reconfiguration dynamique d'adresse. Mais cette fois, elle propose aussi un mécanisme de localisation basé sur la mise à jour de l'association nom d'hôte/adresse du DNS par les mécanismes de DNS dynamique décrits par [64] et [65]. Cette mise à jour de la localisation se fait parallèlement au changement d'adresse IP primaire.

I.2.4.3.c Conclusion sur les solutions basées sur SCTP

Ces solutions ont donc en commun l'utilisation du multi-homing inhérente à SCTP. Elle se base sur l'hypothèse que le MN a le temps d'obtenir une nouvelle adresse IP et de l'ajouter à l'association SCTP avant de perdre la connectivité depuis l'ancienne adresse IP. La validité de cette hypothèse dépend en fait de la taille de la zone de chevauchement et de la vitesse de déplacement du MN. L'utilisation du DNS dynamique comme système de localisation a aussi l'avantage de ne pas nécessiter de nouvelles infrastructures contrairement à Mobile IPv6 et ses extensions. De plus, SCTP permettrait une transition douce d'IPv4 vers IPv6 puisqu'il permettrait l'utilisation privilégiée d'adresse IPv6 si les deux entités les possèdent mais conserverait la possibilité d'utiliser les adresses IPv4. Cependant, la grande majorité des applications actuelles ne sont toujours pas prévues pour fonctionner sur SCTP, alors que sa première spécification est sortie en Octobre 2000.

I.2.4.4. Conclusion sur la mobilité au niveau transport

Les solutions de mobilité les plus évoluées au niveau transport permettent de résoudre un certain nombre des problèmes dont souffrent les solutions de niveaux réseaux. En effet, TCP-Migrate, mSCTP et SIGMA permettent de maintenir des communications en cours actives durant un handover sans nécessiter la mise en place de nouvelles infrastructures. De même, à aucun moment, ces solutions ne souffrent de routage triangulaire ; le routage est toujours optimisé. Enfin, ils fonctionnent aussi bien sur IPv4 que sur IPv6 ; donc la transition serait grandement facilitée.

Cependant, contrairement aux solutions de niveau réseau où l'utilisation de tel ou tel protocole de transport est transparente, ces solutions de niveau transport permettent seulement de gérer le maintien des communications qui utilisent effectivement le protocole associé à la solution. Ainsi, un protocole tel que FTP qui fonctionne sur TCP ne pourra être maintenu par une solution de mobilité basée sur SCTP. De même, la transmission de la voix ou de la vidéo qui se fait généralement sur RTP puis sur UDP ne peut être maintenue par une solution de mobilité basée sur TCP. Pour une solution complète, il faudrait donc qu'une extension pour la mobilité soit faite pour chaque protocole de transport.

De plus, les modifications de la pile TCP préconisées par TCP-Migrate seraient difficiles à mettre en œuvre dans l'architecture réseau réelle et une solution de mobilité basée sur SCTP impliquerait auparavant que les applications soient modifiées pour focntionner sur ce protocole, ce qui n'est pas le cas à l'heure actuelle.

I.2.5. La gestion de la mobilité au niveau de la couche application avec SIP

Dans le modèle en couche que nous utilisons comme référence (voir I.2), la couche application correspond aux trois couches hautes du modèle OSI, qui sont, rappelons le, les couches session, présentation et application. Cette couche application est aussi l'objet de nombreuses études concernant la gestion de la mobilité et ce, essentiellement à travers le protocole d'initiation de session (SIP, *Session Initiation Protocol*) [69] qui présente des fonctionnalités de base très utiles dans un contexte de mobilité. Dans cette partie, nous

présentons d'abord les principes de base de ce protocole avant d'expliquer comment il peut être adapté pour la gestion de la mobilité.

I.2.5.1. Principes de base de SIP

SIP est un protocole de signalisation standardisé par l'IETF conçu pour établir, modifier et terminer des sessions avec un ou plusieurs participants. Sa principale utilisation réside actuellement dans la gestion des sessions de voix sur IP (VoIP, *Voice over IP*), dont il est actuellement le standard ouvert le plus répandu, mais le fait qu'il soit indépendant du type de données transmises et du type de protocole utilisé lui permet aussi de mettre en place de nombreuses autres applications telles que la messagerie instantanée, la vidéoconférence, l'enseignement à distance, les jeux vidéo ou encore la réalité virtuelle. Par exemple, les données de type audio ou vidéo sont généralement transmises sur RTP [70], lui même encapsulé dans UDP. Les paramètres de ces sessions sont alors décrits par le protocole de description de session (SDP, *Session Description Protocol*) [71].

De plus SIP, en encodant ses paquets sous forme de texte, a été conçu de manière très semblable à HTTP (*Hypertext Transfert Protocol*) [72] et SMTP (*Simple Mail Transfert Protocol*) [73], les deux protocoles ayant le plus de succès sur l'Internet, ce qui rend le développement de nouveaux services et le débogage plus facile qu'avec d'autres protocoles tels que H323. Il a d'ailleurs été choisi par le 3GPP (*3rd Generation Partnership Project*), principalement pour ces raisons, comme protocole de contrôle de session pour l'architecture IMS (*IP Multimedia Subsystem*) [74] et de nombreuses entreprises privées telles que Cisco, Microsoft ou encore IBM l'ont aussi retenu comme protocole de contrôle de session.

Dans une architecture SIP, un utilisateur est identifié par une URI (*Uniform Resource Identifier*) SIP semblable à une adresse mail (par exemple sip:utilisateur@domaine.fr), ce qui permet de dissocier l'identification et la localisation des utilisateurs, fonctionnalité très importante dans le cadre de la mobilité comme on a pu le voir précédemment. D'autre part, SIP se base sur différentes entités logiques :

- **L'agent d'utilisateur** (UA), l'extrémité localisée sur le terminal utilisateur qui reçoit et traite les requêtes au niveau de l'UAS (*User Agent Server*) et génère et envoie les requêtes au niveau de l'UAC (*User Agent Client*). Par abus de langage, un UA SIP peut aussi être nommé « client SIP ».
- **Le proxy**, une entité intermédiaire qui joue à la fois le rôle de serveur et de client dans le but de transmettre les requêtes des clients. Les requêtes sont soit traitées en interne, soit transmises à d'autres serveurs, parfois après avoir été modifiées. Un proxy peut aussi interprèter un message SIP et, le cas échéant, le reformuler avant de l'acheminer.
- **Le serveur de redirection** (*Redirect Server*), un serveur qui permet de rediriger une requête SIP vers une ou plusieurs autres URI.
- **Le registrar**, un serveur qui traite les requêtes REGISTER d'un domaine donné dans le but de mettre à jour une base de données de localisation des clients SIP (ou UAs SIP) de ce domaine.

Souvent, le proxy, le serveur de redirection et l'unité d'enregistrement sont réunis en une seule entité qu'on nommera proxy SIP dans la suite de ce mémoire.

Cette architecture permet donc à SIP de gérer aussi la localisation et la disponibilité des utilisateurs.

I.2.5.2. Exemple de session SIP

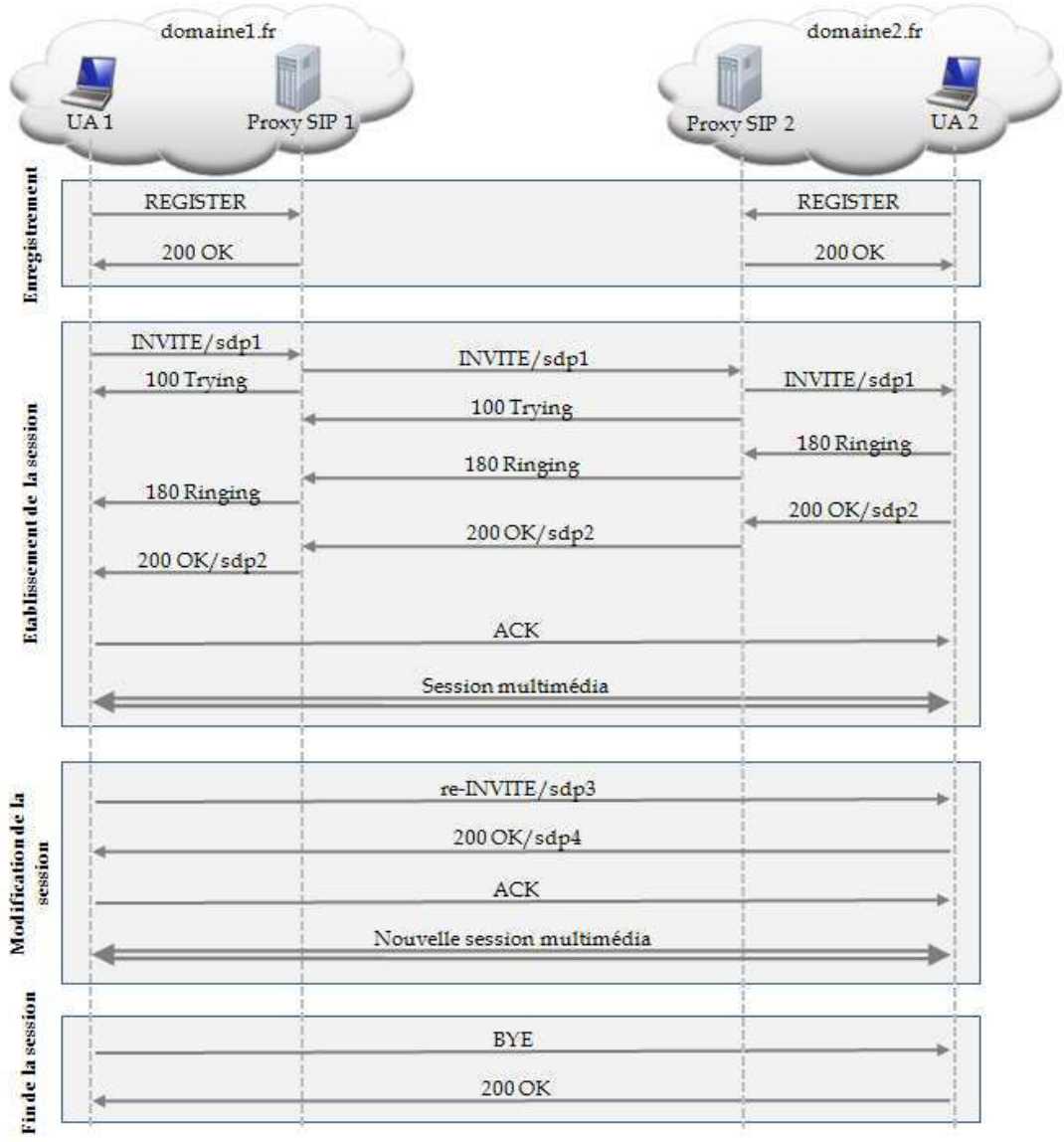


Figure 20 – Exemple de session SIP

La Figure 20 illustre un exemple de session SIP dans lequel deux clients SIP (ou UAs) appartenant à deux domaines distincts (donc gérés chacun par un proxy distinct) veulent établir une session SIP. Les deux clients commencent par s'enregistrer auprès de leur proxy SIP respectif (qui fait aussi office de *Registrar*) en effectuant un échange de message REGISTER/OK qui permet au proxy SIP de mettre en place une association SIP URI/adresse IP. Soit les clients connaissent l'adresse de leur proxy SIP, soit ils peuvent l'obtenir via DNS par exemple. L'UA1 veut ensuite établir une session avec l'UA2 dont la SIP URI est sip:ua2@domaine2.fr. Il envoie alors un message INVITE contenant les informations nécessaires à la mise en place d'une session dont les plus importantes sont :

- Son URI et celle de l'UA2
- Le Call-Id de la session qui en sera l'unique identifiant
- Le contact auquel l'UA2 peut le joindre directement (cela peut être sous la forme d'un *Fully Qualified User Name* comme par exemple pc10.domaine1.fr ou d'une adresse IPv4 ou IPv6).
- Les descripteurs de session SDP qui peuvent inclure les types de média utilisés (audio, vidéo, etc.) et la liste de codecs utilisables (GSM, H263, etc.) mais aussi la version du protocole utilisée, l'identifiant de session ou encore l'adresse e-mail et le n° de téléphone de l'UA1.

Lorsque le proxy SIP 1 reçoit ce message, il cherche l'adresse du proxy en charge de domaine2.fr, soit dans une base de données, soit par les mécanismes de DNS définis par [75], puis lui transmet le message. Le proxy SIP 2 le reçoit et constate que l'UA2 que l'on cherche à joindre est à sa charge. Il lui transmet donc le INVITE. Les messages 100 Trying sont envoyés pour prévenir l'entité précédente que le INVITE a bien été reçu puis transmis. Cependant, ces messages ne sont transmis que lorsque le proxy concerné est en mode *Statefull*, c'est-à-dire qu'il conserve en mémoire les transactions réalisées tout au long de la session et qu'il peut décider, selon ces informations, de modifier le routage d'une requête SIP par exemple. Au contraire, un proxy *Stateless* ne conserve aucune transaction et ne fait que transmettre les requêtes. Il ne peut donc pas, par exemple, distinguer une retransmission d'un message original et ne peut décider lui-même de retransmettre un message.

Lorsque l'UA2 reçoit le message INVITE, il renvoie un 180 Ringing pour indiquer à l'UA1 qu'il a bien été informé de l'appel (par une sonnerie par exemple).

Si l'UA2 accepte d'établir une session, il envoie un message 200 OK contenant le Call-Id, son contact et le descripteur de session modifié en fonction des possibilités et des souhaits de l'UA2. Ce message est alors acheminé jusqu'à l'UA1 de la même manière que pour le INVITE. Finalement, l'UA1 répond par un ACK pour indiquer qu'il a bien reçu le 200 OK et la session peut commencer. On peut remarquer sur la Figure 20 que le ACK est envoyé directement de l'UA1 à l'UA2. En effet, les deux UA connaissent maintenant mutuellement leur adresse et peuvent donc communiquer directement. Cependant, il est possible d'obliger chaque message SIP à passer par les différents proxies SIP traversés. On reviendra sur cette possibilité dans la suite du mémoire car elle nous sera utile.

Au cours de la session, les utilisateurs ont la possibilité d'en modifier les paramètres pour par exemple changer de codec vidéo parce que la qualité n'est pas suffisante. Ceci est réalisé par l'envoi d'un re-INVITE contenant le Call-Id de la session et un nouveau descripteur de session, ce re-INVITE étant exactement du même format qu'un INVITE. La nouvelle session peut alors démarrer une fois que les messages 200 OK et ACK ont été échangés.

Enfin, lorsque l'un des UAs souhaite mettre fin à la session, il envoie un message BYE au correspondant qui répond par un 200 OK.

La procédure de désenregistrement auprès du proxy SIP ne figure pas sur le schéma mais celle-ci est réalisée de la même façon que pour l'enregistrement : un message REGISTER, quasiment semblable au message d'enregistrement est envoyé avec comme seule différence un champ « Expires » à 0.

On peut remarquer, pour les messages OK, Trying et Ringing, l'ajout d'un chiffre devant leur nom. Ce chiffre correspond en fait aux codes de réponse déjà définis par HTTP/1.1 [72]. Par exemple, les codes 1xx correspondent à des codes informationnels tandis que les 2xx indiquent la réception avec succès d'une requête.

I.2.5.3. Gestion de la mobilité nomade

Une des fonctionnalités de base de SIP est de pouvoir gérer la localisation de ses utilisateurs. Il permet donc de gérer la mobilité nomade d'un client SIP mobile (mobilité sans communication en cours) [1]. Ceci est réalisé directement par les serveurs de redirection (voir Figure 21).

On prend ici l'exemple d'un MN qui se déplace de son réseau mère (1.domaine.fr) vers le un réseau visité (2.domaine.fr) sans aucune session SIP en cours. Une fois ce déplacement réalisé, le MN doit alors se réenregistrer auprès de son proxy (en charge de tous les sous domaines de domaine.fr) pour mettre à jour sa localisation. Après ce déplacement, le CN cherche à joindre ce MN et lui envoie donc une requête INVITE à destination de l'URI sip:mn@1.domaine.fr. Le serveur de redirection répond alors par un message du type *302 Moved Temporarily* en indiquant une autre URI à laquelle le CN peut le joindre. Dans notre exemple, il lui indique l'URI sip:mn@2.domaine.fr. Le CN, qui reçoit le message *302 Moved Temporarily*, échange alors classiquement les messages INVITE/200 OK/ACK en utilisant la nouvelle URI et la session peut démarrer.

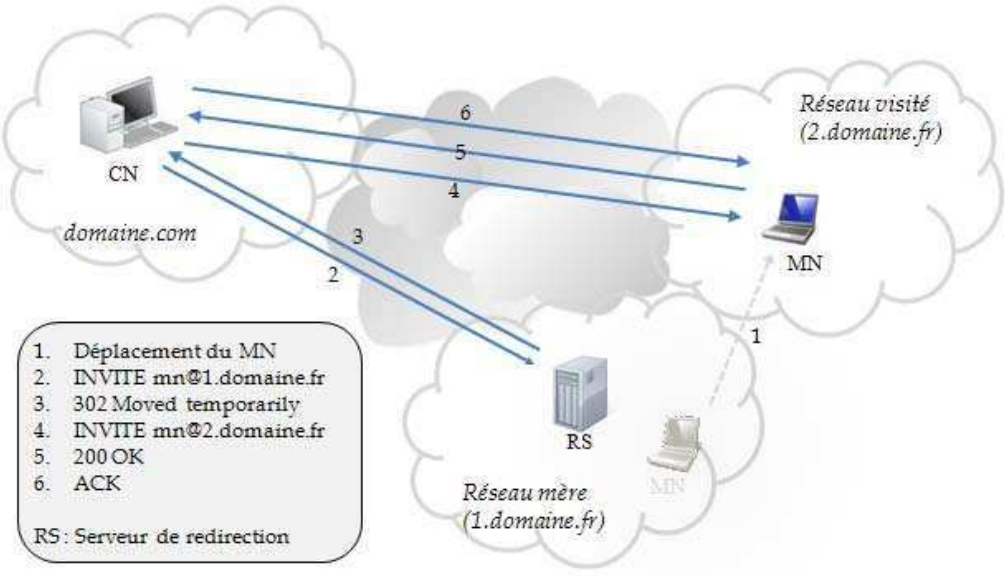


Figure 21 – Gestion de la mobilité nomade par SIP

Après un déplacement, cette solution considère que le MN doit seulement se réenregistrer pour mettre à jour sa localisation auprès du proxy SIP (et de son serveur de redirection) en charge de son réseau mère mais dans le cas où le MN se déplace vers un domaine sous la charge d'un autre proxy SIP, il peut aussi s'enregistrer auprès de ce dernier. Le mode opératoire de ce réenregistrement n'est pas précisé dans la norme pour laisser son implémentation ouverte. Cependant, plusieurs études ([76], [77]) ont analysé et comparé les différentes possibilités qu'a le MN de se réenregistrer et ce, pour différents cas.

I.2.5.4. Gestion de la mobilité continue

Bien que la gestion de la mobilité nomade soit un atout important offert par la mobilité SIP, ce qui nous intéresse essentiellement dans ce chapitre est la gestion de la mobilité continue qui permet de maintenir une communication en cours lors d'un changement de réseau, donc dans le cas de SIP, un changement de réseau alors qu'une session SIP est active (par exemple, une session de VoIP).

Pour résoudre ce problème, une solution a alors été proposée par [1] en se basant sur l'envoi d'un re-INVITE par un MN qui a effectué un changement de réseau en cours de session. Ce message re-INVITE est exactement du même format qu'un INVITE classique et possède le même Call-Id que l'INVITE initial. Cependant, l'adresse du MN doit être modifiée au niveau du champ « contact » ainsi que dans la description de session pour que la session reprenne depuis sa nouvelle localisation.

La description de session peut aussi changer au niveau des types de médias à transmettre, des codecs à utiliser, etc. ce qui représente un des avantages de la gestion de la mobilité au niveau application. En effet, la communication peut s'adapter au nouveau support en négociant de nouveaux paramètres, contrairement aux solutions de niveau inférieur qui sont transparentes pour les applications.

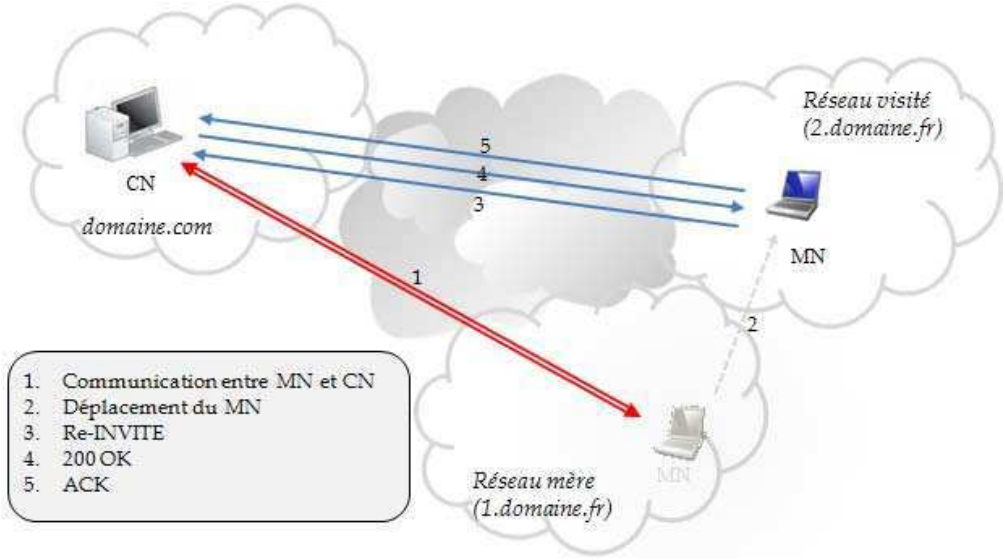


Figure 22 – Gestion de la mobilité continue par SIP

De même que pour les solutions de niveau transport, la solution de mobilité basée sur SIP permet d'éviter le passage par un routage triangulaire, ne nécessite pas plus d'infrastructures qu'une architecture SIP classique et fonctionne aussi bien avec IPv4 ou IPv6. De plus, la localisation des utilisateurs est une fonctionnalité de base de ce protocole et SIP permet aussi de gérer la mobilité de session, la mobilité personnelle et la mobilité de service [1].

Cependant, tout comme les solutions de mobilité de niveau transport, la mobilité basée sur SIP permet seulement de gérer la mobilité des applications contrôlées par SIP lui-même. Mais cette solution, contrairement à celles de niveau transport, présente l'avantage de transporter des informations qui peuvent se réveler très utiles dans la gestion de la QoS des applications à fortes contraintes temporelles qu'elle permet d'initier. Cependant, elle doit

impérativement être couplée à une autre solution pour permettre une gestion de la mobilité pour tous les types d'applications.

I.2.6. Les améliorations possibles

Dans cette partie, nous allons étudier deux mécanismes permettant une amélioration des solutions de gestion de la mobilité : le standard MIH (*Media Independent Handover*) et le principe de *multi-homing* déjà introduit pour SCTP.

I.2.6.1. MIH : un standard pour faciliter l'échange d'informations entre les couches

Le groupe de travail IEEE 802.21 [78] définit actuellement le standard MIH pour optimiser les mécanismes de mobilité en assistant les handover entre différentes technologies d'accès (IEEE 802.3, 802.11, 802.16, 3GPP, etc…) mais aussi à l'intérieur d'une même technologie d'accès. Pour ce faire, le standard permet d'utiliser des informations provenant aussi bien du terminal mobile que de l'infrastructure réseau et de les échanger via des interactions spécifiques entre les couches inférieures (Physique et Liaison) et les couches supérieures (Réseau, Transport et Application). Ces échanges d'informations se font à travers une entité appelée « fonction MIH » (ou MIHF) qui interagit avec les couches inférieures au travers de points d'accès au service (SAPs, *Service Access Points*) spécifiques à chaque média et avec les couches supérieures au travers d'une interface unifiée, nommée MIH-SAP. Pour cela, trois différents services sont définis:

- Le MIES (*Media Independent Event Service*) qui fournit des services aux couches supérieures en signalant les événements locaux (au sein du terminal) et distants (en provenance d'éléments du réseau). Ainsi les événements locaux se propagent depuis les couches inférieures vers la MIHF puis vers les couches supérieures d'un terminal tandis que les événements distants peuvent se propager depuis un élément du réseau vers la couche MIH ou une couche supérieure du terminal. Ces événements peuvent par exemple être du type : *Link Up* (lien connecté), *Link Down* (lien déconnecté), *Link Handover Imminent* (changement de lien imminent), etc.
- Le MICS (*Media Independent Command Service*) dont les commandes peuvent être échangées de façon locale ou distante (terminal/réseau ou réseau/réseau entre l'ancien et le nouveau réseau au cours d'un handover par exemple) depuis les couches supérieures vers le MIHF puis vers les couches inférieures. Ce service utilise les informations fournies par le MIES pour exécuter des commandes telles que *MIH Handover Initiate* (terminal/réseau : initie les handovers et envoie une liste des réseaux et points d'accès suggérés) ou encore *MIH Handover Prepare* (réseau/réseau : la MIHF de l'ancien réseau prévient celle du nouveau de se préparer à une procédure de handover).
- Le MIIS (*Media Independent Information Service*) qui fournit une architecture et des mécanismes permettant à la MIHF de collecter des informations sur les réseaux environnants. Ces informations peuvent concerner le type de réseau (GSM, 802.11e, etc.), le canal à utiliser, l'adresse MAC du point d'accès, le débit, la QoS ou encore la sécurité ainsi que d'autres informations concernant les services des couches

supérieures. Ces informations sont disponibles aussi bien pour les couches supérieures que pour les couches inférieures.

Les interactions locales entre ces différents services et la MIHF sont illustrées par la Figure 23.

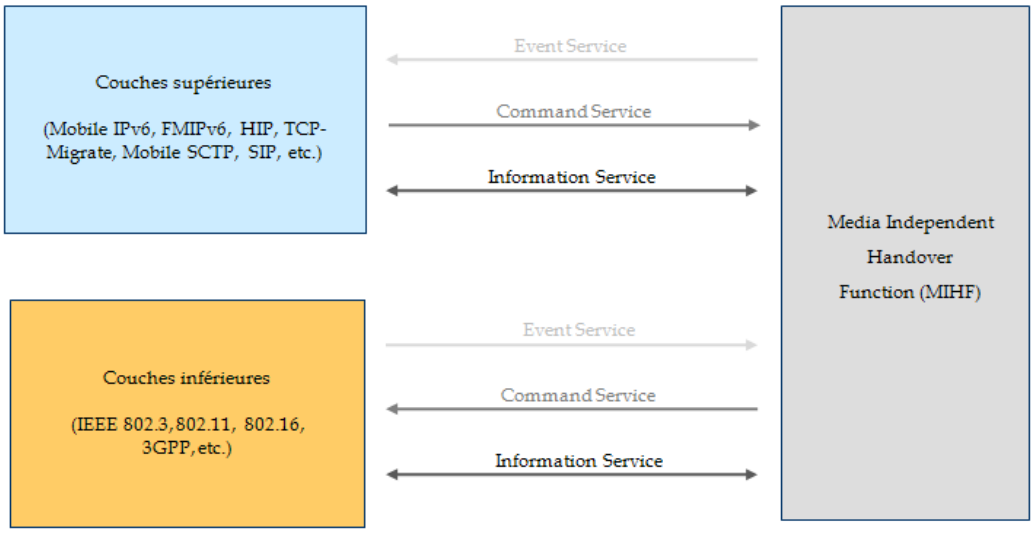


Figure 23 – Interactions entre les composants MIH

Lorsque les interactions sont distantes, les messages MIH sont transportés de façon spécifique à chaque type de média. Ainsi, la norme 802.11u définit leur utilisation sur des réseaux Wi-Fi tandis que la norme 802.16g la définit pour les réseaux WiMAX. Le groupe de travail MIPSHOP [79] de l'IETF étudie, quant à lui, le transport des messages MIH sur IP.

Des études ont alors été menées pour étudier les améliorations que pourrait apporter MIH aux protocoles de mobilité. Ainsi, au niveau de la couche réseau, [80] propose l'utilisation des services MIH pour optimiser la procédure de handover de FMIPv6. Au niveau de la couche transport, [82] propose une interaction entre SCTP et 802.21 pour améliorer les mécanismes de handover dans les réseaux à l'intérieur de campus pour les applications de VoIP. Enfin, au niveau de la couche application, [81] analyse les améliorations qu'apporterait l'interaction entre MIH et la mobilité SIP.

I.2.6.2. Le multi-homing et l'utilisation d'interfaces multiples : un pas vers la mobilité sans interruption

Le *multi-homing* permet à un terminal d'établir une connexion avec plusieurs adresses au niveau de chaque extrémité (éventuellement répartie sur plusieurs interfaces). Initialement utilisé pour permettre l'équilibrage de charge ou encore la redondance, cette caractéristique peut s'avérer très utile dans le cadre de la mobilité. En effet, un utilisateur qui se déplace entre différentes technologies d'accès peut configurer sa nouvelle interface alors que l'ancienne est toujours active et permet encore de recevoir et de transmettre des données : c'est ce qu'on appelle le principe du « make before break ». Cependant, ce principe est soumis à plusieurs conditions :

- L'ancien et le nouveau réseau doivent avoir une zone de chevauchement commune.

- Le passage par cette zone doit être suffisamment long pour permettre la configuration de la nouvelle interface et éventuellement la mise à jour de la localisation du terminal mobile.
- Le déplacement entre deux réseaux ayant la même technologie d'accès ne peut être réalisé que si le terminal mobile possède deux interfaces de cette technologie d'accès.

En considérant que ces conditions sont remplies ou que le changement de réseau implique deux technologies d'accès distinctes, le temps d'interruption peut donc être fortement réduit.

Plusieurs études ont ainsi été menées pour analyser l'impact du *multi-homing* sur les solutions de mobilité.

Au niveau de la couche réseau, [83] propose une extension à Mobile IPv6 en permettant à un MN qui se déplace vers un nouveau réseau, de continuer à communiquer à travers sa CoA1 (interface 1) tout en configurant une CoA2 sur son interface 2 et de mettre à jour son association avec son HA et son CN. Dès que cette association est faite, le CN et le HA voient cette nouvelle association comme si le MN avait changé d'adresse sur la même interface et les communications passent par la nouvelle interface. Dans ce cas là, les deux interfaces sont enregistrées auprès du même HA mais ne peuvent être utilisées simultanément. Pour permettre à un MN d'enregistrer plusieurs CoA simultanément, [84] a proposé l'utilisation d'un BID (*Binding Identification number*) associé à chaque nouvelle CoA. Une proposition a ensuite été faite pour adapter cette possibilité d'enregistrer plusieurs CoA à FMIPv6 [85].

Au niveau de la couche transport, on a déjà pu constater que les solutions utilisant SCTP se basent essentiellement sur ce concept de *multi-homing*.

Enfin, au niveau application, plusieurs solutions ont été proposées en se basant essentiellement sur le nouvel en-tête « JOIN » défini dans [86] permettant, entre autres, d'ajouter un nouveau participant à une session multimédia (ou d'audioconférence). Ainsi, [87] décrit un mécanisme dans lequel un client SIP mobile, équipé de deux interfaces et se déplaçant vers un nouveau réseau, envoie un message INVITE comprenant l'en-tête JOIN à son ancien proxy qui va alors intercepter et dupliquer les messages audio (et video) de la session pour les envoyer vers les deux interfaces simultanément. Une fois, le changement de réseau accompli, le MN envoie un re-INVITE à son CN pour qu'il puisse communiquer directement. Ce mécanisme est assez proche de ceux définis par FMIPv6. Dans [88], le re-INVITE est directement envoyé au CN avec l'en-tête JOIN depuis la seconde interface. Le CN envoie alors un message BYE vers la première interface du MN et la communication reprend en utilisant la deuxième interface du MN.

Dans tous les cas, ces solutions permettent de réduire considérablement le temps d'interruption et les pertes observées lors de changements de réseau et même, dans certains cas, de les annuler totalement pour fournir une véritable mobilité sans interruption, et donc imperceptible pour l'utilisateur. Cependant, comme on l'a déjà précisé, pour un déplacement entre deux réseaux IP distincts mais étant équipé de la même technologie sans fil (par exemple le Wi-Fi), un problème se pose car il faudrait que le terminal mobile soit équipé de deux interfaces Wi-Fi et, si l'on généralise, il faudrait qu'il soit équipé de deux interfaces de chaque technologie sans fil ce qui semble peu acceptable principalement en termes de consommation d'énergie.

I.3. Conclusion sur la mobilité

Ce premier chapitre a tout d'abord permis de préciser la terminologie utilisée dans le domaine de la mobilité dans les réseaux de communication. Ainsi, nous avons pu détailler les principaux types de mobilité existants ainsi que leur cadre d'application pour pouvoir ensuite positionner plus précisément notre étude vers la mobilité de terminal qui soulève actuellement le plus grand nombre de problématiques.

On a aussi pu constater que la mobilité de terminal, elle-même, pouvait être vue de différentes manières : ainsi, d'un point de vue administratif, elle peut être intra ou inter-domaine et d'un point de vue technologie d'accès, elle peut être intra ou inter-technologie. Cependant, le critère qui nous importe le plus est de savoir si la mobilité d'un utilisateur implique ou non un changement d'adresse IP, et, le cas échéant, si ce changement peut être géré localement à l'intérieur d'un réseau d'accès ou non. Nous avons donc aussi pu préciser quelle terminologie nous allions utiliser.

Ensuite, nous avons parcouru les différents niveaux du modèle en couche pour détailler le fonctionnement des principaux standards ou protocoles permettant la gestion de la mobilité de terminal, depuis la couche physique jusqu'à la couche application.

Ainsi, les technologies d'accès Wi-Fi et WiMAX, qui explicitent le fonctionnement des couches 1 et 2, ont été détaillées pour comprendre les principes de base permettant à un terminal mobile de s'associer à un point d'accès (ou à une station de base) et d'en changer en cours de communication. Cependant, on a vu que ces deux solutions nécessitaient naturellement des mécanismes de niveau réseau ou supérieurs pour permettre la gestion d'une mobilité locale ou globale, impliquant un changement d'adresse IP.

Les solutions appartenant aux couches supérieures, indépendantes de la technologie d'accès utilisée, ont donc été parcourues, chacune avec leurs avantages et leurs défauts. Ainsi, les protocoles de la couche réseau présentent l'avantage de pouvoir s'adapter à tout type d'application de façon transparente mais nécessitent des changements importants dans l'infrastructure réseau. De plus, la phase de tunnel bidirectionnel s'avére pénalisante dans bien des cas, particulièrement lorsque l'on considère l'utilisation d'IPv4 puisque l'optimisation de route n'est alors pas réalisable. Il faudrait donc attendre la mise en place d'IPv6 pour que ces solutions soient pleinement efficaces. De son côté, l'architecture HIP présente un concept innovant en séparant identification et localisation d'un utilisateur mais les changements nécessaires semblent trop importants pour être mis en œuvre. Enfin, aux niveaux transport et application, la plupart des solutions proposées présentent l'avantage de relancer la communication directement entre les correspondants et de ne pas nécessiter l'ajout de nouveaux éléments au réseau mais le principal inconvénient est qu'elles ne s'appliquent pas à tous les types d'applications. Par contre, ces dernières solutions situées au dessus de la couche réseau permettraient une transition douce entre IPv4 et IPv6 puisqu'elles sont toutes compatibles avec ces deux protocoles.

Enfin, des améliorations ont été proposées pour permettre d'une part l'échange d'informations entre les technologies d'accès et les protocoles de niveau réseau ou supérieur avec MIH et d'autre part l'adaptation des solutions de mobilité à un utilisateur possédant plusieurs interfaces et pouvant les utiliser simultanément.

On peut donc constater qu'aucune couche ne semble posséder tous les avantages nécessaires à la gestion de la mobilité en s'adaptant à tout type d'application. En effet, au cours d'un changement de réseau, les exigences d'une application permettant par exemple le téléchargement de fichiers sont différentes de celles d'une application de VoIP. Il convient donc, pour comprendre le problème de la mobilité de façon plus globale, d'étudier les mécanismes permettant la mise en place d'une qualité de service (QoS) adaptée à ces différents types d'applications et d'étudier les différentes possibilités de couplage de ces deux notions : la mobilité et la QoS.

Le deuxième chapitre de ce mémoire aura donc pour objectif de présenter les principales architectures de QoS existantes, pour pouvoir ensuite étudier leur couplage avec les solutions de mobilité présentées précédemment.

II. La Gestion de 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 video 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 ce chapitre, nous allons donc, dans un premier temps, introduire les principes fondamentaux de la QoS. Ensuite, les modèles précurseurs de la QoS, IntServ et DiffServ seront détaillés ainsi que trois protocoles de signalisation, COPS, NSIS et SIP, pouvant participer à la mise en place de la QoS. Enfin, nous étudierons différentes propositions d'architecture permettant la gestion de la QoS pour un utilisateur mobile.

II.1. Les principes fondamentaux liés à la QoS

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.

II.1.1. Les principales métriques

Selon le standard ISO 8402 [89], 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 [90] 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 [91], 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 comme définies dans [92]:

- 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.

II.1.2. 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 1 présente les différentes recommandations de l'ITU-T [93] 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.

Tableau 1 – Recommandations G1010 de l'ITU-T pour les applications audio et vidéo

Application	Degré de symétrie	Débits 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 vocale	Unidirectionnel	4 – 32 kbit/s	Lecture : < 1 s Enregistrement : < 2 s	< 1 ms	< 3 %
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 %

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.

II.1.3. Exigences de QoS pour les applications de données

Le Tableau 2 présente les exigences de QoS recommandées par l'ITU-T [93] 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.

Tableau 2 – Recommandations G1010 de l'ITU-T pour les applications données

Application	Degré de symétrie	Quantités de données typiques	Délai aller	Taux de perte
Navigation Web HTML	Unidirectionnel	~ 10 Ko	Idéal : < 2 s/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: < 4 s	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 %

II.2. Les principaux modèles existants pour garantir la QoS

Deux propositions majeures ont été faites par l'IETF pour garantir la QoS et ainsi assurer le bon fonctionnement des services IP, temps réel ou non : IntServ (*Integrated Services*), puis DiffServ (*Differenciated Services*) associé chacun à un groupe de travail du même nom.

II.2.1. Le modèle IntServ

II.2.1.1. Principes de base

Les travaux du groupe de travail IntServ ont abouti en 1994 à la définition d'une architecture de services intégrés [94] composée essentiellement de deux éléments : un modèle de service étendu, appelé modèle IS et un cadre d'implémentation de référence permettant la mise en œuvre de ce modèle.

Cette architecture, permettant de prendre en charge la QoS sans modifier le protocole IP, se base sur une réservation de ressource par flux. Chaque routeur doit alors conserver l'état des flux qui le traversent, ce qui modifie fondamentalement le fonctionnement de l'Internet, qui, au contraire, se basait jusqu'à présent sur une conservation de l'état des flux au niveau des terminaux utilisateurs. Les routeurs se voient alors ajouter 4 fonctionnalités supplémentaires :

- L'**ordonnanceur** de paquet, en charge de l'acheminement des différents flux de paquets, utilisant un ensemble de files ainsi que d'autres mécanismes tels que les timers.
- Le **classifieur** qui réalise la correspondance entre un paquet entrant et la classe de service à laquelle il est associé. Le niveau de QoS fourni par chaque classe de service est programmable pour chaque flux.
- Le **contrôle d'admission** qui implémente l'algorithme de décision que le routeur utilise pour déterminer si un nouveau flux peut ou non obtenir la QoS demandée sans dégrader les garanties précédemment offertes.
- Le **protocole d'établissement de réservation**, nécessaire pour créer et maintenir l'état des flux au niveau des routeurs. Le protocole choisi pour réaliser cette fonction est RSVP (*ReSerVation Protocol*), défini un peu plus tard comme *Resource reSerVation Protocol* dans [95].

Deux nouvelles classes de service sont alors définies en plus du best-effort qui ne reçoit aucun traitement spécifique au niveau des routeurs :

- Le service garanti (GS, ***Guaranteed Service***) [96] permet d'obtenir des garanties en termes de bande passante et de délai maximal d'acheminement des paquets, exprimables quantitativement. Si le flux respecte les paramètres réservés, ce service garantit que tous les paquets arriveront avec un délai maximal et qu'ils ne seront pas perdus dans les files d'attente en cas de congestion. Ce service se prête à des applications temps réel ayant de fortes contraintes de délai telles que les applications de vidéoconférence ou de VoIP. Cependant, aucun délai moyen n'est garanti, c'est donc à l'application elle-même de gérer au niveau du récepteur les variations de ce délai en utilisant des mécanismes de bufferisation.
- Le service de contrôle de charge (CL, ***Controlled-Load***) [97] est un service exprimable qualitativement en termes de bande passante qui assure à l'utilisateur que son flux de données sera transmis avec une QoS proche de celle d'un réseau non surchargé (non congestionné).

Les garanties sont obtenues de bout-en-bout par la concaténation de ces garanties offertes séparément par chaque routeur traversé le long du chemin. De plus, comme on l'a précisé précédemment, le protocole permettant de configurer ces routeurs est RSVP.

II.2.1.2. Le protocole RSVP

Dans cette partie, nous précisons le principe de fonctionnement du protocole de réservation de ressource RSVP.

Dans un premier temps, un message PATH est propagé depuis l'émetteur (application émettrice) vers le récepteur. Ce message contient la spécification du trafic (TSPEC) qui sera généré par l'application. Cette spécification ne peut être modifiée tout au long du chemin ; par contre, d'autres informations peuvent être ajoutées par l'intermédiaire de spécification additionnelle (ADSPEC) pour préciser des contraintes de ressources spécifiques. Une fois le message arrivé à destination, le récepteur répond par un message RESV qui contient la description du flux de trafic auquel la réservation de ressource doit s'appliquer (Receiver TSPEC) ainsi que les paramètres requis pour mettre en œuvre le service demandé (RSPEC). Ces descriptions peuvent aussi changer en cours de chemin.

Les messages RESV doivent suivre le chemin inverse des messages PATH et déclenchent la réservation effective des ressources (état à mémoriser au niveau de chaque routeur) si les mécanismes de contrôle d'admission de chaque routeur valident la requête. Si un routeur valide la requête, il crée et maintient un état correspondant à ce flux. Cependant, la durée de vie des réservations est limitée et les messages PATH/RESV doivent être périodiquement échangés pour que la réservation reste valable. Cela permet d'être robuste aux changements de routage entre autres.

Une fois la QoS configurée, lorsqu'un flux ayant fait l'objet d'une réservation traverse un routeur, il est identifié par son quintuplet {Adresse IP source, Adresse IP destination, Protocole, Port TCP/UDP source, Port TCP/UDP destination} par le classifieur. L'ordonnanceur s'occupe ensuite de la gestion des files.

Pour permettre la libération de ressource lorsqu'une application se termine, les messages « PathTear » et « ResvTear » sont respectivement envoyés pour indiquer aux routeurs de supprimer les états concernant la route et les états de réservation. D'autre part, des messages d'erreur, PathErr et ResvErr, indiquent des erreurs concernant respectivement le chemin et la demande de réservation et, enfin, le message ResvConf est envoyé par le dernier routeur recevant le message RESV pour confirmer au récepteur que la réservation a bien eu lieu.

II.2.1.3. Conclusion sur RSVP et le modèle IntServ

Le protocole RSVP est donc un protocole de signalisation qui permet de réserver dynamiquement de la bande passante et de garantir un délai maximal, et cela de bout en bout. Cette réservation, initiée par le récepteur, permet d'éviter que certaines applications émettrices monopolisent des ressources inutilement et permet dans le cas de communication multicast de différencier la réservation (et donc la facturation) par récepteur. De plus, son fonctionnement dynamique permet de s'adapter aux évolutions des communications (changement du nombre de participants, changements de route, etc.). Enfin, ce protocole présente aussi l'avantage de pouvoir s'adapter aussi bien à IPv4 qu'à IPv6 et passe de façon transparente les routeurs non RSVP.

Cependant, RSVP oblige à maintenir des informations d'état pour chaque flux au niveau de chaque nœud ou routeur le long d'un chemin reliant un émetteur à son récepteur. Donc, lorsque le nombre d'usagers et de flux augmente, le nombre d'états devient conséquent et le trafic est d'autant plus saturé que les rafraichissements de RSVP entre routeurs deviennent importants et créent de l'overhead. Le principal défaut du modèle IntServ et du protocole associé RSVP reste donc leur manque d'adaptation au facteur d'échelle, d'autant plus que la

réservation dans RSVP est unidirectionnelle. Donc, pour une application bidirectionnelle nécessitant de la QoS dans les deux sens, la quantité de messages est deux fois plus élevée. Le modèle IntServ/RSVP est donc plus adapté à des réseaux de petites tailles, tels que les réseaux locaux (LAN).

C'est pour cette raison notamment qu'un autre modèle d'architecture a été proposé : l'architecture DiffServ.

II.2.2. Le modèle DiffServ

II.2.2.1. Principes de base

Pour résoudre le problème de passage à l'échelle posé par la gestion de la QoS par flux dans les routeurs du cœur de réseau, le groupe de travail DiffServ a donc proposé ([98] puis [99]) de séparer le trafic par classes de service. Ainsi, le traitement par flux est repoussé au niveau des routeurs situés aux extrémités du réseau (ou routeurs de bordure) qui agrègent les flux par classe de trafic. Le nombre d'états maintenus par les routeurs de cœur est donc réduit au nombre de classes et non plus au nombre de flux, ce qui en réduit fortement la complexité.

Chaque classe de service est identifiée par une valeur codée dans un champ existant de l'en-tête IP, redéfini par le groupe DiffServ et nommé DSCP (*DiffServ Code Point*), ce qui présente l'avantage de ne pas nécessiter l'usage d'un protocole de signalisation supplémentaire. Il s'agit du champ TOS (*Type Of Service*) pour IPv4 et TC (*Traffic Class*) pour IPv6.

II.2.2.2. Domaines DiffServ, SLAs et « bandwidth broker »

L'architecture DiffServ se base tout d'abord sur le concept de domaine DiffServ qui consiste au regroupement d'un ou plusieurs réseaux soumis à une seule autorité administrative. Ce domaine est alors constitué de nœuds ou routeurs de cœurs qui ne sont connectés qu'à des nœuds appartenant au même domaine DiffServ et de routeurs de bordure qui interconnectent le domaine DiffServ avec d'autres domaines, DiffServ ou non. Un domaine DiffServ est représenté par la Figure 24. Les routeurs de bordure peuvent aussi bien jouer le rôle de routeur d'entrée lorsque le trafic entre dans un domaine DiffServ et de sortie lorsque le trafic sort du domaine.

Le client d'un domaine DiffServ (qui peut être soit un utilisateur, soit un autre domaine DiffServ) doit négocier avec le fournisseur de service en charge de ce domaine un contrat qui spécifie les termes et conditions de l'utilisation des services concernés : ce contrat est appelé un SLA (*Service Level Agreement*) et sa partie technique est spécifiée par différents SLS (*Service Level Specification*). Un SLA contient les informations suivantes :

- Le trafic que le client est susceptible de générer en termes de volume de données, de débit, de nombre d'utilisateurs, etc.
- La QoS que le fournisseur de service s'engage à offrir au client en termes de disponibilité, de sécurité, de fiabilité ou encore de performance (délai, bande passante, etc.).

- La politique suivie par le fournisseur de service en cas de dépassement de trafic (rejeté, accepté mais avec surtaxe, etc.).

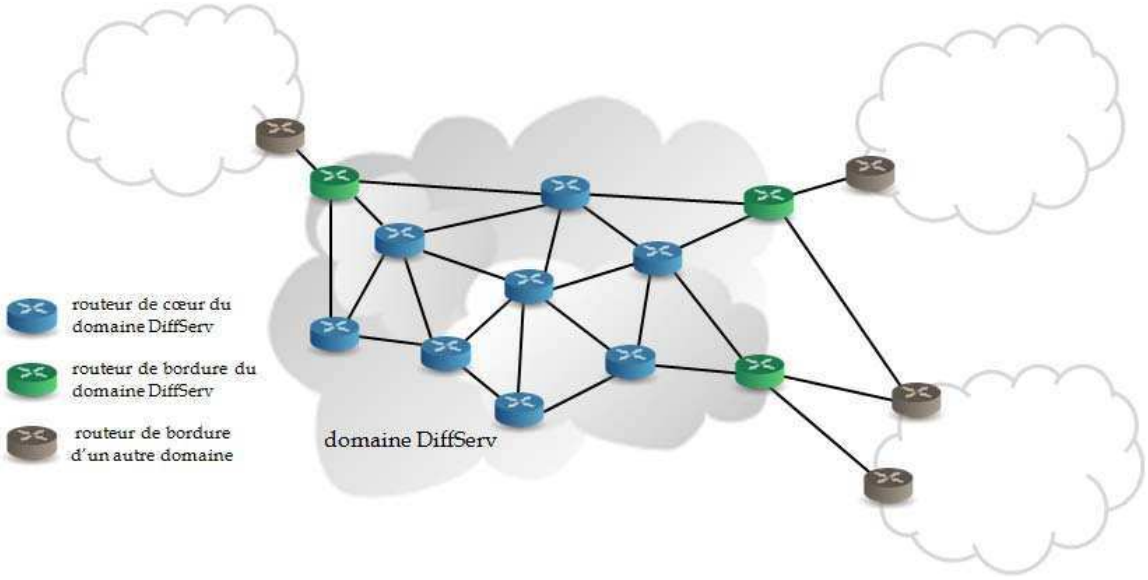


Figure 24 – Les routeurs dans un domaine DiffServ

Enfin, [100] définit une entité nommée *bandwidth broker* ayant une connaissance de la disponibilité des ressources et des politiques adoptées dans le domaine auquel il est associé. Une de ses tâches essentielles est donc le contrôle d'admission. De plus, pour permettre une allocation de bout-en-bout des ressources à travers les différents domaines tenant compte des différents SLAs négociés entre eux, cette entité doit communiquer avec les *bandwidth broker* des domaines voisins.

II.2.2.3. Gestion du trafic et notion de PHB dans un domaine DiffServ

Le fournisseur de service en charge du domaine DiffServ commence tout d'abord par dimensionner son réseau en fonction des différents SLAs contractés avec l'ensemble de ses clients.

Le traitement des paquets entrant dans le domaine DiffServ se fait alors au niveau des routeurs de bordure en charge de la classification des flux par classe de service et du conditionnement. Pour ce faire, ils sont composés d'un classifieur, d'un métreur, d'un marqueur, d'un régulateur et enfin d'un suppresseur de trafic non-conforme. Le classifieur identifie un flux à partir soit du champ DSCP uniquement, soit d'une combinaison de un ou plusieurs champs tels que l'adresse IP source, l'adresse IP destination, le DSCP, le protocole ID, les ports source et destination ou d'autres informations comme l'interface d'entrée. Ensuite, des mécanismes de profilage et de mesure de trafic permettent d'une part le marquage ou re-marquage des paquets et d'autre part la mise en forme des flux ou leur suppression totale ou partielle pour respecter les profils de trafic négociés.

Lors du marquage, le champ DSCP est mis à jour en utilisant l'une des différentes classes de service ou PHB (*Per Hop Behaviour*). En plus de la classe Best Effort qui ne fait l'objet d'aucun traitement spécifique, deux PHB ont été définis par l'IETF :

- **Expedited Forwarding (EF)** [101] qui correspond à la plus forte priorité et assure le transfert de flux à fortes contraintes temporelles (ex. VoIP, vidéoconférence) en garantissant une certaine bande passante ainsi que des délais, gigues et taux de perte faibles.
- **Assured Forwarding (AF)** [102] qui définit quatre niveaux de garantie (AF1, AF2, AF3, AF4) sur l'acheminement de certains paquets en cas de congestion.

Une fois que les paquets ont été marqués par les routeurs de bordure, ils sont traités à l'intérieur du domaine DiffServ par les routeurs de cœur en fonction du PHB codé dans le DSCP qui caractérise le comportement de relayage que doivent avoir ces routeurs. Les priorités sont alors assurées par des algorithmes d'ordonnancement tels que PQ (*Priority Queuing*), WFQ (*Weighted Fair Queuing*) ou encore CBQ (*Class Based Queuing*)

II.2.2.4. Conclusion sur le modèle DiffServ

Le modèle DiffServ offre donc une gestion de la QoS plus adaptée et plus réaliste que celui proposé par IntServ. La gestion par classe (ou par agrégat) permet en effet d'être beaucoup plus résistant au passage à l'échelle. De plus, Diffserv présente la particularité de ne pas nécessiter de protocole de signalisation comme RSVP en adaptant l'en-tête des paquets IP, ce qui permet un gain de bande passante.

Cependant, le dimensionnement d'un domaine DiffServ en fonction des différents SLAs contractés avec les domaines voisins et les utilisateurs est une tâche lourde et complexe à mettre en œuvre qui ne permet pas de s'adapter dynamiquement à des changements rapides de trafic. De même, le fait de ne pas utiliser de protocole de signalisation au niveau de l'application utilisateur implique que ce dernier n'est pas en mesure de modifier dynamiquement les ressources en fonction de ses besoins.

Pour essayer de résoudre ces problèmes et pour offrir une QoS de bout-en-bout, l'IETF a proposé une solution couplant les modèles IntServ et DiffServ [103] en se basant sur IntServ pour les réseaux de bordure (LAN, entreprise, etc.) et sur DiffServ pour les réseaux de cœur mais la plupart des problèmes spécifiques à chacun des modèles persistent.

II.2.3. DiffServ vs. Intserv

En conclusion, DiffServ est avantageux du point de vue du passage à l'échelle ainsi que de sa simplicité d'implémentation, mais manque de mécanismes de signalisations de bout-en-bout. L'applicabilité du modèle IntServ est limitée par le nombre d'informations de traitement et d'état à maintenir sur chaque nœud, ce qui le conduit à des problèmes de passages à l'échelle. Le Tableau 3 résume les caractéristiques de ces deux modèles.

Tableau 3 – DiffServ vs. IntServ

Caractéristiques	IntServ	DiffServ
Mise en œuvre de la QoS	Par flux	Par agrégat
Portée	Bout-en-bout	Domaine DiffServ (bordure à bordure)
Réservation de ressource	Contrôlé par l'application	Configuré au niveau des routeurs de bordure en se basant sur les SLAs
Signalisation	RSVP	Basé sur le DSCP transporté dans les en-têtes IP
Passage à l'échelle	Limité par le nombre de flux	Limité par le nombre de classes de service
Classe de Service	GS, CL et BE	EF, AF et BE
Complexité	Forte	Faible

II.3. Protocoles de signalisation pour la QoS

Les paragraphes précédents montrent donc que, bien qu'IntServ s'avère inadapté au passage à l'échelle, l'utilisation d'un protocole de réservation de ressources (RSVP) lui permet une gestion plus fine (adaptée aux besoins de chaque flux) et plus dynamique de la QoS. L'idée d'utiliser un ou plusieurs protocoles de signalisation pour négocier et établir une QoS de bout-en-bout a alors fait l'objet d'un certain nombre de travaux de recherche parmi la communauté scientifique.

L'objectif de cette partie est alors de présenter quelques uns de ces protocoles qui permettent la mise en œuvre de la QoS, les autres notions (sécurité, surveillance, etc.) n'étant pas abordées.

II.3.1. COPS et la notion de gestion de QoS par politique

Le groupe de travail RAP (*Resource Allocation Protocol*) de l'IETF a défini en 2000 une architecture basée sur la notion de politique pour améliorer les mécanismes de contrôle d'admission d'un réseau [104]. Une politique est définie comme un ensemble de règles permettant l'administration, la gestion et le contrôle d'accès aux ressources d'un réseau [105].

Chaque règle est alors associée à un ensemble de conditions auxquelles correspondent des séries d'actions à entreprendre dans le cas où ces conditions sont respectées. Pour permettre l'échange de ces politiques sur un modèle client/serveur, un protocole est aussi défini par ce même groupe de travail, le protocole COPS (*Common Open Policy Service*) [106].

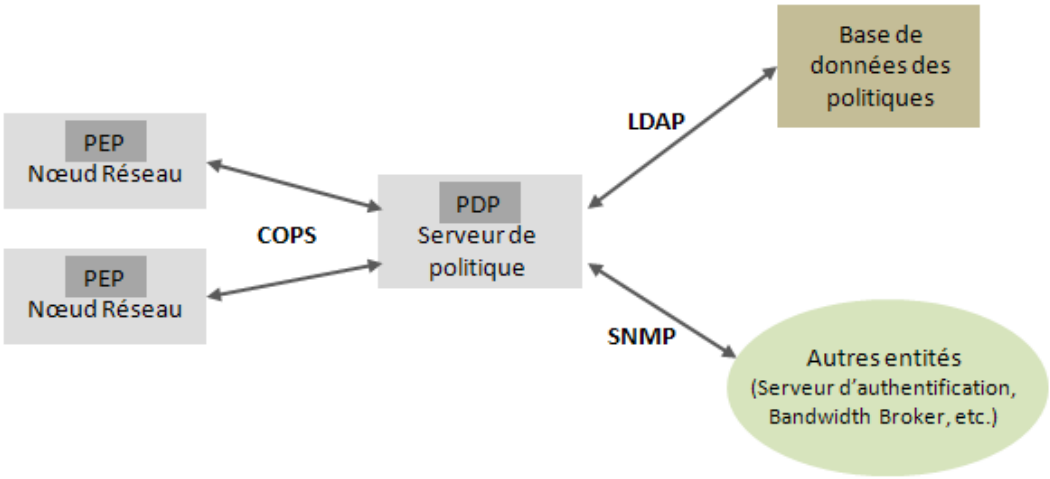


Figure 25 – Modèle à base de politique

Ce modèle de gestion par politique est composé de deux éléments centraux : le PEP (*Policy Enforcement Point*), en charge de l'application des décisions politiques et le PDP (*Policy Decision Point*), en charge de prendre les décisions basées sur la politique qui a été définie. Ces deux éléments communiquent au travers du protocole COPS. Pour prendre ses décisions, le PDP communique avec une base de données des politiques au travers du protocole LDAP (*Leightweight Directory Access Protocol*) et peut interroger d'autres entités telles qu'un serveur d'authentification ou encore un *bandwidth broker* en utilisant SNMP (*Simple Network Management Protocol*) par exemple. La Figure 25 résume ces différents échanges. En cas d'absence de PDP, une entité locale peut être utilisée au niveau d'un PEP, le LPDP (*Local PDP*).

Ce modèle générique de contrôle de réseau par politique permet de réaliser deux modes de contrôle distinctes : l'*outsourcing* et la *configuration (*aussi appelé *provisionning)*.

Dans le modèle *outsourcing*, un routeur (incluant un PEP) devant prendre une décision sur l'admission d'une réservation envoie une requête COPS au PDP et celui-ci renvoie la décision prise à partir des règles de politique. Il peut donc surtout être utilisé en relation avec le modèle IntServ lorsqu'un routeur reçoit un message RESV et doit décider d'accepter ou de refuser la réservation de ressource. En effet, cette décision peut nécessiter des informations autres que les ressources disponibles localement au niveau du routeur et l'utilisation d'un PDP peut donc s'avérer judicieuse.

Dans le modèle *configuration*, lorsque des événements extérieurs impliquent une modification de la configuration des routeurs (et donc des PEPs), le PDP peut communiquer à ces derniers des nouvelles règles à appliquer au travers du protocole COPS. Ceux-ci n'auront plus à solliciter le PDP avant de prendre une décision. Ce modèle peut donc pallier à la principale faiblesse de DiffServ dont la configuration des classes de service est statique. En

effet, un administrateur de réseau peut alors définir par exemple deux types de politique, l'une appropriée à la journée pendant laquelle un grand nombre de communications VoIP ont lieu et l'autre pour la nuit qui s'adapte plutôt à des sauvegardes de serveurs ou des téléchargements de données. Dans ce cas, les routeurs de bordure jouent le rôle de PEPs tandis que le *bandwidth broker* joue le rôle de PDP.

COPS a fait l'objet de différentes extensions telles que COPS-RSVP [107] qui spécifie l'utilisation de COPS dans un contexte IntServ/RSVP ou encore COPS-PR [108] qui s'oriente vers le provisionnement de politique de QoS dans un environnement DiffServ.

II.3.2. NSIS : un pas vers la signalisation générique

L'IETF a lancé en 2002 le groupe de travail NSIS (*Next Steps in Signaling*) [109] afin de définir un cadre générique pour la signalisation IP en tenant compte en premier lieu de la QoS mais aussi de la sécurité ou encore de la mobilité, etc.…

Pour tenter d'unifier la signalisation IP, le groupe de travail se base principalement sur les mécanismes du protocole RSVP en les simplifiant et en tentant de les appliquer pour mettre en œuvre un modèle plus général. Pour cela, l'architecture NSIS, définie dans [110], a été décomposée selon deux couches distinctes :

- La couche inférieure, NTLP (*NSIS Transport Layer Protocol*), dédiée au transport de la signalisation entre les différentes entités NSIS. Pour assurer ce rôle, un protocole nommé GIST (*General Internet Signaling Protocol*) a été spécifié [111]. Son principe de fonctionnement est le suivant : le niveau GIST s'occupe de transmettre les messages de signalisation à la prochaine entité NSIS après avoir établi avec elle une association négociée en trois phases (association qui pourra être réutilisée ultérieurement). Lorsque le message est reçu par l'entité NSIS suivante, celle-ci la transmet au niveau NSLP pour traitement puis transmet le message à la prochaine entité, et ainsi de suite jusqu'au récepteur final. Enfin, tout comme RSVP, GIST conserve des états au niveau des entités NSIS.
- La couche supérieure, NSLP (*NSIS Signaling Layer Protocol*), spécifique à chaque application de signalisation qui définit ainsi ses messages selon ses propres besoins. Différents études sont actuellement réalisées pour mettre en œuvre des protocoles de ce niveau dont :
  - QOS NSLP [112] qui, associé avec GIST, fournit des fonctionnalités similaires à celles de RSVP en les étendant quelque peu. Ce protocole se veut indépendant de l'architecture de QoS sous jacente et fournit un support pour différents modèles de réservation.
  - NATFW NSLP [113] qui définit un NSLP permettant de configurer les NATs (*Network Address Translators*) et les pare-feux en fonction des besoins des flux de données applicatifs. Pour l'instant, il permet aux hôtes localisés derrière un NAT d'obtenir une adresse publique joignable et aux hôtes localisés derrière un pare-feu de recevoir du trafic de données.
  - [114] qui analyse les effets que pourrait avoir la mobilité sur la série de protocole NSIS et tente de définir comment les protocoles opéreraient si ils étaient combinés

à des protocoles de mobilité tels que Mobile IPv4 ou Mobile IPv6 selon différents scénarios.

Dans sa version actuelle, NSIS suit une approche « on-path » qui implique que les entités de signalisation se trouvent sur le chemin de données. La signalisation est alors effectuée saut par saut entre les entités NSIS (NE), les dispositifs ne supportant pas NSIS acheminant simplement les messages sans les traiter. La Figure 26 résume ce fonctionnement. Chaque hôte ainsi qu'une partie des routeurs situés sur le chemin de données implémentent une entité NSIS (NE) qui échange des messages de signalisation.

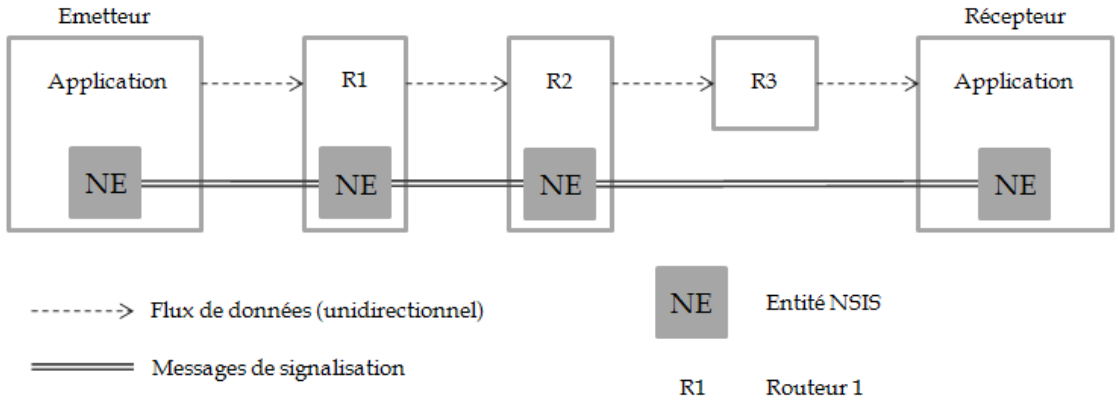


Figure 26 – Signalisation NSIS couplée au chemin de données

II.3.3. SIP : le contrôle de session au service de la QoS

Le protocole d'initiation de session SIP a déjà été présenté auparavant dans la partie I.2.5. Dans cette partie, nous nous intéressons plus particulièrement à la manière avec laquelle ce protocole peut être utilisé pour la réservation et la libération dynamique de ressources.

Le protocole SIP transporte effectivement des informations qui peuvent s'avérer très utiles lors de la réservation de QoS : les paramètres contenus dans les descripteurs de session. Ces paramètres sont négociés au cours de l'établissement de la session mais aussi lors des éventuelles modifications. Cette négociation peut être directement réalisée par les clients SIP situés au niveau des utilisateurs mais ces derniers, bien qu'ils soient conscients de certains paramètres importants tels que les codecs qu'ils sont capables d'utiliser, n'ont pas de réel moyen de connaître l'état des ressources disponibles le long du chemin que leur communication va parcourir. De plus, cela impliquerait l'intégration à ces clients SIP de mécanismes de QoS tel que RSVP, COPS, etc., ce qui présente le double désavantage d'alourdir les clients SIP et de ne pas permettre aux clients SIP non évolués pour la QoS d'en profiter.

Ces nouvelles fonctionnalités ont donc toutes les raisons de s'intégrer plutôt au niveau des entités intermédiaires que nous avons déjà rencontrées : les proxies SIP. En effet, ceux-ci peuvent permettre aux opérateurs de connaître la durée de la session, le nombre de médias impliqués ainsi que les codecs utilisés et leurs caractéristiques associées (bande passante, etc.). A partir de ces informations, une gestion automatique de la réservation/libération des ressources et du contrôle d'admission en fonction de la politique adoptée par l'opérateur devient donc réalisable et aucune modification n'est nécessaire au niveau des clients SIP.

Deux modes d'établissement de session avec réservation de QoS peuvent alors être distingués :

- Le mode « enabled » dans lequel l'établissement de la session et la réservation des ressources sont réalisés parallèlement mais ne dépendent pas l'un de l'autre. La session démarrera quel que soit le résultat de la réservation. Par exemple dans le cas d'un réseau DiffServ, la QoS sera de type BE si la réservation a échoué et EF si elle a fonctionné.
- Le mode « assured » dans lequel l'établissement de la session ne se fait que si la réservation de QoS a bien été réalisée.

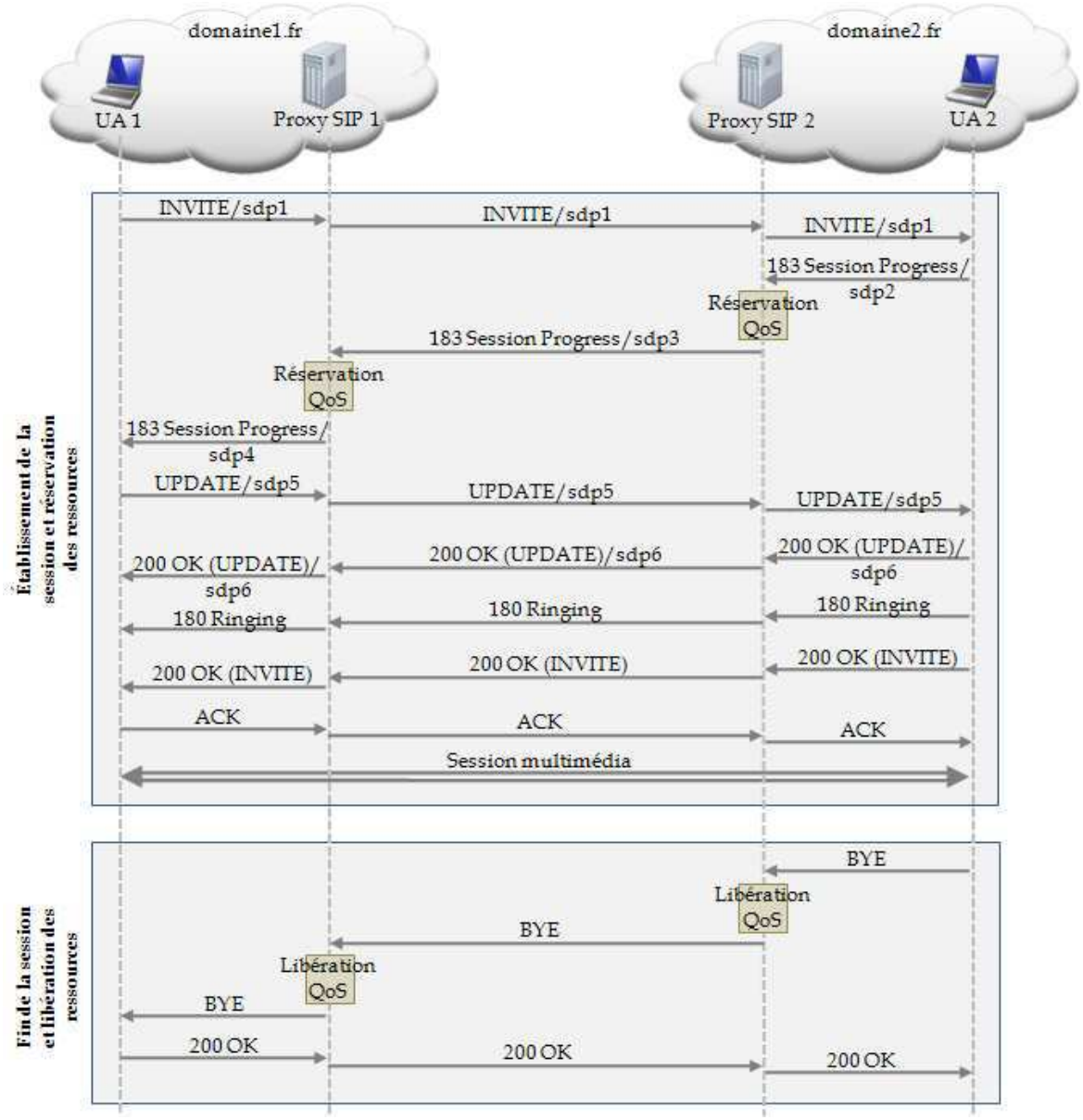


Figure 27 – Exemple de session SIP avec réservation/libération de ressources

Un standard [115] détaillant le fonctionnement de ces deux modes a d'ailleurs été proposé. Il se base sur des nouveaux messages tels que le Session Progress, le UPDATE ou le PRACK ainsi que sur un ensemble de préconditions ajoutées aux descripteurs de session. Les clients SIP peuvent alors préciser si la mise en place de la QoS est « mandatory » (correspond

au mode « assured ») ou « optional » (mode « enabled ») pour chaque sens de la communication (réception, émission) et si la QoS doit être e2e (*end-to-end*, de bout-en-bout) ou locale (au niveau des réseaux d'accès). Par contre, la RFC 3312 [115] considère que la QoS est mise en œuvre par les participants de la session SIP tandis que nous considérerons que celle-ci est mise en œuvre par l'intermédiaire des proxies SIP. La Figure 27 illustre alors un exemple de session SIP intégrant la réservation et la libération de ressources en mode « assured ». Les messages PRACK/200 OK (PRACK) ne sont pas présentés pour rendre la figure plus compréhensible, ils sont normalement échangés à la réception d'un message 1xx pour le fiabiliser. On peut aussi remarquer que dans cet exemple, tous les messages sont échangés via tous les proxies SIP, ceci justement dans le but de gérer la QoS le long du chemin. Enfin, les réservations et libérations de QoS de la Figure 27 peuvent par exemple représenter schématiquement l'échange de messages (COPS, RSVP, etc.) permettant aux proxies SIP de communiquer avec les entités en charge de la gestion des ressources à l'intérieur des domaines concernés. Par exemple, si entre domaine1.fr et domaine2.fr, on considère un domaine DiffServ, les proxies SIP pourront échanger des messages COPS avec le Bandwidth Broker en charge du domaine DiffServ. Ce dernier pourra alors configurer les routeurs de bordure de ce domaine pour prioriser les flux de la future session SIP.

De plus, quel que soit le mode d'établissement de la session, si une modification de session est réalisée par l'intermédiaire d'un message re-INVITE, les proxies SIP peuvent analyser les nouveaux paramètres et prévenir automatiquement les entités en charge de la gestion des ressources des changements en cours. La signalisation SIP permet donc une gestion beaucoup plus dynamique de la QoS pour les applications qu'elle permet de contrôler.

SIP, bien qu'il ait été conçu, à l'origine, pour permettre le contrôle de session peut donc être utilisé de façon très efficace dans la gestion dynamique de la QoS.

II.4. Architecture de QoS pour les réseaux de nouvelle génération

Sur la base des parties précédentes concernant la QoS, de nombreuses recherches et efforts de standardisation ont été menés pour proposer des solutions de gestion de la QoS de bout-en-bout dans des réseaux IP hétérogènes. Aussi connues sous le nom d'architectures pour les réseaux de nouvelle génération (NGN, *Next-Generation Networks*), elles visent à tenir compte des nouvelles contraintes de QoS des applications actuelles (en particulier, les applications multimédias ou de VoIP) ainsi que des exigences de gestion des fournisseurs de service Internet. Pour ce faire, ces architectures définissent les processus de provisionnement, d'invocation et de gestion de la QoS qui, dans un souci d'adaptation à l'échelle, sont répartis entre les différents domaines administratifs, ces derniers ayant une vision plus précise de leurs capacités. Des accords (de type SLA) doivent alors être passés entre ces différents domaines.

Dans le processus d'invocation de la QoS, ces architectures définissent généralement l'utilisation de protocole de signalisation, notamment SIP. Parmi celles-ci, les principales sont les suivantes :

- L'architecture NGN ITU-T qui se définit comme « un réseau basé sur le mode paquet capable de fournir des services de télécommunication et d'utiliser des technologies de

transport à large bande et à QoS et dans lequel les fonctions liées aux services sont indépendantes des technologies de transport sous-jacentes. Il offre aux utilisateurs un accès sans entrave aux réseaux et aux différents fournisseurs de services ainsi qu'aux services de leur choix. Il supporte la mobilité généralisée qui permettra un accès aux services permanents et indépendants de la localisation des utilisateurs. » [116]. Elle définit en fait une architecture fonctionnelle séparant services, transport et application sans pour l'instant préciser les implémentations.

- L'architecture IMS (*IP Multimedia Subsystem*) [117], initialement développée par le 3GPP (*3rd Generation Partnership Project*) pour les réseaux UMTS, qui propose une architecture de QoS de bout-en-bout en se focalisant toutefois sur la partie réseaux d'accès. Cette architecture a aussi pour but de permettre la convergence fixe-mobile, c'est-à-dire la possibilité d'utiliser une infrastructure commune pour les réseaux filaires et sans fil. Sa collaboration avec l'ETSI-TISPAN (*ETSI-Telecommunications and Internet converged Services and Protocols for Advanced Networking*) en a été l'élément déclencheur. De plus, l'IMS a choisi, entre autres, SIP comme protocole de contrôle de session et COPS comme protocole pour transférer les politiques entre PDPs et PEPs. Ces deux protocoles sont donc très importants dans la mise en place de la QoS, SIP faisant aussi l'objet de recherche pour gérer la partie mobilité des applications SIP (VCC, *Voice Call Continuity*).
- De nombreux projets IST ont défini des architectures orientées QoS. L'un des plus récents prenant en compte le concept de NGN est le projet EuQoS [118] (*End-to-end Quality of Service support over heterogeneous networks*) qui propose une architecture globale de gestion de la QoS de bout-en-bout dans un environnement Internet multi-domaine hétérogène. Ce projet se focalise essentiellement sur la signalisation inter et intra-domaine et sur le contrôle d'admission spécifique aux différentes technologies sous-jacentes en s'appuyant également sur des protocoles tels que COPS, NSIS et SIP.

II.5. Mobilité et QoS : les solutions existantes

Nous avons déjà pu voir dans le paragraphe précédent que les architectures NGN s'intéressaient particulièrement aux concepts de mobilités et de QoS. En effet, si un utilisateur fixe a actuellement besoin de QoS, il en est de même pour les utilisateurs mobiles. Cependant, cette mobilité rend encore plus complexe la gestion des ressources déjà suffisamment problématique. En effet, de nouveaux problèmes apparaissent :

- Changement de réseau entre des domaines hétérogènes implémentant différents types d'architecture de QoS (ex : IntServ => DiffServ).
- Changement de réseau entre différentes technologies offrant des ressources variables (ex : Satellite => Wi-Fi).
- Ré-établissement du chemin de réservation avec RSVP et reconfiguration et ré-autorisation des paramètres de QoS avec DiffServ.
- Libération des ressources dans le réseau précédent pour éviter une allocation inutilisée de bande passante.
- Signalisation dupliquée (signalisation de mobilité et de QoS).

- Dimensionnement des réseaux (difficulté de prévoir la quantité de ressources nécessaires en tenant compte des nœuds mobiles)

L'objectif de cette partie est alors d'étudier les solutions existantes permettant à un nœud mobile d'obtenir des garanties de QoS lors de ses déplacements, en se focalisant sur les solutions basées sur IPv6.

II.5.1. Le support de la QoS pour Mobile IPv6 et ses extensions

Le protocole Mobile IPv6 et ses extensions ont fait l'objet d'un certain nombre de propositions pour le support de la QoS. Un standard [119] a d'ailleurs été réalisé pour définir les exigences d'une solution de gestion de la QoS pour Mobile IP.

Des solutions ont tout d'abord été établies pour fonctionner dans le contexte d'une architecture IntServ. Ainsi, [120] propose une extension du protocole RSVP pour supporter la mobilité (MRSVP) et suggère une réservation anticipée, basée sur des proxies locaux et distants, dans les différents domaines que le mobile peut traverser durant sa communication (réservation active pour le chemin courant et passive pour les futurs chemins potentiels) mais plusieurs problèmes interviennent dont : 1) le nombre d'états au niveau des routeurs qui augmente considérablement avec le nombre de mobiles, ce qui rend cette solution encore moins adaptée au facteur d'échelle que RSVP, 2) la détermination des potentiels déplacements du mobile et 3) la non prise en compte de l'optimisation de route de Mobile IPv6.

De même, d'autres propositions proposent l'interaction d'une extension de RSVP avec HMIPv6 pour les services temps réel [121] basée sur une entité proxy gérant les réservations et pré-réservations en limitant ces dernières au domaine MAP courant, ou encore une autre extension appelée FH-RSVP fonctionnant avec F-HMIPv6 [122] en considérant uniquement les déplacements à l'intérieur d'un domaine MAP et en utilisant le MAP et les routeurs d'accès comme agents proxy de QoS. Cependant, ces solutions, bien qu'elles réduisent les pré-réservations sont essentiellement étudiées pour fonctionner uniquement à l'intérieur d'un domaine MAP et de plus, les problèmes de passage à l'échelle persistent.

Pour finir, l'utilisation de RSVP avec Mobile IPv6 présente certaines incompatibilités de base. Ainsi, elle nécessite la mise en œuvre d'un nouveau mécanisme, le *RSVP tunneling* [123], pour faire face aux encapsulations qui ont lieu lors des différentes phases de Mobile IPv6, ce qui ajoute encore de la complexité à ce type de solution. De même, RSVP suppose que les messages RESV suivent le même chemin que les messages PATH, ce qui n'est pas le cas lorsque le routage triangulaire est utilisé (le MN communique directement avec le CN tandis que le CN communique avec le MN par l'intermédiaire du HA).

D'autres études ont alors été menées sur Mobile IPv6 et ses extensions dans un environnement DiffServ. [124] a défini cinq grandes catégories de problèmes qui peuvent intervenir dans ce contexte : le dimensionnement du réseau dans un environnement mobile, le manque de dynamisme pour la reconfiguration des SLAs, la définition et sélection des SLAs qui dépend du mode utilisé par Mobile IPv6 (routage triangulaire, routage optimisé ou tunnel bidirectionnel), l'identification du flux d'un mobile (Mobile IPv6 rajoute des entêtes spécifiques) et la facturation.

[125] définit une architecture qui considère différents domaines administratifs (DAN) composés chacun d'un GQS (Global QoS Server) qui a une connaissance globale des ressources disponibles dans son domaine et de différents QLNs (QoS Local Node) localisés au niveau des routeurs de bordure ou dans des composants de réseau d'accès sans fil tels qu'une *Base Station*. Le nœud mobile (MN) interagit avec le GQS pour négocier de la QoS et ce dernier configure les QLNs qui agiront (marquage, politique, etc.) en conséquence. Les communications entre GQSs et entre GQS et QLN sont effectuées par l'intermédiaire du protocole COPS.

Ensuite, sur la base de cette architecture et de la nouvelle option IPv6 nommée « QoS Object » définie dans [126] et incluse dans les messages BU et BACK pour transporter les informations de QoS (IntServ, DiffServ, MPLS) des flux IP entre le MN et le CN, différentes propositions (dont [127] et [128]) ont été faites en modifiant quelque peu les différents éléments. Ainsi, elles considèrent aussi différents domaines administratifs composés chacun d'un GQA (Global QoS Agent) en charge de la gestion globale des ressources d'un domaine et de différents LQAs (Local QoS Agent) et HAs en charge respectivement de la QoS et de la mobilité de ces sous-réseaux du domaine administratif. Les communications entre GQAs (pour échanger les accords de QoS tels que les SLAs) et entre GQA et LQA sont elles aussi effectuées par l'intermédiaire du protocole COPS. Par contre, le MN négocie sa QoS avec le routeur de bordure localisé dans le réseau courant du MN. Cependant, ces solutions ne prennent pas en compte certaines phases implémentées par Mobile IPv6 au cours d'un déplacement de réseau (mise à jour de la CoA auprès du HA, tunnel bidirectionnel ou phase triangulaire) et les nombreux échanges entre les entités impliquées allongent fortement le temps d'interruption des communications. De plus, ces solutions impliquent d'importants changements dans l'infrastructure réseau (nouvelles fonctionnalités Mobile IPv6 au niveau des routeurs d'accès, nouvelles fonctionnalités au niveau des routeurs de bordure pour interpréter les « QoS Object) en plus des modifications nécessaires à Mobile IPv6.

Une autre solution proposée par l'IETF a ensuite été l'introduction d'un nouveau protocole : CTP (*Context Transfert Protocol*) [129] qui permet le transfert de contexte de QoS entre l'ancien et le nouveau routeur d'accès du MN. Cette solution permet d'éviter le renégociation des SLAs et le ré-établissement complet de la QoS lorsque le déplacement se produit à l'intérieur d'un domaine administratif. Cependant, la prise en compte de la QoS n'est faite qu'au niveau du nouveau routeur d'accès et non pas de bout-en-bout. Une proposition pour pallier à ce problème a alors été faite [130]. Elle définit l'utilisation du transfert de contexte tout en permettant une mise en place de la QoS de bout-en-bout sur le nouveau chemin en transférant l'option « QoS Object ». Cette solution repose sur F-HMIPv6.

Cependant, toutes les solutions précédentes fonctionnant pour DiffServ se basent sur cette nouvelle option IPv6 qui n'a jamais été standardisée depuis 2000. Pour éviter son utilisation, une autre solution [131] couplant Mobile IPv6 et DiffServ s'appuie sur le protocole SIP pour mettre en œuvre la QoS mais présente l'inconvénient de nécessiter la définition d'un nouveau message SIP, appelé PRECONFIGURE.

Les solutions de mobilité basées sur Diffserv permettent donc d'éviter les problèmes liés aux incompatibilités que présente RSVP dans un environnement mobile, mais la plupart reposent sur une option qui n'a pas été standardisée et qui implique des changements

importants au niveau de l'infrastructure réseau. L'idée d'utiliser SIP pour mettre en place les mécanismes de QoS est toutefois intéressante, bien qu'elle implique la définition d'un nouveau message, ce qui ne semble pas indispensable.

II.5.2. Le support de la QoS pour mSCTP

Les études concernant le support de la QoS pour les solutions de mobilité de la couche transport ont essentiellement été réalisées pour mSCTP en tirant profit du *multihoming*. Parmi ces études, [132] présente une solution couplant mSCTP avec RSVP. Contrairement aux solutions basées sur MobileIPv6, la fonction de *multihoming* de mSCTP lui permet de ne pas nécessiter de changement au niveau du protocole RSVP. De plus, cette solution propose une méthode pour déterminer la partie commune du chemin emprunté par la communication avant et après le changement de réseau. Cependant, cette solution ne tient pas compte des communications bidirectionnelles et bien qu'aucune extension de RSVP ne soit nécessaire, le problème de passage à l'échelle inhérent à IntServ persiste.

Une autre étude [133] propose l'utilisation d'un proxy mSCTP pour mettre en œuvre la mobilité et la QoS en se plaçant dans une architecture IMS. Un tunnel mSCTP dans lequel sont encapsulés les sessions TCP/UDP est alors mis en place entre le MN (l'agent mSCTP localisé au niveau du MN) et le proxy mSCTP. Lors d'un changement de réseau, les mécanismes de mSCTP sont tout d'abord effectués pour ajouter la nouvelle adresse du MN ainsi que celle du proxy à l'association SCTP. Ensuite, un nouveau message SIP SWITCH est échangé entre le MN et le proxy pour que la session soit transmise en utilisant les nouvelles adresses. La QoS est ré-établie par l'activation des contextes PDP au niveau du MN, du proxy mSCTP et du CN en tenant compte des nouveaux paramètres SDP contenu dans le re-INVITE. Cependant, cette solution ne précise pas où doivent être localisés les proxies mSCTP à l'intérieur de l'architecture IMS alors que leur utilisation introduit un routage triangulaire qui peut s'avérer pénalisant au même titre que pour Mobile IPv6. Cette analyse est donc cruciale. De plus, cette solution nécessite aussi la définition d'un nouveau message SIP, ce qui rend sa mise en œuvre plus difficile.

II.5.3. Conclusion sur la gestion de la mobilité et de la QoS

Les solutions permettant la gestion de la QoS pour un utilisateur mobile sont donc essentiellement basées sur Mobile IPv6 et quelques propositions ont été faites pour mSCTP. Mais comme on l'a vu, ces solutions présentent des inconvénients majeurs qui rendent difficile leur implémentation dans un environnement réel que ce soit au niveau de la gestion de la QoS, de la mobilité ou des deux. Ainsi, la mise en œuvre d'une nouvelle option IPv6 semble difficilement envisageable, l'utilisation de solutions basées sur RSVP reste inadaptée au passage à l'échelle et l'utilisation de mSCTP ne permet pas de résoudre le problème pour les protocoles basés sur UDP ou TCP sans introduire un routage triangulaire pénalisant.

Par contre, on a pu voir que certaines solutions utilisaient SIP, en plus d'un protocole de mobilité, pour mettre en œuvre la QoS pour les applications interactives gérées par SIP. Or, on a déjà vu dans le paragraphe I.2.5.4 que SIP pouvait permettre la gestion de la mobilité de terminal pour ce type d'applications. Donc l'utilisation d'un protocole distinct tel que Mobile

IPv6 ou mSCTP n'est pas indispensable et SIP pourrait permettre à lui seul de gérer la mobilité ainsi que la qualité de service de façon efficace.

II.6. Conclusion

Ce deuxième chapitre nous a donc permis de délimiter les grands enjeux de la mise en œuvre d'une architecture de qualité de service dans l'Internet actuel.

En effet, la présentation des 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, a permis de comprendre que l'introduction de mécanismes de gestion des ressources devenait indispensable, particulièrement pour les applications interactives à fortes contraintes temporelles.

Nous avons ensuite présenté les principaux modèles précurseurs de la QoS : IntServ et DiffServ, pour comprendre leurs avantages et inconvénients respectifs. Ainsi, l'approche IntServ offre une gestion dynamique particulièrement fine de la QoS de bout-en-bout puisqu'elle permet, au travers de ses différentes classes de service, de définir exactement les besoins d'un flux applicatif. Cependant, la finesse due à cette gestion par flux de la QoS, rendue possible grâce à l'utilisation du protocole RSVP, est aussi à l'origine de son principal défaut : le non passage à l'échelle. En effet, la grande quantité de messages que cette solution nécessite ainsi que le grand nombre d'état que les routeurs de l'Internet doivent conserver en mémoire ne permettent pas son utilisation pour des réseaux de grande taille. Le modèle DiffServ a alors permis de résoudre ce problème de passage à l'échelle. Pour cela, le traitement par flux est repoussé au niveau des routeurs en bordure des domaines qui agrègent les flux par classe de trafic. Ces classes de trafic correspondent à différents niveaux de QoS prédéfinis par l'opérateur du domaine, ce qui rend cette solution relativement statique et beaucoup moins fine que IntServ.

Par la suite, nous avons détaillé des protocoles tels que COPS, NSIS ou SIP pour comprendre comment ces derniers pouvaient faciliter une gestion plus fine et plus dynamique de la QoS et comment ils pouvaient interagir avec les modèles DiffServ ou IntServ. Sur cette base, les architectures proposées actuellement pour offrir une QoS de bout-en-bout dans les réseaux de nouvelle génération ont été présentées.

Enfin, la dernière partie de ce chapitre aborde le point central de ce mémoire : la gestion de la QoS pour un utilisateur mobile. Différentes propositions sont alors présentées en parcourant les protocoles de mobilité présentés dans le premier chapitre.

Ces deux premiers chapitres nous permettent donc de constater l'intérêt que présentent certains protocoles dans la gestion de la mobilité et/ou de la QoS. C'est le cas tout particulièrement de SIP qui permet d'une part d'initier les processus de QoS de façon précise grâce aux descripteurs de session contenus dans ses messages et d'autre part de gérer de nombreuses formes de mobilité (dont la mobilité de terminal) grâce à son architecture complète et aux nombreux standards qui ont étendu ses fonctionnalités. Son utilisation quasi systématique dans toutes les architectures NGN permet d'ailleurs de le confirmer comme prépondérant dans la mise en œuvre de la QoS pour les applications interactives (de type VoIP, vidéoconférence, messagerie instantanée, etc…) et le fait qu'il fasse l'objet de

recherches approfondies quant à sa gestion de la mobilité pour ce type d'applications dans l'architecture IMS est aussi révélateur de l'intérêt que lui porte la communauté scientifique.

Le reste du mémoire vise alors à mettre en œuvre une architecture permettant la gestion de la QoS et de la mobilité des utilisateurs, en s'intéressant tout particulièrement au cas de réseaux de communication impliquant un système satellite géostationnaire de type DVB-S2/RCS. Pour cela, nous nous attacherons tout particulièrement à définir une solution permettant au protocole SIP de gérer efficacement la mobilité et la QoS des applications multimédias d'un utilisateur. Mais parallèlement, pour toutes les autres applications que SIP ne peut pas gérer, nous définirons également une solution plus globale de mobilité et de QoS en se basant sur Mobile IPv6 ou ses extensions comme protocole de gestion de la mobilité.

III. Propositions d'Architectures pour la Mobilité et la QoS dans un Système DVB-S2/RCS

Ce chapitre va nous permettre de présenter nos contributions concernant l'étude et la mise en œuvre de la qualité de service et de la mobilité dans un système satellite DVB-S2/RCS. Ces systèmes, qui permettent entre autres d'offrir un accès bidirectionnel aux utilisateurs se situant dans des zones non couvertes par les réseaux terrestres (filaires ou sans fil), doivent en effet s'adapter aux nouveaux services qui y sont proposés (QoS, mobilité, sécurité, etc.) pour pouvoir s'intégrer au mieux dans l'Internet. Toutefois, ces réseaux souffrent de caractéristiques pénalisantes telles que de longs délais de transmission et une bande passante limitée. Des études concernant l'impact de ces caractéristiques sur le fonctionnement des solutions de mobilité et de QoS doivent donc être réalisées.

Après avoir décrit brièvement le fonctionnement d'un système satellite DVB-S2/RCS et détaillé les objectifs du projet SATSIX dans le cadre duquel la majeure partie de nos travaux ont été réalisés, nous analyserons donc les exigences particulières et l'impact d'un tel système sur la QoS et la mobilité. Ensuite, sur la base des études réalisées, nous proposerons des architectures permettant d'intégrer des mécanismes de gestion de la QoS à des solutions de gestion de la mobilité. Nous nous focaliserons sur les solutions adaptées à IPv6 (en tenant cependant compte de leur capacité à s'adapter à IPv4) et permettant la gestion de la mobilité pour des applications multimédias à fortes contraintes temporelles, de type VoIP ou visioconférence, initiées par SIP et fonctionnant sur UDP. En effet, nous considérons que ces types d'applications sont les plus exigentes en termes de QoS et de mobilité et doivent donc être traités prioritairement. Dans le cadre de notre mémoire, nous ne traiterons donc pas les solutions de mobilité spécifiques à TCP. De même, nous n'étudierons pas les solutions basées sur SCTP puisqu'actuellement, très peu d'applications utilisent ce protocole et, d'autre part, ces solutions permettent uniquement de traiter le cas d'un terminal « multi-homed » ce qui ne couvre pas toutes les situations possibles (par exemple le déplacement entre deux réseaux Wi-Fi distincts pour un terminal équipé d'une seule interface Wi-Fi). Enfin, nous ne traiterons pas non plus le cas de HIP qui nécessite de nombreuses modifications tant au niveau de l'infrastructure réseau qu'au niveau des terminaux utilisateurs et des applications.

III.1. Les réseaux satellite DVB-S2/RCS

Les réseaux satellites géostationnaires ont pendant longtemps été essentiellement dédiés aux services de diffusion, notamment de la télévision tandis que l'accès Internet était principalement fourni par des technologies filaires. L'apparition des technologies xDSL a d'ailleurs confirmé cette tendance pour devenir rapidement la technologie prédominante en permettant d'offrir aux utilisateurs un accès Internet large bande peu coûteux dans les zones à forte densité de population. Cependant, ces technologies se révèlent peu rentables dans les zones faiblement peuplées ce qui a amené la communauté scientifique à tenter de trouver des solutions alternatives.

Les systèmes satellites se sont alors révélés comme la technologie la plus à même de relever ce défi grâce aux avantages inhérents qui la caractérisent : une faible infrastructure terrestre, une zone de couverture très large et un déploiement simple et rapide des terminaux satellites. La démocratisation des terminaux DVB-S (*Digital Video Broadcast via Satellite*) [134] a alors permis, dans un premier temps, de réduire significativement le coût des équipements en permettant à la communauté satellite d'adopter des standards communs et, dans un second temps, la standardisation d'une voie de retour par satellite DVB-RCS (*Digital Video Broadcast Return Channel via Satellite*) [135], en 1999, a totalement affranchi les systèmes satellites d'une voie de retour terrestre en offrant un accès bidirectionnel aux services IP large bande. Enfin, les avancées importantes réalisées dans le domaine des transmissions et techniques de codage, qui ont conduit à la standardisation de la norme DVB-S2 [136], successeur du DVB-S, ont confirmé l'intérêt suscité par les systèmes satellites au niveau des chercheurs et industriels de la communauté scientifique.

III.1.1. Les principales caractéristiques d'un système DVB-S2/RCS

Les principales caractéristiques d'un système satellite géostationnaire DVB-S2/RCS sont les suivantes :

- Une zone de couverture très large (un satellite géostationnaire peut couvrir jusqu'à 1/3 du globe) qui permet de couvrir des zones isolées telles que les zones rurales mais aussi des zones parcourues lors de déplacement sur de grandes distances dans des zones non terrestres (par avion, bateau, etc.)
- Une faible infrastructure terrestre et un déploiement simple et rapide qui permet de couvrir des zones ayant subi une catastrophe naturelle ou encore des zones reculées accueillant un événement ponctuel.
- Une capacité de diffusion inhérente qui se révèle particulièrement adaptée aux services de communication multipoints tels que la télévision, la radio, la vidéo à la demande (VoD, *Video on Demand*), etc.
- Un délai de propagation (sol→ satellite) incompressible de 125 ms. Un bond satellite (sol→ satellite→ sol) correspond alors à 250 ms.
- Une bande passante limitée en émission.

III.1.2. Les éléments d'un système DVB-S2/RCS

Un système satellite DVB-S2/RCS est composé des éléments suivants, présentés dans la Figure 28 :

- Une **Gateway** (GW) qui permet l'interconnexion du système satellite avec le cœur du réseau Internet.
- Des **Terminaux Satellites** dotés d'une voie de retour (RCST, *Return Channel Satellite Terminal* mais nous utiliserons ST par abus de langage) qui permettent d'interconnecter les utilisateurs au réseau satellite.
- Un **Satellite** qui peut être de deux types :
  - **Transparent** : dans ce cas, à la réception d'un signal sur la voie montante (sol → satellite), le satellite ne fait que convertir la fréquence et réémettre sur la voie

descendante (satellite → sol). La topologie du système est alors dite **étoilée** : toutes les communications sont centralisées par la GW qui émet en DVB-S2 vers les STs tandis que ces derniers émettent en DVB-RCS vers la GW. La communication entre deux STs nécessite alors un « double bond » (ST1 → satellite → GW → satellite → ST2).

- **Régénératif** : il permet alors d'effectuer un traitement à bord des signaux (*On Board Processing*). Dans ce cas, il peut convertir les signaux DVB-RCS en signaux DVB-S2 et effectuer des opérations de démultiplexage/multiplexage directement. La topologie est alors dite **maillée** : les STs peuvent communiquer directement les uns avec les autres par un « simple bond » (ST1 → satellite → ST2) et la GW peut alors être vu comme un ST classique sauf qu'elle garde tout de même sa fonction d'interconnexion avec le cœur du réseau Internet.

A un satellite sont associés un ou plusieurs spots montants et un ou plusieurs spots descendants. Chaque spot est associé à une zone d'émission (spot montant) ou de réception (spot descendant). L'utilisation de plusieurs spots présente différents avantages : tout d'abord, cela permet de réutiliser les mêmes fréquences d'un spot à l'autre ce qui augmente les capacités du système; d'autre part, l'utilisation de spots moins larges implique un signal plus concentré dont le gain est plus élevé ce qui permet de diminuer la taille des antennes.

- Un **Network Control Center** (NCC) qui est principalement en charge de la gestion des ressources (politique de QoS, SLA, etc.) du réseau satellite. Il gère aussi les procédures liées à la connexion d'un ST (authentification, contrôle d'admission) ou à la synchronisation du système. Le NCC est colocalisé avec la GW dans une topologie étoilée tandis que dans une topologie maillée, il constitue a priori un élément à part entière.

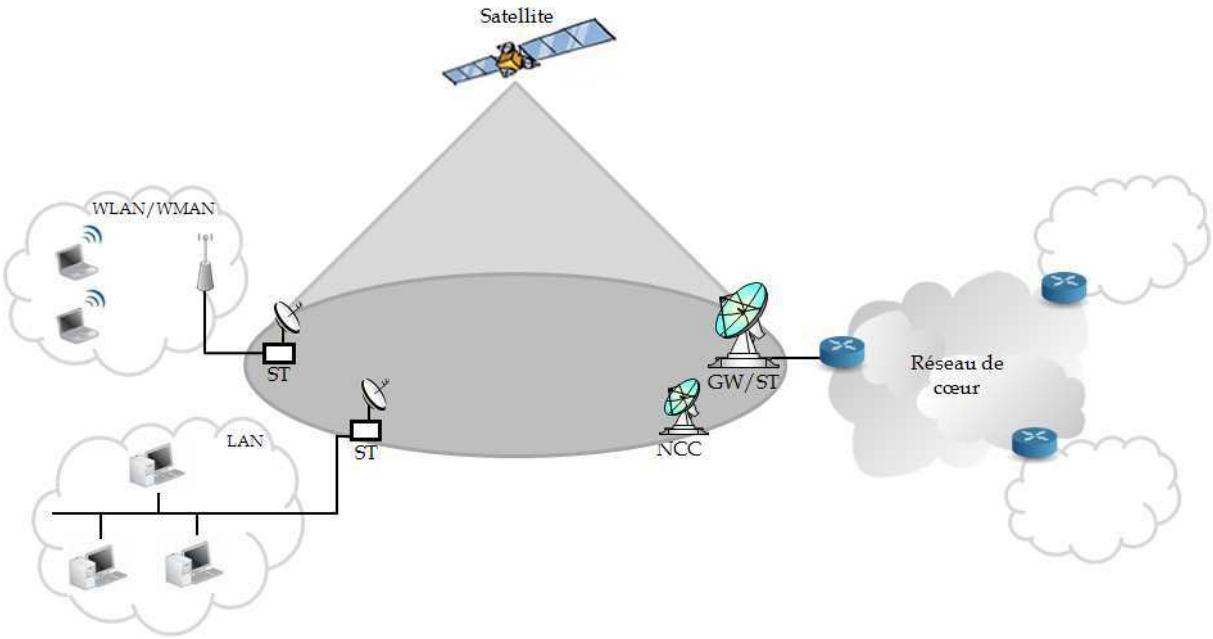


Figure 28 – Architecture de base d'un système satellite DVB-S2/RCS

Les utilisateurs finaux peuvent être directement reliés au réseau satellite par un ST ou peuvent être connectés par l'intermédiaire d'un LAN, d'un WLAN ou encore d'un WMAN, lui-même connecté à ce ST.

III.2. Le projet SATSIX

Le projet SATSIX [138] est un projet européen IST (*Information Society Technologies*) initié dans le cadre du 6ème Programme Cadre pour la Recherche et le Développement (FP6, *Sixth Framework Programme*, 2002-2006). Dans la continuité du projet SATIP6 [139], ses principaux objectifs sont de :

- Réduire les coûts de l'accès aux systèmes satellites à large bande à travers le développement de nouvelles techniques d'accès et l'intégration des boucles locales sans fil (Wi-Fi et WiMAX).
- Consolider les standards européens de satellite à large bande DVB-S2 et DVB-RCS
- Définir des recommandations et mettre en œuvre une plateforme d'évaluation ainsi que des tests réels pour montrer comment les systèmes satellites à large bande pourraient s'intégrer dans les NGN, en se basant sur IPv6 et comment ils pourraient supporter les nouvelles applications multimédias (VoIP, visioconférence, jeux interactifs, etc.).

Pour atteindre ces différents objectifs, le projet propose alors une approche structurée en 4 étapes principales:

- Analyser et identifier les impacts des nouvelles applications IPv6 et des services liés aux NGN sur les réseaux satellites DVB-S2/RCS. Les principaux objectifs sont donc de définir une architecture de QoS, une architecture Multicast, une architecture de sécurité et une architecture de mobilité ainsi que la mise en œuvre de protocoles de transport adaptés aux besoins des systèmes satellites et l'intégration de PEPs (*Performance Enhancing Proxies*).
- Proposer l'intégration de réseaux hétérogènes couplant le système satellite avec le Wi-Fi et le WiMAX
- Valider les principaux concepts (mobilité, QoS, sécurité, etc.) au travers de simulations mais aussi d'émulations réalisées sur une plateforme de test qui doit aussi être développée au cours du projet.
- Présenter des expérimentations mettant en œuvre des applications IPv6 de bout-en-bout prouvant l'intégration de technologies compétitives.

Dans le cadre de notre mémoire, nous nous intéresserons plus particulièrement à l'architecture de QoS ainsi qu'aux solutions de mobilité que nous avons contribuées à mettre en œuvre dans le projet SATSIX.

III.3. La QoS dans les réseaux DVB-S2/RCS

Dans une architecture étoilée, la GW centralise l'ensemble des données et de la signalisation sur la voie aller (GW → STs) et occupe donc toute la largeur de la bande

passante offerte par le système. Les mécanismes de partage de ressources sont donc limités à l'ordonnancement des paquets en fonction de leur destination et de leur classe de service.

Mais si l'on considère la voie retour (en mode étoilé ou maillé), cela devient plus complexe puisque la bande passante doit être répartie entre les différents STs/GW connectés au système satellite. La méthode d'accès qui a alors été définie par la norme DVB-RCS pour résoudre ce problème est MF-TDMA (*Multi Frequency Time Division Multiple Access*). Elle offre une décomposition fréquentielle du lien de retour en plusieurs bandes de fréquence, elles-mêmes partagées en slots temporels. Au logon, chaque ST se voit alors allouer par le NCC une certaine capacité qui lui permet de transmettre des paquets MPEG2-TS ou des cellules ATM selon un plan d'allocation bien précis défini par le TBTP (*Terminal Burst Time Plan*), une trame de signalisation émise périodiquement par le NCC. Ces paquets sont transmis dans des timeslots réunis en trames, elles mêmes réunies en supertrames.

III.3.1. Le DAMA : un mécanisme d'allocation de bande passante à la demande

En plus de la capacité allouée au logon d'un ST, l'un des principaux avantages de la norme DVB-RCS est d'offrir un mécanisme d'allocation de bande passante à la demande, appelé DAMA (*Demand Assignment Multiple Access*) et qui fonctionne sur le principe client-serveur (client=ST, serveur=NCC). Ce mécanisme offre à chaque ST la possibilité de requérir dynamiquement des capacités de transmission auprès du NCC qui les lui alloue en fonction des besoins exprimés et des ressources disponibles. Cinq catégories de requête de capacité sont alors disponibles :

- CRA (*Continuous Rate Assignment*) qui permet de requérir une quantité de timeslots fixes et disponibles durant toute la durée de connexion du ST.
- RBDC (*Rate Based Dynamic Capacity*) qui permet de requérir une quantité de timeslots négociée pour une durée située entre 1 et 15 supertrames. Chaque nouvelle requête RBDC annule la précédente. Ce type de requête s'exprime typiquement en kbit/s.
- VBDC (*Volume Based Dynamic Capacity*) qui permet de définir un volume de données à transmettre qui peut être réparti sur plusieurs supertrames. Ces requêtes sont cumulatives. Ce type de requête s'exprime typiquement en nombre de cellules ATM ou de paquets MPEG.
- AVBDC (*Absolute Volume Based Dynamic Capacity*) : idem que VBDC sauf que les requêtes ne sont pas cumulatives, une nouvelle requête annulant la précédente.
- FCA (*Free Capacity Assignment*) est une capacité exprimée en volume qui permet de redistribuer entre les différents STs la bande passante inutilisée. Contrairement aux 4 requêtes précédentes qui sont directement négociées entre ST et NCC, FCA est réalisée automatiquement.

Ces requêtes de capacité (CRs, *Capacity Requests*) suivent l'ordre de priorité CRA > RBDC > A(VBDC) > FCA et sont transportées selon deux types de signalisation : une signalisation intra-bande qui permet d'encapsuler les requêtes dans des bursts normalement dédiés au trafic selon la méthode DULM (*Data Unit Labeling Method*) et/ou une signalisation

hors-bande basée sur la méthode « minislot » avec ou sans contention qui permet périodiquement d'allouer à chaque ST des bursts de plus courte durée que pour le trafic. La méthode la plus couramment utilisée est la signalisation hors bande.

Une table d'allocation des ressources (TBTP) est alors émise périodiquement toutes les 500 ms pour communiquer aux STs l'affectation des ressources.

III.3.2. La QoS de niveau IP

Pour permettre la mise en œuvre d'une QoS de niveau IP cohérente avec les exigences de niveau MAC décrite par la norme DVB-RCS, le groupe SatLabs de l'ESA (*European Space Agency*) travaillant pour la compatibilité et l'interopérabilité des systèmes DVB-S2/RCS, a défini des recommandations spécifiant une architecture de QoS pour ce type de systèmes.

Ces recommandations partent du principe qu'un ST/GW doit supporter la différenciation de QoS comme décrit dans l'architecture DiffServ. Dans ce contexte, le ST se comporte alors comme un routeur de bordure du domaine DiffServ, équivalent au système satellite, ce qui signifie que chaque nœud à l'intérieur du domaine opère selon les mêmes spécifications de PHB.

Dans le cadre de l'harmonisation de l'architecture de QoS des systèmes DVB-S2/RCS, le groupe SatLabs a alors défini qu'un ST devrait au moins supporter les PHBs suivantes [137] :

- Expedited Forwarding (EF).
- Au moins une classe Assured Forwarding (AF) subdivisée en au moins deux sous classes (par ex. AF31 et AF 32).
- Best Effort (BE).

Le groupe SatLabs recommande alors à titre d'exemple que chaque ST d'un système DVB-S2/RCS puisse offrir les performances de QoS détaillées par le Tableau 4.

Tableau 4 – Performances de QoS perçues par le groupe SatLabs comme un exemple raisonnable pour le déploiement d'un réseau DVB-RCS

PHB	Délai		Gigue		Priorité	Bande passante		Taux de perte de paquet	
	Nominal	Sur-charge	Nominal	Sur-charge		Nominal	Sur-charge	Nominal	Sur-charge
EF	Suffisamment bas (pas de surréservation)		Minimum possible			Offerte entièrement durant la session, si admission			
	300 ms		50-100 ms					<0.1%	
AF31 /AF32	Aussi bon que possible				Haute	Selon le contrat			
	850 ms	Plus long pour le trafic hors de profile						<0.1%	
BE	Non controlé		Non controlé		Basse				
	850 ms	Plus long pour le trafic hors de profile							

Les différentes classes de service de niveau IP doivent ensuite être mises en correspondance (ou mappées) avec les classes de trafic de niveau MAC. En effet, s'il est

important de définir des classes de trafic de niveau IP, l'architecture de QoS ne peut reposer uniquement sur elles. Leur mise en correspondance avec une architecture de QoS de niveau MAC est alors indispensable pour pouvoir utiliser pleinement les mécanismes offerts par le DAMA.

III.3.3. La QoS de niveau MAC

La norme DVB-RCS définit donc également des classes de trafic au niveau MAC sur lesquelles sont mappées les files DiffServ, conformément aux recommandations du groupe SatLabs [137]. Chaque classe de niveau MAC, aussi nommée classe de requête (RC, *Request Class*), est associée à une ou plusieurs requêtes de capacité et identifiée par un *Channel_ID*. Les classes de requête suivantes sont alors définies :

- La classe RT (*Real Time*) pour les applications à fortes contraintes temporelles (ex. VoIP, Visioconférence) et sur laquelle est mappée la file EF. Cette classe doit pouvoir utiliser l'allocation de type CRA et doit être associée à une connexion dédiée (canal logique) pour son trafic, identifiée par le VPI/VCI (*Virtual Path Identifier*/*Virtual Connection Identifier*) dans le cas d'ATM et le PID (Packet Identifier) dans le cas de MPEG2-TS.
- La classe CD (*Critical Data*) pour le trafic de données critiques, sur laquelle sont mappées les files AF.
- La classe BE (*Best Effort*) pour le trafic non critique (ex. e-mail), sur laquelle est mappée la file BE.

Toutes les classes doivent pouvoir utiliser la capacité FCA.

Les classes CD et BE peuvent partager un même canal logique ou en avoir chacune un différent mais dans tous les cas, le ST doit au moins permettre l'utilisation de deux canaux logiques dont un pour la classe RT.

L'ordre de priorité des classes est : RT > CD > BE ; ainsi, la capacité CRA allouée à un ST doit, par exemple, être utilisée prioritairement pour la classe RT mais si une partie de cette capacité reste inutilisée, elle peut être exploitée par une des autres classes.

Les différentes classes de requête peuvent ensuite utiliser les différentes catégories de requêtes de capacité (RBDC, A/VBDC).

En plus de ces classes définies pour le trafic de données, la norme DVB-RCS définit aussi la possibilité d'utiliser des classes spécifiques au trafic de gestion : les classes de niveau IP, INM et LNM, sont alors mappées sur une classe NM (*Network Management*) de niveau MAC.

III.3.4. La QoS dans le projet SATSIX

Dans le cadre du projet SATSIX [138], l'architecture DiffServ a aussi été choisie au niveau IP, d'une part pour rester en accord avec les recommandations du groupe SatLabs mais aussi pour ses capacités de passage à l'échelle. Trois catégories sont alors utilisées : une classe EF, une classe AF divisée en 3 sous classes (AF31, AF32 et AF33) et une classe BE. Pour ce qui est de la QoS niveau MAC, trois classes de trafic sont utilisées : DVB-RT pour le trafic temps réel, DVB-nRT pour le trafic prioritaire non temps réel et BE pour le reste,

chacune de ces classes étant associée à un canal logique spécifique. On peut voir sur la Figure 29 qu'une fois classifiés, les paquets IP passent par le module SAR (*Segmentation and Reassembly*) où ils sont segmentés en paquets MPEG2-TS ou en cellules ATM pour ensuite être répartis entre les deux classes de niveau MAC. La classe EF est mappée sur la file DVB-RT qui utilise prioritairement l'allocation statiquement allouée au logon du ST ainsi que des requêtes de capacité de type RBDC (mais on verra que certains mécanismes peuvent permettre l'utilisation de CRA), les classes AF sont mappées sur la file DVB-nRT correspondant aussi à des requêtes de type RBDC tandis que la classe BE est mappée sur la file MAC BE associée, a priori, à des requêtes de type RBDC ou VBDC. Cependant, ces associations entre file MAC et requête de capacité sont totalement configurables et peuvent être modifiées à souhait. De plus, les files nRT et BE peuvent bien sûr utiliser les timeslots alloués statiquement lorsque la file RT est vide. Finalement les paquets MPEG2-TS (ou cellules ATM) sont encapsulés dans des trames DVB-RCS envoyées sur le lien satellite.

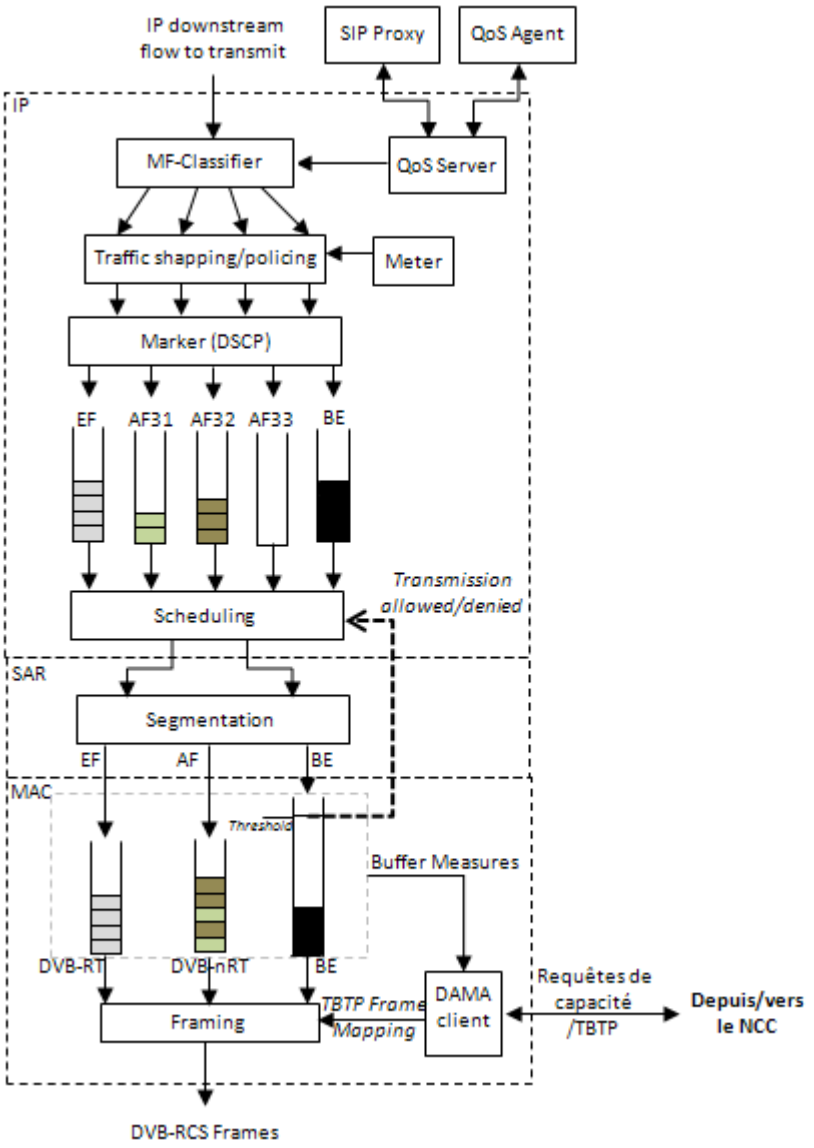


Figure 29 – Architecture de QoS SATSIX d'un ST

Le client DAMA, localisé au niveau de la couche MAC du ST, est en charge de contrôler l'accès au lien satellite en communiquant avec les autres ST ainsi qu'avec le NCC. Il reçoit aussi des mesures concernant la taille des files MAC et du taux de paquet les traversant.

Pour éviter que les files de niveau MAC débordent, des mécanismes d'interaction MAC/IP ont aussi été mis en place dans le cadre du projet. Par exemple, lorsque la file BE (au niveau MAC) dépasse un seuil fixé (correspondant au *threshold* indiqué sur la Figure 29) un algorithme de « backpressure » permet ainsi d'empêcher le module d'ordonnancement (*scheduling*) de continuer à alimenter la file. Le seuil peut être défini en fonction de l'algorithme d'allocation de bande passante à la demande.

Comparée à une architecture « traditionnelle » dans laquelle les paquets IP, après avoir été régulés, sont placés dans des buffers correspondant à différentes classes de trafic, segmentés en paquets MPEG2-TS, mappés vers des classes de niveau MAC, puis re-régulés et finalement réordonnancés, l'architecture proposée offre plusieurs avantages : 1) d'une part, cela évite la duplication des procédures d'ordonnancement, de mise en forme et d'application des politiques réalisées aux niveaux MAC et IP ; 2) ensuite, cela évite des débordements au niveau des files MAC qui impliquent une transmission partielle des paquets IP, ce qui cause un gaspillage de ressources ; 3) lorsqu'une congestion a lieu, les paquets s'accumulent dans les files IP et lorsque la congestion se termine, l'ordonnanceur est capable d'utiliser tous les critères de discrimination qui y ont été configurés, sélectionnant ainsi les paquets IP ayant les plus fortes exigences de délai.

Pour la configuration de la QoS, différents outils, initialement introduits dans le cadre du projet SATIP6 [139] ont été aussi la base de différentes améliorations et développements que j'ai réalisé, notamment pour le projet SATSIX. Ces outils, décrits dans [140], sont les suivants :

- Le QoS Server, un outil localisé au niveau de chaque ST/GW qui collecte des informations identifiant un flux, lui permettant ainsi de le rediriger dans une file DiffServ spécifique.
- Le QoS Agent, un outil localisé au niveau d'un terminal utilisateur et permettant à ce dernier d'indiquer dans un message de type RESV au QoS Server dans quelle classe de service il souhaite qu'un flux soit redirigé.
- L'idée de faire communiquer un proxy SIP avec le QoS Server avait été précédemment introduite et développée pour automatiser la configuration de la QoS mais uniquement pour la redirection d'un flux vers la file désirée dans le cas des applications SIP.

Cependant, pour une configuration plus fine de la QoS, même s'il est nécessaire de pouvoir rediriger des flux dans différentes classes de service en fonction des exigences de QoS qui leurs sont propres, il est aussi indispensable de mettre en place d'autres mécanismes permettant de quantifier précisemment les ressources qui vont être consommées par les flux pour lesquels on veut effectuer une réservation et configurer ainsi les différents élements du système satellite après avoir vérifié que ces ressources sont effectivement disponibles. Nous allons donc présenter nos propositions d'architecture pour la gestion de la QoS dans le paragraphe suivant.

III.3.5. Contributions pour l'amélioration de la gestion de la QoS

Nous présentons dans cette partie les différents développements et améliorations que nous avons apportés pour permettre une gestion plus fine de la QoS dans un système satellite DVB-S2/RCS. Une méthode de reconfiguration des files DiffServ ainsi que des mécanismes de contrôle d'admission sont ainsi proposés pour améliorer l'efficacité des outils précédemment décrits.

III.3.5.1. Les améliorations du QoS Server

Dans le cadre de la gestion de la QoS pour des applications à fortes contraintes temporelles au niveau des STs et de la GW, nous avons précisé qu'il était nécessaire d'avoir des informations permettant la reconfiguration dynamique des files DiffServ. L'information essentielle que le QoS Server doit connaître porte alors sur le débit du flux (par exemple débit moyen dans le cas de flux à débit constant) pour lequel on veut effectuer une réservation de QoS puisque, grâce à cette information, le QoS Server peut en effet vérifier la disponibilité des ressources et reconfigurer la taille des files de façon précise. Lorsque le QoS Server reçoit une requête de réservation concernant une communication VoIP utilisant un codec ayant un débit moyen de 50 kbit/s et nécessitant un service EF, il peut alors réaliser une procédure de CAC (*Connection Admission Control*) locale permettant de définir si l'utilisateur est autorisé à faire ce type de modifications et si les ressources nécessaires sont disponibles au niveau du système. Une fois la requête validée ou rejetée, le QoS Server prévient l'entité émettrice (QoS Agent ou proxy SIP) du résultat de l'opération et, si la réservation est acceptée, il reconfigure les files DiffServ au niveau du ST. Pour l'exemple précédent, la file EF sera augmentée de 50 kbit/s tandis que la file BE sera diminuée d'autant.

III.3.5.2. Le QoS Agent amélioré

Le QoS Agent est un outil permettant de détecter les flux applicatifs IPv4 et IPv6 sortants du terminal sur lequel il est installé et de les associer à différents niveaux de QoS, selon les souhaits (et le contrat d'abonnement) de l'utilisateur. Dans le cadre du projet SATSIX et des améliorations apportées au QoS Server, nous avons élargi les capacités du QoS Agent en ajoutant une option permettant à l'utilisateur de préciser un débit associé à l'application concernée. Grâce à l'interface graphique présentée Figure 30, l'utilisateur peut donc sélectionner le flux, la classe de service et le débit souhaité. Une fois la requête validée, celle-ci est envoyée au QoS Server localisé au niveau du ST. Le QoS Server peut alors réaliser les procédures décrites dans le paragraphe précdent. De son côté, si la requête a été validée, le QoS Agent mémorise la réservation effectuée : sur la Figure 30, on peut par exemple voir qu'une réservation de type EF avec un débit de 50K a été faite pour un flux iperf.

Dès qu'une application ayant bénéficié d'une réservation de QoS se termine, nous avons aussi ajouté la possibilité que le QoS Agent le détecte automatiquement et envoie directement l'information au QoS Server qui pourra ainsi modifier la configuration du ST en conséquence.

Cependant, il est important de noter que cet outil permet seulement de configurer la QoS des flux sortants. Dans le cas d'une communication bidirectionnelle ou d'une communication dans laquelle l'utilisateur reçoit des données, s'il souhaite améliorer la qualité de la réception,

il doit nécessairement prévenir son correspondant pour que celui-ci configure le QoS Server et donc le ST qui lui est associé, mais les applications ne sont pas prévues pour ce genre de situations ; cela ne sera donc pas toujours possible.

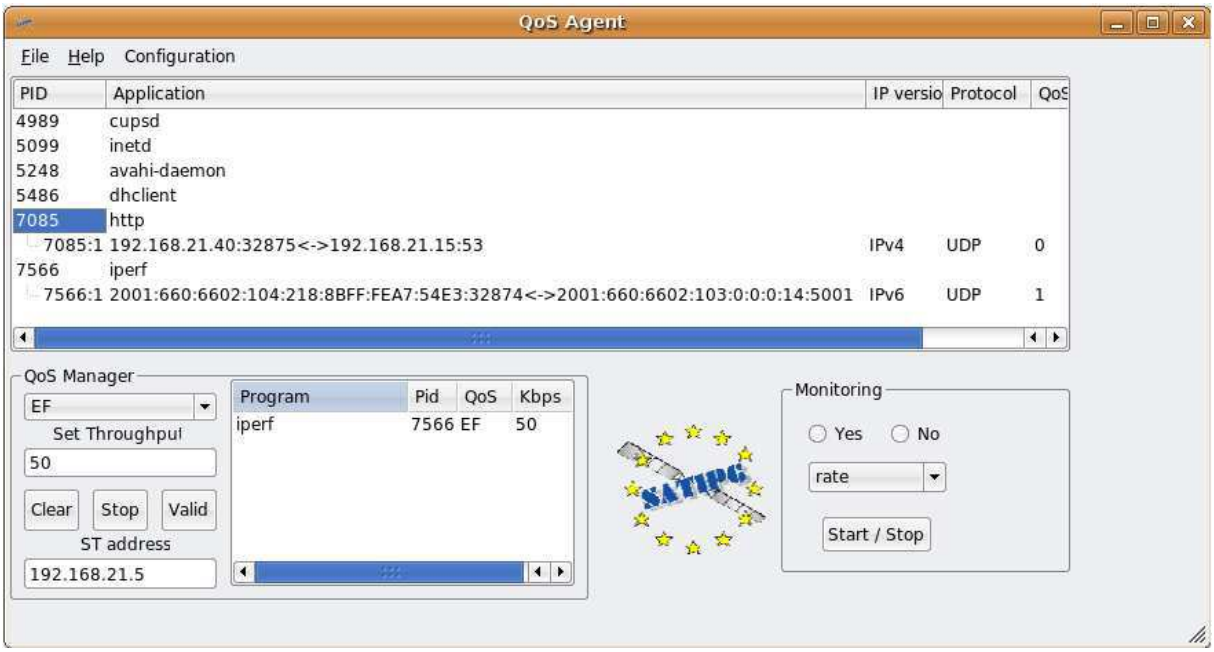


Figure 30 – Interface graphique du QoS Agent

De plus, il n'est pas toujours possible pour l'utilisateur d'évaluer correctement le débit qu'il veut associer à tel ou tel flux, il garde donc la possibilité d'envoyer une requête sans cette valeur, ce qui impliquera que la QoS Server ne fera que rediriger les paquets du flux correspondant dans la file souhaitée sans modifier sa taille. Si la taille de la file est trop faible, cela pourra donc avoir des effets néfastes sur le bon fonctionnement des autres flux traversant cette file, ce qui pose donc problème.

Pour les raisons indiquées précédemment, l'usage d'un QoS Agent est donc peu adapté et, dans tous les cas, le fait qu'un utilisateur puisse choisir lui-même le débit et la QoS associée à son trafic sortant semble peu réaliste. Cependant, nous verrons par la suite comment nous allons modifier cet outil pour permettre de configurer la QoS dynamiquement lors des déplacements d'un utilisateur mobile équipé de Mobile IPv6.

III.3.5.3. Le proxy SIP au service de la QoS

Une solution alternative à celle du QoS Agent pour la configuration de la QoS au niveau des ST est l'utilisation d'un proxy SIP amélioré en s'inspirant du concept expliqué dans la partie II.3.3. En effet, même si les applications multimédias sont elles aussi non conscientes de la QoS qu'elles requièrent, le fait qu'elles soient initiées, modifiées et terminées par des messages SIP contenant des descripteurs de session SDP permet une mise en œuvre efficace et automatique de la QoS.

Deux solutions sont alors possibles : une solution distribuée où un proxy SIP se trouve derrière chaque ST (chacun étant équipé d'un QoS Server) et une solution centralisée où un seul proxy SIP se trouve au niveau du NCC et s'occupe d'effectuer toutes les réservations auprès des QoS Server. Cependant, cette dernière solution présente le désavantage de

nécessiter des bonds satellites supplémentaires dans la plupart des cas comme par exemple celui où deux correspondants SIP sont chacun localisés derrière deux STs différents ; en effet, tous les messages SIP doivent alors réaliser un bond supplémentaire pour passer par le NCC, de même que les messages RESV et FREE échangés entre le proxy SIP et les QoS Servers. Les temps d'établissement de la session sont alors plus longs ce qui est pénalisant, particulièrement dans le cas où l'on considérera un utilisateur mobile (voir partie III.4.1). Pour nos évaluations, implémentations et expérimentations, nous considérons donc la solution distribuée où un proxy SIP se trouve derrière chaque ST.

Le proxy SIP amélioré, en plus de ses fonctions classiques de relais des messages SIP, doit alors être capable d'intercepter les descripteurs de session et de les interpréter pour en déduire les caractéristiques de chaque média impliqué dans la session. Il est alors nécessaire que celui-ci fonctionne en mode « stateful » afin de maintenir les informations d'état associées à la session. De plus, tous les messages SIP doivent passer par les proxies SIP en charge de la gestion de la QoS ; pour cela, le champ « Record Route » doit être configuré en conséquence. Ainsi, pour une communication entre deux clients SIP, chacun rattaché à un ST distinct, tous les messages SIP devront passer par au moins deux proxies, localisés derrière leur ST respectif, comme la Figure 31 le montre. Dans cet exemple, les proxies sont configurés en mode « assured » ce qui, rappelons le, signifie que la session ne pourra débuter que si la QoS a bien été mise en place, mais le mode « enabled » est aussi disponible au niveau des proxies. De même, nous avons choisi ici de représenter un établissement de session comme défini par la RFC 3312 [115] mais les proxies doivent aussi être capables de réserver la QoS dans le cas d'une session classique initiée uniquement par les messages INVITE/OK/ACK, auquel cas la QoS sera établie lors de la réception du message OK. Toutes ces modifications ont été réalisées dans le cadre de nos travaux de thèse.

Une fois que les caractéristiques des médias impliqués dans la session ont été déduites des descripteurs de session, chaque proxy SIP envoie un message à son QoS Server associé, de la même manière que pour le QoS Agent et ainsi, la QoS des STs peut être configurée automatiquement de la même manière que dans le cas du QoS Agent (modification de la taille des files en fonction du débit et marquage des paquets pour les rediriger vers la file DiffServ désirée).

Pour réaliser tout cela, différentes fonctionnalités doivent être ajoutées ou modifiées au niveau de chaque proxy :

- Modifications pour forcer tous les messages SIP à traverser tous les proxies.
- Modifications pour permettre la réservation aussi bien dans le cas d'une session classique qu'une session suivant la RFC 3312 [115].
- Un module d'extraction et d'analyse de descripteurs de session SDP, déjà réalisé auparavant.
- Une table d'association entre la session (son Call-Id) et les médias impliqués (audio, vidéo, etc.), chaque média étant associé à une série de caractéristiques (IP_src, Port_src, IP_dest, Port_dest, débit maximum, gigue maximum, délai maximum et taux de perte maximum), déduite soit d'une table de correspondance locale, soit par l'intermédiaire de services Web (détails dans le paragraphe III.3.5.5). Cette partie a été largement modifiée et améliorée dans le cadre de notre thèse.

- Un module qui réalise les procédures de réservation/libération de ressources en échangeant les messages RESV, FREE et ACK avec le QoS Server associé. Ce module a fait l'objet d'un redéveloppement presque total pour pouvoir s'adapter à tous les types de sessions possibles (INVITE envoyé par le client SIP 1 au client SIP 2, re-INVITE envoyé par le client SIP 1 ou 2 et BYE envoyé par le client SIP 1 ou 2) contrairement à l'implémentation précédente qui considérait uniquement le cas particulier où il fallait que ce soit le même client SIP qui envoie les messages INVITE et BYE et qui ne considérait pas le cas d'un re-INVITE.

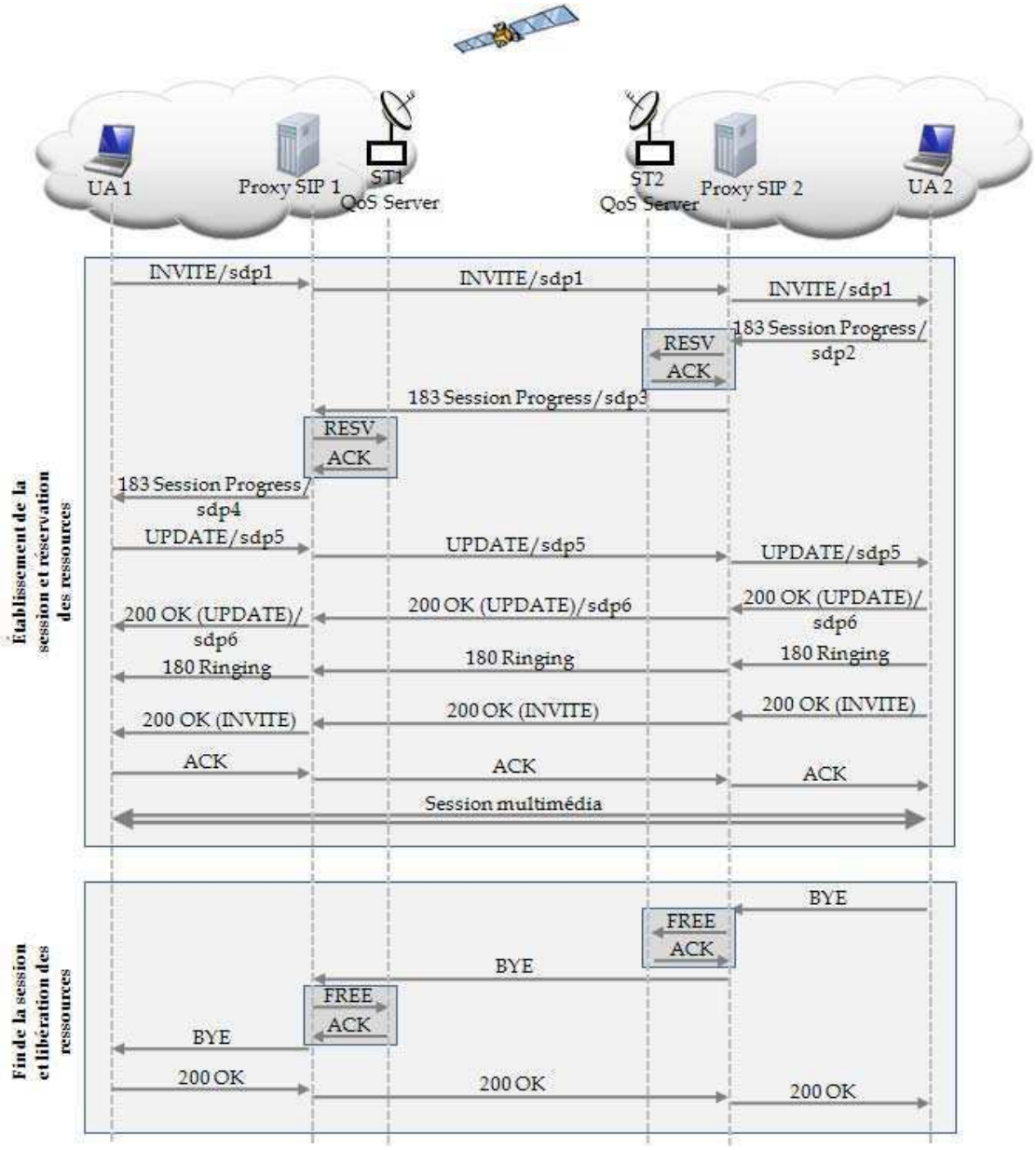


Figure 31 – Réservation/libération de QoS au cours d'une session SIP

Un avantage évident de cette solution basée sur le proxy SIP est qu'elle ne nécessite pas l'intervention des utilisateurs qui ne sont pas toujours aptes à juger des besoins des applications qu'ils utilisent. De plus, contrairement à la solution basée sur le QoS Agent, on peut constater sur la Figure 31 que la réservation de QoS se fait automatiquement dans les

deux sens de la communication, ce qui semble indispensable particulièrement pour les communications VoIP par exemple. Ensuite, les proxies peuvent être configurés par les FAI qui choisissent ainsi les politiques à appliquer, les classes de service à associer à tel ou tel type de média, etc. Enfin, si les paramètres SDP sont modifiés en cours de session (par l'utilisation d'un re-INVITE par exemple), les proxies SIP concernés pourront instantanément modifier la configuration de la QoS au niveau des ST (les QoS Server vont modifier la taille des files DiffServ), ce qui rend cette solution beaucoup plus dynamique.

Nous décrirons dans les deux paragraphes suivants les autres fonctionnalités associées à la configuration de la QoS du système satellite que nous avons ajoutée au proxy SIP. Enfin, les modifications aussi liées à la configuration de la QoS mais tenant compte de la possibilité que le client SIP soit mobile sont présentées dans le paragraphe III.5.3.

III.3.5.4. L'ARC pour permettre la modification dynamique des paramètres du DAMA

En plus du QoS Server qui permet d'effectuer un contrôle d'admission local et de configurer la QoS au niveau de chaque ST, une autre entité, appelée ARC (*Access Resource Controller*) a été définie au niveau du NCC pour permettre la modification dynamique des paramètres du serveur DAMA, qui, dans notre cas, correspond à l'augmentation de la quantité de CRA allouée à un ST donné.

Cette modification dynamique est utilisée dans le cas de communications interactives ayant de fortes contraintes temporelles et nécessitant une augmentation de la quantité de CRA allouée à un ST, les requêtes RBDC ne permettant pas d'assurer une QoS suffisante. Lors de l'initiation d'une session SIP de vidéoconférence par exemple, en plus des mécanismes liés aux QoS Server, les proxies SIP vont alors communiquer avec l'ARC, pour lui demander d'augmenter la quantité de CRA qui leur est allouée en fonction des caractéristiques des médias impliqués, contenues dans le message. L'ARC vérifie alors si les abonnés concernés sont autorisés à réaliser ce type d'opération et, en fonction de cela, décide de leur allouer ou non de la bande passante garantie si les ressources sont disponibles.

Pour réaliser cela, il est donc aussi nécessaire de rajouter au proxy SIP la possibilité d'échanger des messages RESV, FREE et ACK avec l'ARC pour lui demander l'augmentation de la quantité de CRA alloué au ST correspondant lors d'une initiation de session ou d'une ré-initiation de session (modification ou après un changement de réseau) et la diminution du CRA lors d'une fin de session ou d'un changement de réseau. Ces messages sont échangés avec l'ARC en même temps que les messages RESV ou FREE envoyés au QoS Server (voir Figure 31).

III.3.5.5. Améliorations des mécanismes de réservation de ressources à l'aide des Web Services

Pour permettre au proxy SIP de connaître les caractéristiques associées aux codecs utilisés lors d'une session et ainsi prévenir le QoS Server, une table de correspondance doit être mise en œuvre au niveau du proxy SIP. Cette table, sous format XML, énumère les codecs dont elle connaît les caractéristiques. De plus, puisque le QoS Server configure les

files au niveau IP, il a besoin de connaître le débit au niveau IP qui varie donc en fonction de la version d'IP puisque les entêtes IPv4 et IPv6 ont une taille différente. Il est donc nécessaire de préciser la version utilisée d'IP. Par exemple, pour le codec GSM-FR (GSM full rate), cela donne `< audiocodec ipversion="IPv6" number="3" name="GSM-FR" throughput="37.2" maxJitter="1" maxLoss="3" maxDelay="400" />` (pour cet exemple, le bitrate serait à 29.2 kbps en mode IPv4). Cependant, lorsque les caractéristiques d'un codec n'y sont pas présentes, la QoS ne peut être mise en place correctement au niveau du ST, ce qui met en péril la qualité de la future session. Pour résoudre ce problème, nous avons donc décidé d'utiliser les services Web (ou *Web Services*).

Les services Web permettent à des applications de dialoguer à distance via Internet, et ceci, indépendamment des plates-formes et des langages sur lesquelles elles reposent. Pour ce faire, ils s'appuient sur un ensemble de protocoles standards définis par le W3C (*World Wide Web Consortium*) [141] et s'articulent essentiellement autour des trois acronymes suivants :

- WSDL (*Web Services Description Language*) qui donne la description au format XML des Web Services en précisant les méthodes pouvant être invoquées, leur signature et le point d'accès (URL, port, etc…).
- UDDI (*Universal Description, Discovery and Integration*) qui normalise une solution d'annuaire distribué de Web Services, permettant à la fois la publication et la découverte d'un service.
- SOAP (*Simple Object Access Protocol*) qui est un protocole d'échange inter-applications, indépendant de toute plate-forme et basé sur le langage XML. Une invocation de service SOAP est en fait un flux ASCII encadré par des balises XML et transporté dans le protocole HTTP.

Les étapes successives liées à un Web Services sont alors : la définition du service par le fournisseur en utilisant WSDL, la publication du service par le fournisseur en utilisant UDDI, la découverte du service par le client en utilisant UDDI de nouveau et l'invocation du service par le client auprès du fournisseur en utilisant SOAP. Ces différentes étapes sont illustrées par la Figure 32.

Dans notre cas, nous proposons donc de mettre en œuvre un serveur, nommé MTR (*Media Type Repository*) offrant le service suivant : **fournir les caractéristiques QoS associées à un codec donné et à une version d'IP**, ces caractéristiques pouvant être obtenues auprès de sources telles que l'ITU-T par exemple.

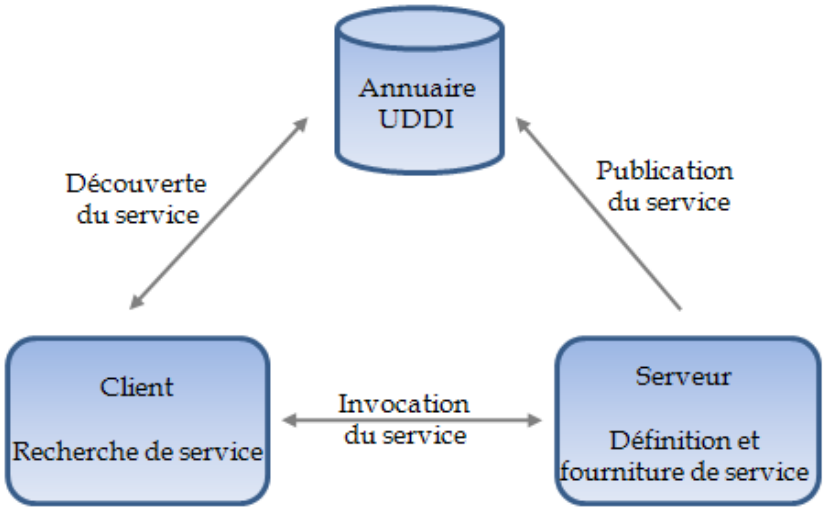


Figure 32 – Principe de fonctionnement des services Web

Ainsi, pour chaque type de média (audio, vidéo), le MTR stocke les caractéristiques associées aux codecs présents dans sa base de données, en fonction de la version d'IP utilisée. Lorsque le proxy SIP ne connaît pas localement les caractéristiques associées à un codec donné, il va invoquer le service correspondant en indiquant le codec recherché, le type de média concerné (audio ou vidéo) et la version d'IP. Le MTR lui répondra en indiquant les caractéristiques suivantes : peakBitRate (débit crête) en kpbs, maxJitter (Gigue maximale) en ms, maxLoss (taux de perte maximum) en % et maxDelay (délai maximum) en ms.

III.4. La mobilité dans les réseaux DVB-S2/RCS

En plus des différents types de mobilité que nous avons pu décrire dans la partie I.1, les systèmes satellites, de par leurs caractéristiques inhérentes, introduisent un certain nombre de nouveaux types de mobilité. En effet, si l'on considère un système satellite monospot qui couvre un tiers du globe, un utilisateur mobile équipé d'un ST embarqué pourra théoriquement se déplacer dans toute la zone de réception associée et conserver une connectivité continue. Mais si le système satellite est composé de plusieurs satellites et que chaque satellite est multi-faisceau, l'utilisateur équipé d'un ST embarqué pourra changer de spot de réception ou encore de satellite relai.

Cependant, ces différents types de mobilité qui font l'objet d'études spécifiques dans le cadre de la norme DVB-RCS+M [142] ne seront pas étudiés dans ce mémoire et, comme nous l'avons déjà précisé en I.1.6, nous nous focaliserons sur la mobilité de terminal qui, dans un contexte satellite, consiste en un utilisateur mobile non équipé d'un ST, pouvant se rattacher à un ST par l'intermédiaire de technologies telles que le Wi-Fi ou le WiMAX (elles-mêmes couplées au ST) et pouvant changer de ST de rattachement en cours de communication.

En fait, nous nous intéressons d'abord à l'étude de l'intégration de solutions de mobilité dans un système satellite. En effet, le délai de 250 ms introduit par le satellite risque d'allonger considérablement les temps d'interruption lors d'un changement de réseau. Mais si les systèmes satellites veulent réellement s'intégrer dans l'Internet du futur, il est important

pour eux de permettre la mise en place des mêmes services que ceux des réseaux terrestres et la mobilité en est l'un des principaux.

Le but, dans un premier temps, est donc d'évaluer les solutions de mobilité les plus prometteuses dans le cadre d'un système satellite DVB-S2/RCS. Pour notre étude, nous nous concentrerons sur les solutions MobileIPv6, FMIPv6, HMIPv6 et mobilité SIP qui sont, selon nous, les solutions ayant le plus de chance d'être réellement implémentées par la suite (voir introduction du chapitre III). En effet, MobileIPv6 et ses extensions présentent l'avantage de s'adapter à tous les types d'applications et SIP est le protocole de contrôle de session préféré pour la plupart des architectures NGN. De plus, dans le cadre de notre étude qui vise à coupler gestion de mobilité et gestion de QoS, on a pu voir qu'il présentait de nombreux avantages. Cependant, nous avons pu voir que la mobilité SIP n'était pas totalement spécifiée et que la procédure d'enregistrement par exemple pouvait être réalisée de différentes façons [76]. Nous allons donc commencer par spécifier le fonctionnement de la mobilité SIP intégrée à un système satellite.

III.4.1. Spécification de la mobilité SIP dans un système DVB-S2/RCS

La mobilité SIP peut être vu de différentes façons dans un système satellite. En effet, on peut considérer que le système satellite constitue un seul et même domaine ; dans ce cas là, un seul proxy serait nécessaire et serait naturellement positionné au niveau de l'élément central, c'est-à-dire au niveau de la GW. Dans une topologie étoilée, cela n'ajouterait aucun délai additionnel puisque toutes les communications doivent passer par la GW mais dans une topologie maillée, chaque message SIP devrait passer par la GW ce qui ajouterait au minimum un bond satellite supplémentaire par message SIP dès lors qu'une communication a lieu entre deux STs autres que la GW. Ceci pourrait à la limite être acceptable pour la mise en place de la communication, mais dans le cadre de la gestion de la mobilité, le temps d'interruption serait fortement allongé, ce qui n'est pas envisageable.

Nous avons donc opté pour une solution distribuée dans laquelle un proxy SIP est déployé au niveau de chaque ST. Ceci permet d'une part d'être compatible aussi bien avec une topologie maillée qu'une topologie étoilée, d'autre part de réduire les délais d'établissement de session et d'interruption dans le cas de déplacements en cours de communication et enfin, puisque que l'objectif final de notre thèse est d'intégrer une solution couplant QoS et mobilité, ce choix nous permet d'être en accord avec la solution de QoS dynamique basée sur les proxies SIP améliorés décrits dans la partie III.3.5.3.

Dans les paragraphes suivants, nous allons maintenant présenter plus en détail nos choix d'implémentation concernant le réenregistrement et la ré-initiation de session d'un client SIP mobile en considérant les différentes possibilités.

III.4.1.1. Problème de réenregistrement SIP d'un MN

Si l'on considère un MN, initialement enregistré au niveau de son proxy SIP mère, qui se déplace entre différents réseaux du système satellite, dans tous les cas, il doit être réalisé un réenregistrement au niveau de deux proxies SIP : le nouveau proxy SIP en charge du domaine dans lequel le MN s'est déplacé et le proxy SIP mère qui pourra ainsi rediriger les requêtes SIP à destination du MN vers ce nouveau proxy SIP. De plus, si le MN se déplace d'un 1er

réseau visité vers un 2ème (ces 2 réseaux étant distincts du réseau mère), le MN devra en plus se désenregistrer auprès du proxy en charge du 1er réseau visité. Or, les études [76] et [77] ne tiennent pas compte de ce désenregistrement et des mécanismes de timer ne sont pas suffisants dans le cadre de notre étude où nous voulons coupler gestion de la mobilité et de la QoS.

Nous avons alors envisagé plusieurs solutions en tenant compte aussi des contraintes spécifiques au satellite :

- L'envoi d'un message REGISTER/DEREGISTER pour chaque proxy SIP concerné (voir Figure 33) ; ces messages pouvant être envoyés simultanément depuis le MN pour gagner du temps, le temps d'enregistrement est de minimum 2 bonds satellite, c'est-à-dire 600 ms (on considère que 1 bond satellite = temps de propagation sur le lien satellite [250 ms] + temps de propagation sur lien Wi-Fi et/ou Ethernet + temps de traitement au niveau de l'émetteur, du récepteur ≈ 300 ms). Mais cette approche impose l'envoi de 3 messages quasi-similaires sur le lien radio auquel le MN vient de se connecter, ce qui est inefficace en termes d'utilisation de bande passante.

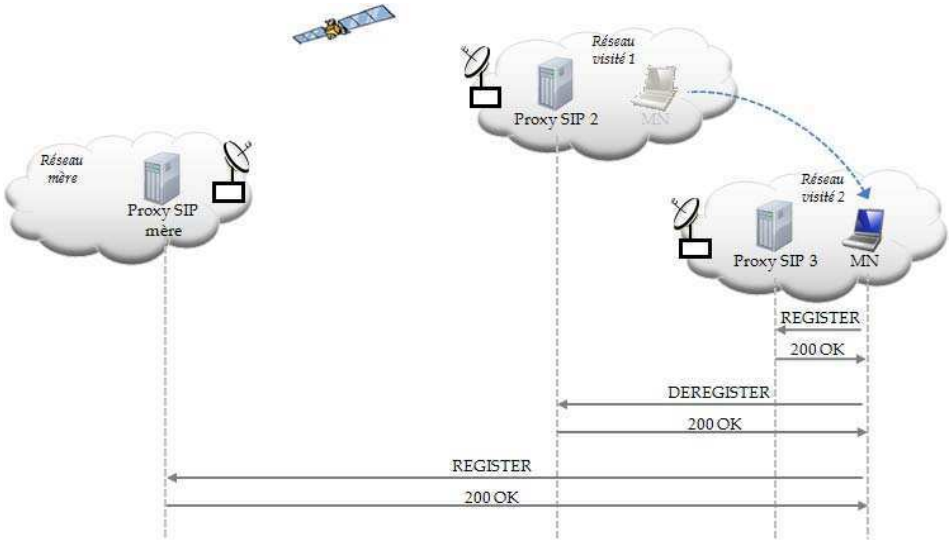


Figure 33 – Enregistrements initiés par le MN

- L'envoi d'un message REGISTER au proxy SIP mère qui se charge de transmettre le ou les autres messages aux autres proxies concernés (voir Figure 34). Cette solution résout le problème d'envoi de 3 messages au travers du lien radio mais pose problème en termes de temps d'enregistrement. En effet, 4 passages par le lien satellite sont nécessaires en considérant que les REGISTER/DEREGISTER envoyés depuis le « Home Proxy SIP » sont réalisés simultanément. Dans le cas d'une mobilité nomade, cette approche peut rester acceptable mais dans le cas d'une mobilité continue, le temps d'interruption serait augmenté d'au minimum 1,2s (300ms *4), ce qui n'est pas envisageable. De plus, cette solution implique le passage de 6 messages SIP au travers du lien satellite.

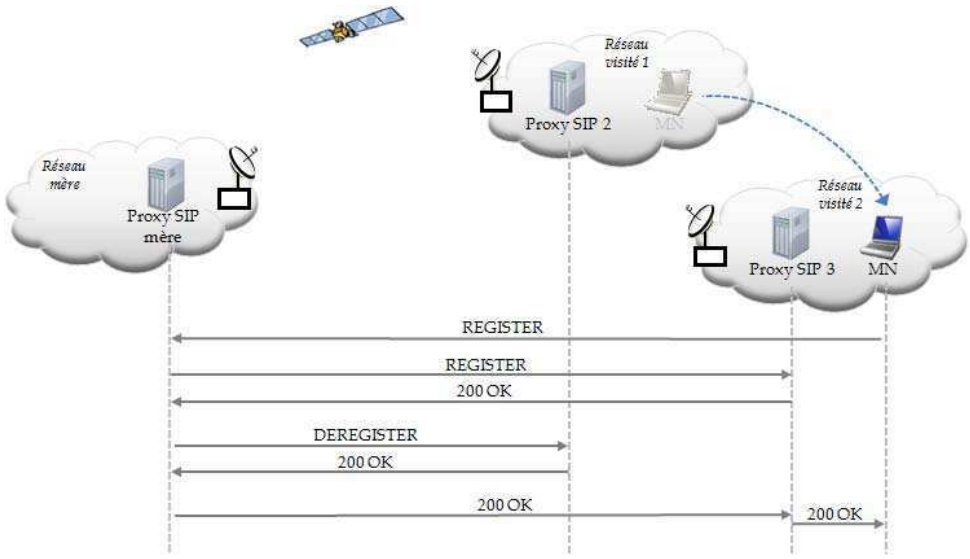


Figure 34 – Enregistrements initiés par le proxy SIP mère

- L'envoi d'un message REGISTER au proxy SIP en charge du domaine courant dans lequel se trouve le MN, ce proxy SIP transmettant alors le ou les messages REGISTER/DEREGISTER aux autres proxies (voir Figure 35). Cette solution permet une utilisation optimale aussi bien du lien radio (2 messages) que du lien satellite (4 messages). De plus, elle permet un temps d'enregistrement de 600 ms minimum.

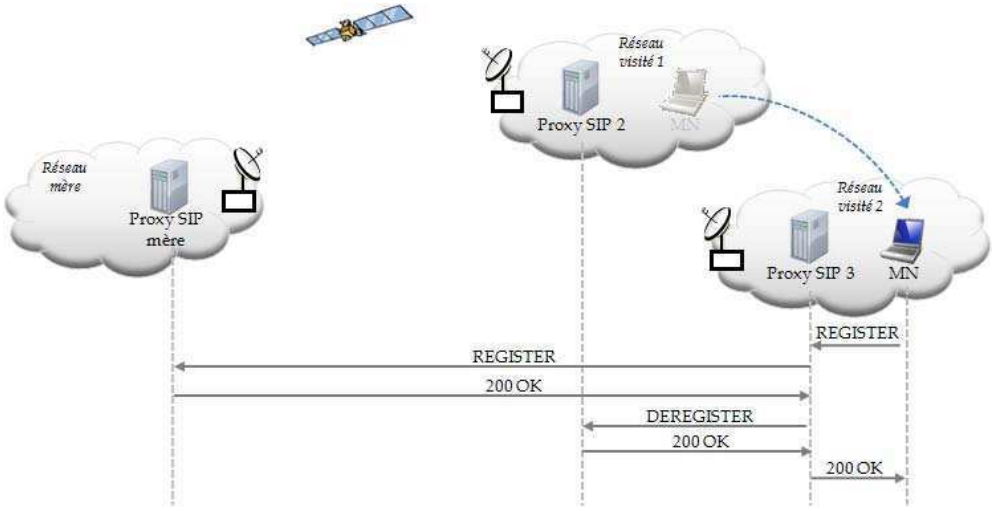


Figure 35 – Enregistrements initiés par le proxy SIP local

La 3ème solution correspondant aux enregistrements initiés par le proxy SIP local est donc la meilleure en termes de temps d'exécution. Cependant, un délai d'enregistrement de 600 ms reste pénalisant dans le cadre d'un déplacement en cours de communication. Dans ce cas là, nous utiliserons donc une solution basée sur cette 3ème proposition mais dans laquelle le OK final est renvoyé par le proxy SIP local au MN, juste après qu'il ait reçu le 1er REGISTER, comme indiqué Figure 36.

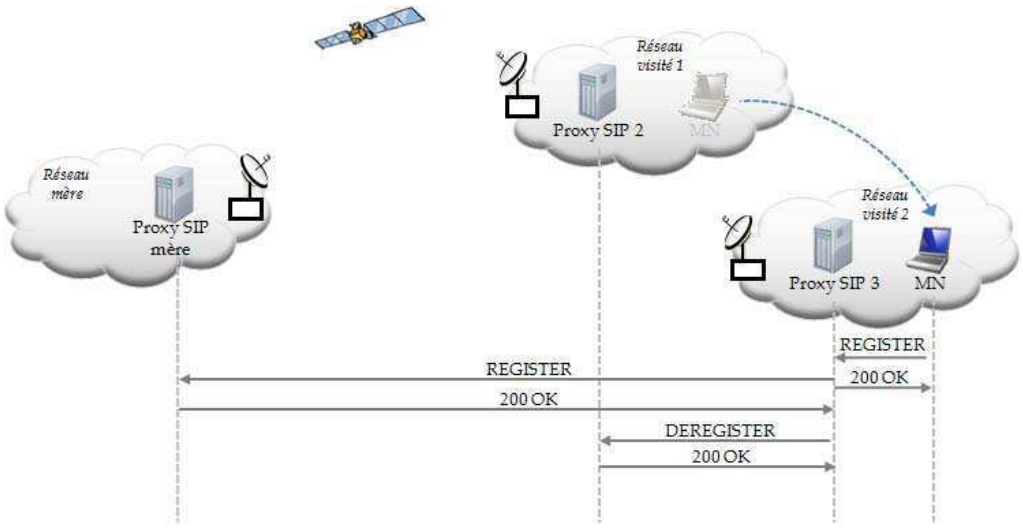


Figure 36 – Solution choisie pour le réenregistrement SIP d'un MN

On peut alors considérer les extensions suivantes pour les deux modes décrits dans le paragraphe II.3.3 en ce qui concerne le réenregistrement d'un client SIP mobile:

- Dans le mode « enabled », les proxies SIP sont configurés comme indiqué sur la Figure 36, pour permettre une procédure de réenregistrement plus rapide mais sans garantie que le réenregistrement auprès du réseau mère et le désenregistrement auprès de l'ancien proxy aient effectivement eu lieu. C'est sur ce mode que nos évaluations et expériences se baseront.
- Dans le mode « assured », les proxies SIP sont configurés comme indiqué sur la Figure 35. La procédure de réenregistrement est alors plus longue à réaliser, mais elle permet de garantir que le réenregistrement auprès du réseau mère et le désenregistrement auprès de l'ancien proxy ont effectivement eu lieu. Ce mode peut être utilisé, sans être pénalisant, dans les cas de mobilité nomade (voir paragraphe I.2.5.3).

III.4.1.2. Problème de ré-initiation de session SIP dans le cas d'une mobilité continue

Nous avons déjà vu dans la partie I.2.5.4 comment une session SIP en cours pouvait être ré-initié lorsqu'un déplacement avait lieu. Cependant, cette solution considère le cas simple où les deux correspondants relancent la communication en échangeant les messages directement l'un avec l'autre sans préciser le fonctionnement dans le cas où l'on souhaite que tous les messages passent au travers de proxies SIP. De plus, dans les parties II.3.3 et III.3.5.3, nous avons pu constater que des messages SIP supplémentaires étaient échangés dans le cadre de session SIP avec réservation de ressources (Session Progress, UPDATE, etc.) en suivant la RFC3312 [115]. L'objectif de cette partie est donc d'étudier l'impact de l'utilisation de ce standard dans le cas d'une mobilité continue.

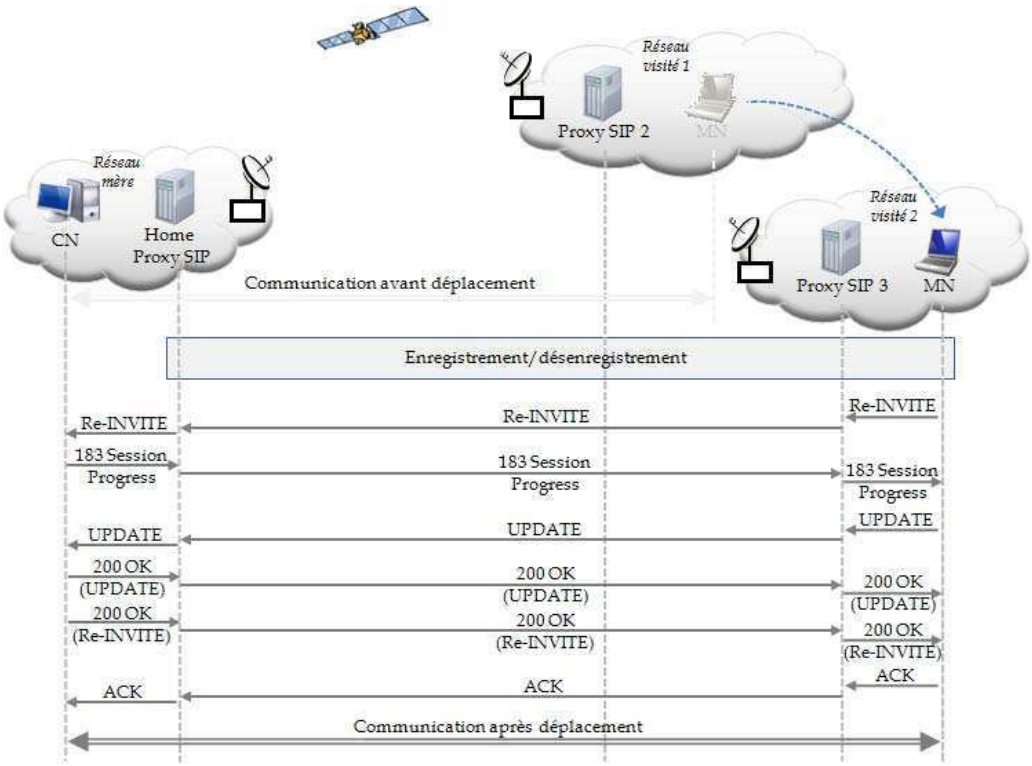


Figure 37 – Ré-initiation de session SIP selon la RFC 3312

La Figure 37 montre l'enchaînement de messages nécessaire à la ré-initiation d'une session SIP après le déplacement du MN en cours de session. On peut constater que 5 bonds satellite minimum sont nécessaires dans ce cas de figure, en considérant que le 200 OK (UPDATE) et le 200 OK (re-INVITE) sont envoyés quasi simultanément. Cela correspond donc à un temps d'interruption minimum de 1500 ms (5 * 300 ms) juste pour l'échange des messages SIP, sachant que pour obtenir le temps d'interruption global, il faut aussi tenir compte des temps de ré-association de niveau 2, de ré-acquisition d'une adresse IPv6, etc.

Il est donc difficilement envisageable de répondre à la RFC 3312 dans le cas d'une mobilité continue, sauf si le terminal mobile est équipé de deux interfaces et que la ré-initiation de session peut se faire depuis la 2ème interface tandis que la communication continue sur la 1ère. Dans le cas contraire, l'utilisation des messages SIP classiques re-INVITE/OK/ACK permet un gain de 600 ms ce qui convient beaucoup mieux au cas d'un nœud mobile.

Dans [1], les auteurs préconisent même une ré-initiation uniquement basée sur les messages re-INVITE et OK, sans utiliser le ACK qui sert essentiellement à fiabiliser la réception du OK. Ceci permet alors un gain de 300 ms supplémentaire au niveau du temps d'interruption correspondant à l'échange des messages SIP. En fait, si l'on considère que le CN envoie des messages juste après le 200 OK, le MN recevra même ses premiers paquets avec 600 ms d'avance par rapport au cas où le ACK est utilisé (puisque dans ce cas, le CN enverra ses premiers paquets à la réception du ACK, c'est-à-dire 2 bonds satellites plus tard).

Pour la suite de notre mémoire, nous nous focaliserons sur une solution compatible avec le standard, considérant d'une part que la ré-initiation de session SIP sera effectuée par l'échange des messages re-INVITE/200 OK/ACK comme l'indique la Figure 38 et d'autre

part, que si la QoS doit être mise en place, elle le sera au niveau du OK. Par contre, dans le cas d'une initiation de session SIP classique, la RFC 3312 pourra être suivie. On considérera tout de même, à titre de comparaison, le cas où le ACK continue à être échangé mais où la communication peut reprendre juste après le OK.

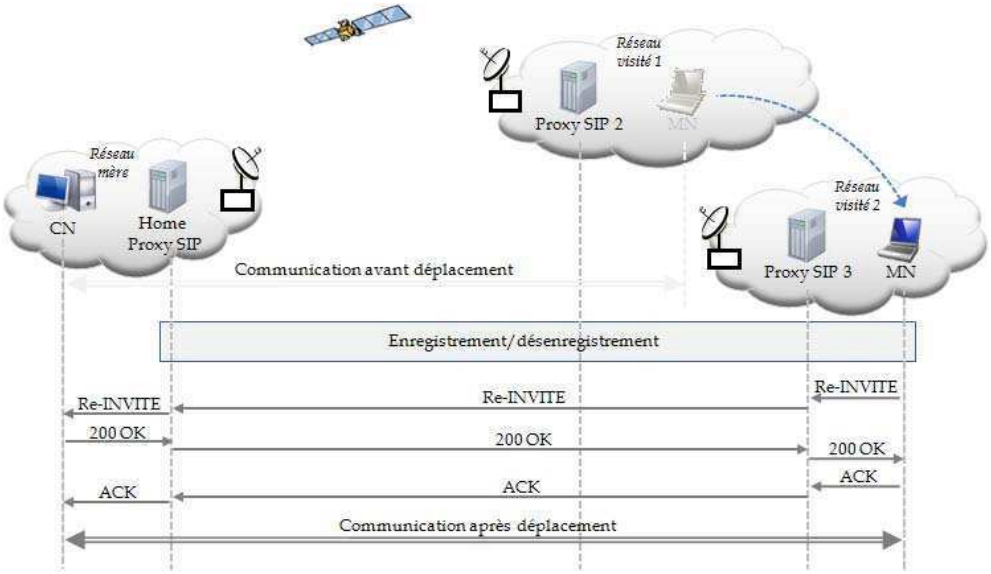


Figure 38 – Solution choisie pour la ré-initiation de session SIP

III.4.2. Evaluation théorique et recommandations

Dans cette partie, nous allons évaluer théoriquement les temps d'interruption et les délais de transfert de paquets pour chacune des solutions de mobilité considérées (Mobile IPv6, HMIPv6, FMIPv6, Mobilité SIP) et pour différents cas de déplacement présentés sur la Figure 39. On considère pour cela un système satellite composé des éléments suivants :

- Un satellite en mode régénératif.
- Trois STs/GW équipés d'une voie de retour DVB-RCS et communiquant directement les uns avec les autres. De plus, la GW/ST1 relie le système satellite au reste de l'Internet.
- Un NCC colocalisé avec la GW/ST1.
- Derrière chaque ST/GW se trouvent un proxy SIP pour la mobilité SIP et un MAP pour HMIPv6.
- Le réseau mère est connecté au système satellite par l'intermédiaire de la GW/ST1. Il comprend un HA pour MobileIPv6 et ses extensions.
- Le réseau correspondant est connecté au système satellite par l'intermédiaire du ST3. Il comprend le CN.
- Les réseaux visités 1 et 2 sont connectés au système satellite par l'intermédiaire du ST2.
- Tous les réseaux considérés (mère, correspondant et visités) sont des réseaux Wi-Fi dont le point d'accès (AP) joue le rôle de routeur d'accès (AR) et donc le rôle de PAR ou de NAR pour FMIPv6 selon la position dans le déplacement considéré.

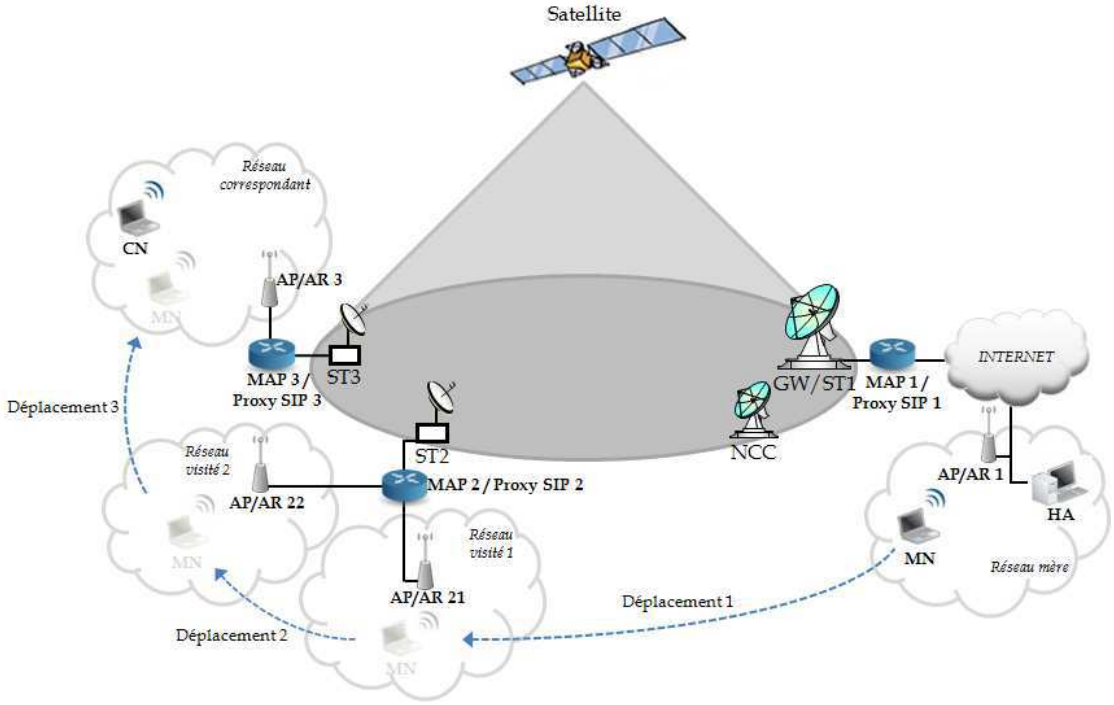


Figure 39 – Les principaux types de déplacement dans un système satellite

Le MN est initialement localisé dans son réseau mère et on considère qu'il est en communication de VoIP avec le CN, la session étant contrôlée par SIP. Quatre types de déplacements sont alors considérés (même si 3 seulement apparaissent sur la Figure 39 pour des raisons de clarté) :

- Le déplacement 1 dans lequel le MN passe du réseau mère au réseau visité 1. Ce type de déplacement est considéré comme de la mobilité globale (ou macro-mobilité).
- Le déplacement 2 dans lequel le MN passe du réseau visité 1 au réseau visité 2. Ce type de déplacement est considéré comme de la mobilité locale (ou micro-mobilité).
- Le déplacement 3 dans lequel le MN passe du réseau visité 2 au réseau correspondant, aussi considéré comme de la mobilité globale.
- Le déplacement 4 dans lequel le MN passe du réseau correspondant au réseau mère, aussi considéré comme de la mobilité globale.

Pour ce qui est du temps d'interruption T, on considère qu'il correspond au temps écoulé entre le dernier paquet reçu par le MN dans l'ancien réseau et le 1er paquet reçu par le MN dans le nouveau réseau. Pour le calculer, quatre temps s'avèrent utiles :

- T_2. En considérant le cas de l'utilisation du 802.11, ce temps correspond alors au temps nécessaire pour la synchronisation, l'authentification et l'association. Ce temps est très difficile à évaluer car il diffère beaucoup selon le type de carte Wi-Fi utilisé au niveau du terminal mobile ainsi que le type de point d'accès auquel il doit se rattacher [143]. De plus, la version utilisée du 802.11 a aussi une forte influence. Par exemple, dans [144], le délai de handover du 802.11b est évalué à 160 ms tandis que la norme 802.11r promet des délais inférieurs à 50ms. Pour toutes ses raisons, nous prendrons arbitrairement comme valeur de référence **T_2 = 100 ms**. Ce temps étant le même pour

toutes les solutions comparées, le choix de cette valeur n'influencera pas les évaluations obtenues.

- T_3 qui représente le temps nécessaire pour obtenir une nouvelle adresse IPv6, c'est-à-dire le temps pour recevoir un Router Advertisement (RA) et le temps nécessaire pour que les mécanismes du DAD (Duplicate Address Detection) [23] soient effectués. Or, pour la configuration de radvd dans le cadre de l'utilisation de Mobile IPv6, la période moyenne entre deux RA est de 50 ms. Donc en moyenne, un MN arrivant dans un nouveau réseau recevra un RA en 25 ms et on considérera ce temps pour toutes les solutions, y compris celle n'utilisant pas Mobile IPv6 ou ses extensions. De plus, selon [145], T (DAD)=1500 ms. Dans ce cas là, on a donc **T_3 = 1525 ms**.
- Cependant, 1500 ms représentent un temps très long juste pour obtenir une adresse IPv6 valide et pouvoir l'utiliser. Des études ont donc été réalisées pour réduire ce temps et ont donné naissance à un mécanisme appelé Optimistic DAD [146] qui permet l'utilisation d'une adresse IPv6 avant que la procédure de DAD n'ait été complétée. On considère dans ce cas que T_3 correspond uniquement au temps de réception d'un RA, c'est-à-dire **T_3 = 25 ms**.
- T_{m1} représente le temps écoulé entre l'envoi du 1er message du protocole de mobilité considéré (dans le réseau d'arrivée) et le 1er paquet reçu par le MN dans le nouveau réseau.
- T_{m2} représente le temps écoulé entre l'envoi du 1er message du protocole de mobilité considéré (dans le réseau d'arrivée) et le 1er paquet reçu en mode « route optimisée » par le MN dans le nouveau réseau. On a alors $T_{m2} = T_{m1}$ + temps écoulé entre l'envoi du 1er message nécessaire pour l'optimisation de route et la réception par le MN du 1er paquet en mode « route optimisée ».

Dans le Tableau 5, on indiquera le temps correspondant à T_3 uniquement lorsqu'il est différent de 1525 ms. Pour ce qui est de T_2, on considère qu'il est toujours égal à 100 ms donc il ne figurera pas dans le tableau.

Par rapport aux 300 ms nécessaires pour échanger un paquet devant traverser le système satellite, on considérera que l'échange de paquets entre deux entités situées derrière le même ST est proche de zéro et donc négligeable. Ces valeurs sont une fois de plus des ordres de grandeur pour nous donner une idée des temps d'évaluation et pouvoir faire une étude comparative théorique.

Enfin, nous ajoutons les notations suivantes :

- T* = temps écoulé entre le dernier paquet reçu par le MN dans l'ancien réseau et le premier paquet reçu en route optimisée par le MN dans le nouveau réseau. On obtient donc **$T^* = T_2 + T_3 + T_{m2}$.**
- T' = temps écoulé entre le dernier paquet reçu par le MN dans l'ancien réseau et le 1er paquet reçu par le tunnel entre le HA et le MN (et non pas par le tunnel FMIPv6 entre le PAR et le NAR). Cette notation est spécifique à FMIPv6.
- D = délai de transfert des paquets reçus par le MN entre T et T* (ou pour FMIPv6 entre T et T' lorsque T' existe).
- D' = délai de transfert des paquets reçus par le MN entre T' et T*.
- D* = délai de transfert des paquets reçus par le MN après T*.

- Lorsque nous indiquons qu'un délai ou un temps est ≈ 0, cela signifie qu'il est très faible, de l'ordre de quelques millisecondes maximum.

Tableau 5 –Temps d'interruption et délais de transfert des paquets

	Déplacement 1		Déplacement 2		Déplacement 3		Déplacement 4	
	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau
MobileIPv6 sans RO avec DAD	Tm1 = 600 ms T = 2225 ms	D = 600 ms	Tm1 = 600 ms T = 2225 ms	D = 600 ms	Tm1 = 600 ms T = 2225 ms	D = 600 ms	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms	D = 300 ms
MobileIPv6 avec RO et RRT avec DAD	Tm1 = 600 ms Tm2 = 2400 ms T = 2225 ms T* = 4025 ms	D = 600 ms D* = 300 ms	Tm1 = 600 ms Tm2 = 2400 ms T = 4025 ms T* = 4025 ms	D = 300 ms D* = 300 ms	Tm1 = 600 ms Tm2 = 1800 ms T = 3425 ms T* = 3425 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300 ms
MobileIPv6 avec RO et RRT avec Optimistic DAD	T3 = 25 ms Tm1 = 600 ms Tm2 = 2400 ms T = 725 ms T* = 2525 ms	D = 600 ms D* = 300 ms	T3 = 25 ms Tm1 = 600 ms Tm2 = 2400 ms T = 2525 ms T* = 2525 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm1 = 600 ms Tm2 = 1800 ms T = 1925 ms T* = 1925 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* =300 ms
HMIPv6 sans RO avec DAD	Tm1 = 600 ms T = 2225 ms	D = 600 ms	Tm1 ≈ 0 ms T = 1625 ms	D = 600 ms	Tm1 = 600 ms T = 2225 ms	D = 600 ms	Tm1 ≈ 0 ms T = 1625 ms	D = 300 ms
HMIPv6 avec RO et RRT avec DAD	Tm1 = 600 ms Tm2 = 2400 ms T = 2225 ms T* = 4025 ms	D = 600 ms D* = 300ms	Tm1 ≈ 0 ms Tm2 = 0 ms T = 1625 ms T* = 1625 ms	D = 300 ms D* = 300 ms	Tm1 = 600 ms Tm2 = 1800 ms T = 3425 ms T* = 3425 ms	D ≈ 0 ms D* ≈ 0 ms	Tm1 ≈ 0 ms Tm2 = 1200 ms T = 2825 ms T* = 2825 ms	D = 300 ms D* = 300 ms
HMIPv6 avec RO et RRT avec Optimistic DAD	T3 = 25 ms Tm1 = 600 ms Tm2 = 2400 ms T = 725 ms T* = 2525 ms	D = 600 ms D* = 300ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 0 ms T = 125 ms T* = 125 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm1 = 600 ms Tm2 = 1800 ms T = 1925 ms T* = 1925 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300ms
FMIPv6 Mode prédictif sans RO avec DAD	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms T' = 725 ms	D = 600 ms D' = 600 ms	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms T' = 725 ms	D = 600 ms D' = 600 ms	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms T' = 725 ms	D = 600 ms D' = 600 ms	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms T' = 125 ms	D = 300 ms D' = 300 ms
FMIPv6 Mode prédictif avec RO et RRT avec DAD	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 2400 ms T = 125 ms T' = 725 ms T* = 2525 ms	D = 600 ms D' = 600 ms D* = 300 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 2400 ms T = 125 ms T* = 2525 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1800 ms T = 125 ms T* = 1925 ms	D = 600 ms D* ≈ 0 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1200 ms T = 125 ms T* = 1325 ms	D = 300 ms D* = 300 ms
FMIPv6 Mode réactif sans RO avec DAD	Tm1 = 1800 ms T = 2825 ms T' = 3425 ms	D = 600 ms D' = 600 ms	Tm1 = 600 ms T = 1625 ms T' = 2225 ms	D = 600 ms D' = 600 ms	Tm1 = 1800 ms T = 2825 ms T' = 3425 ms	D = 600 ms D' = 600 ms	T3 = 25 ms Tm1 = 1200 ms T = 1325 ms T' = 1325 ms	D = 300 ms D' = 300 ms
FMIPv6 Mode réactif avec RO et RRT avec DAD	Tm1 = 1800 ms Tm2 = 3600 ms T = 2825 ms T' = 3425 ms T* = 5225 ms	D = 600 ms D' = 600 ms D* = 300 ms	Tm1 = 600 ms Tm2 = 2400 ms T = 1625 ms T* = 4025 ms	D = 600 ms D* = 300 ms	Tm1 = 1800 ms Tm2 = 3000 ms T = 2825 ms T* = 4625 ms	D = 600 ms D* ≈ 0 ms	T3 = 25 ms Tm1 = 1200 ms Tm2 = 2400 ms T = 1325 ms T* = 2525 ms	D = 300 ms D* = 300 ms
FMIPv6 Mode réactif avec RO et RRT avec Optimistic DAD	T3 = 25 ms Tm1 = 1800 ms Tm2 = 3600 ms T = 1325 ms T' = 1925 ms T* = 3725 ms	D = 600 ms D' = 600 ms D* = 300 ms	T3 = 25 ms Tm1 = 600 ms Tm2 = 2400 ms T = 125 ms T* = 2525 ms	D = 600 ms D* = 300 ms	T3 = 25 ms Tm1 = 1800 ms Tm2 = 3000 ms T = 1325 ms T* = 3125 ms	D = 600 ms D* ≈ 0 ms	T3 = 25 ms Tm1 = 1200 ms Tm2 = 2400 ms T = 1325 ms T* = 2525 ms	D = 300 ms D* = 300 ms
Mobilité SIP après le ACK avec DAD	Tm2 = 1200 ms T = 2825 ms T* = 2825 ms	D = 300 ms D* = 300 ms	Tm2 = 1200 ms T = 2825 ms T* = 2825 ms	D = 300 ms D* = 300 ms	Tm2 ≈ 0 ms T = 1625 ms T* = 1625 ms	D ≈ 0 ms D* ≈ 0 ms	Tm2 = 1200 ms T = 2825 ms T* = 2825 ms	D = 300 ms D* = 300 ms
Mobilité SIP après le ACK avec Optimistic DAD	T3 = 25 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm2 ≈ 0 ms T = 125 ms T* = 125 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300 ms
Mobilité SIP avant le ACK avec DAD	Tm2 = 600 ms T = 2225 ms T* = 2225 ms	D = 300 ms D* = 300 ms	Tm2 = 600 ms T = 2225 ms T* = 2225 ms	D = 300 ms D* = 300 ms	Tm2 ≈ 0 ms T = 1625 ms T* = 1625 ms	D ≈ 0 ms D* ≈ 0 ms	Tm2 = 600 ms T = 2225 ms T* = 2225 ms	D = 300 ms D* = 300 ms
Mobilité SIP avant le ACK avec Optimistic DAD	T3 = 25 ms Tm2 = 600 ms T = 725 ms T* = 725 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm2 = 600 ms T = 725 ms T* = 725 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm2 ≈ 0 ms T = 125 ms T* = 125 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm2 = 600 ms T = 725 ms T* = 725 ms	D = 300 ms D* = 300 ms

Nous allons maintenant analyser en détail les résultats présentés dans le Tableau 5 en parcourant chaque protocole séparément.

III.4.2.1. Mobile IPv6

En ce qui concerne Mobile IPv6, nous pouvons constater que les résultats sont répartis sur 3 lignes. La 1ère ligne correspond à l'utilisation de Mobile IPv6 sans optimisation de route (RO) et sans Return Routability Test (RRT). Dans ce cas là, lorsqu'il se déplace en dehors de son réseau mère, le MN doit recevoir un RA, procéder au DAD puis échanger un BU/BACK avec son HA. La mobilité du MN est alors complètement transparente pour le CN qui continue à envoyer ses paquets à destination de l'adresse mère du MN. Les paquets sont alors interceptés par le HA qui les tunnèle jusqu'à la position courante du MN. Mais le délai de transfert des paquets est toujours de 600 ms, même lorsque le MN et le CN se trouvent derrière le même ST (après le déplacement 3) ce qui semble très inapproprié. Ce délai n'est donc pas compatible avec les conversations audio et vidéo (voir Tableau 1). Il est donc fortement recommandé d'utiliser MobileIPv6 avec RO dans le cas de déplacement impliquant un système satellite.

Tableau 6 – Rappel des évaluations concernant Mobile IPv6

	Déplacement 1		Déplacement 2		Déplacement 3		Déplacement 4	
	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau
MobileIPv6 sans RO avec DAD	Tm1 = 600 ms T = 2225 ms	D = 600 ms	Tm1 = 600 ms T = 2225 ms	D = 600 ms	Tm1 = 600 ms T = 2225 ms	D = 600 ms	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms	D = 300 ms
MobileIPv6 avec RO et RRT avec DAD	Tm1 = 600 ms Tm2 = 2400 ms T = 2225 ms T* = 4025 ms	D = 600 ms D* = 300 ms	Tm1 = 600 ms Tm2 = 2400 ms T = 4025 ms T* = 4025 ms	D = 300 ms D* = 300 ms	Tm1 = 600 ms Tm2 = 1800 ms T = 3425 ms T* = 3425 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300 ms
MobileIPv6 avec RO et RRT avec Optimistic DAD	T3 = 25 ms Tm1 = 600 ms Tm2 = 2400 ms T = 725 ms T* = 2525 ms	D = 600 ms D* = 300 ms	T3 = 25 ms Tm1 = 600 ms Tm2 = 2400 ms T = 2525 ms T* = 2525 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm1 = 600 ms Tm2 = 1800 ms T = 1925 ms T* = 1925 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300 ms

Dans le cas de l'utilisation du RO (2ème ligne), on considère que la procédure de RRT est obligatoire pour les raisons de sécurité déjà évoquées dans la partie I.2.2.1.c. Les temps d'interruption diffèrent alors selon le type de déplacement considéré:

- Après le déplacement 1, le MN doit recevoir un RA (25 ms), procéder au DAD (1500 ms), échanger un 1er BU/BACK avec son HA (600 ms), réaliser le RRT (HoTi/HoT et CoTi/CoT en parallèle : 1200 ms) puis échanger un 2ème BU/BACK avec le CN (600 ms). Cette procédure est donc très longue à réaliser. Cependant, le MN recommence à recevoir des données après avoir mis à jour son association au niveau du HA qui peut lui transmettre par le tunnel : ces données sont donc reçues avec 600 ms de délai.
- Après le déplacement 2, le MN doit réaliser les mêmes procédures que précédemment mais la différence est que le CN et le MN communiquaient directement alors que le MN se trouvait dans un réseau visité 1. Cela implique que la mise à jour de

l'association au niveau du HA ne permet pas la mise en place d'un tunnel puisque le CN continue d'envoyer ses paquets à destination de la CoA obtenue dans le réseau visité 1 et n'accepte pas de paquets en provenance d'une adresse différente de cette CoA sans qu'il y ait eu une nouvelle mise à jour BU/BACK. Pour que la communication reprenne, le MN doit donc auparavant réaliser le RRT et la mise à jour au niveau du CN. Par contre, la communication reprendra directement en mode « route optimisée », ce qui implique que **T=T***.

- Après le déplacement 3, la gestion est similaire à celle du déplacement 2 sauf que le MN se trouve dans le même réseau que le CN. Donc les échanges entre ces deux entités ne nécessitent pas un bond satellite et le temps d'interruption est réduit. Pour la même raison, le délai de transfert des paquets est très faible (≈ 0).
- En ce qui concerne le déplacement 4, la procédure de DAD n'est pas nécessaire puisque le MN utilise son HoA. Pour la procédure de RRT, seul l'échange HoTi/HoT est ici nécessaire puisque la CoA correspond à la HoA du MN. Le temps d'interruption est alors bien supérieur au cas de MobileIPv6 sans RO puisque la communication ne peut pas reprendre avant que le MN ait mis à jour son association au niveau du CN (HoTi/HoT puis BU/BACK).

Cependant, comme cela est expliqué dans la RFC 3775 [31], dans le cas où le MN revient dans un réseau qu'il a déjà récemment visité (et reçoit la même CoA) et pour lequel il a déjà reçu une *care-of keygen token* récente, la procédure de RRT peut éventuellement être validée sans aucun échange de message. Nous étudierons cette possibilité plus en détail dans la partie IV.4.

Comme on l'avait déjà vu dans la partie I.2.1.2.d, on peut limiter la durée du RRT en le démarrant au même moment que le BU envoyé au HA, cela permet de gagner 600 ms.

Le troisième cas d'utilisation de Mobile IPv6 permet juste de réaliser à quel point l'utilisation d'une méthode telle que l'Optimistic DAD permet de gagner du temps au niveau de l'interruption.

III.4.2.2. HMIPv6

Nous reprenons maintenant pour HMIPv6, les 3 mêmes cas que pour Mobile IPv6. On considère que, lorsque le MN arrive dans un nouveau domaine MAP et qu'il doit (1) s'autoconfigurer une RCoA ainsi que (2) une LCoA, le temps nécessaire pour réaliser ces procédures est le même que pour obtenir une CoA dans le cas de Mobile IPv6 (on considère que les deux procédures (1) et (2) sont réalisées en parallèle).

Lorsque l'optimisation de route n'est pas utilisée, on retrouve le même problème que pour Mobile IPv6 qui implique un délai de transfert des paquets entre le CN et le MN de l'ordre de 600 ms dès que le MN se trouve hors de son réseau mère. On voit quand même que pour le déplacement 2, le temps T est fortement diminué puisque le MN a juste à prévenir son MAP localement (d'où un T_{m1} très faible). A l'inverse, dans le cas d'un retour dans le réseau mère (déplacement 4), contrairement à Mobile IPv6, le MN doit quand même obtenir une RCoA et une LCoA. Donc, le temps d'interruption est allongé de 1500 ms (procédure de DAD). Il est aussi important de noter que dans le cas de HMIPv6, la communication va se faire en utilisant des encapsulations multiples puisque les paquets envoyés depuis le CN

jusqu'à la HoA du MN sont d'abord encapsulés par le HA à destination du MAP qui les encapsule à nouveau à destination de la LCoA du MN. On a donc un overhead important qui implique une utilisation non optimale du lien satellite ainsi que du lien Wi-Fi. Nous analyserons plus en détail la question de l'overhead concernant Mobile IPv6 et ses extensions dans le chapitre IV.

Tableau 7 – Rappel des évaluations concernant HMIPv6

	Déplacement 1		Déplacement 2		Déplacement 3		Déplacement 4	
	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau
HMIPv6 sans RO avec DAD	Tm1 = 600 ms T = 2225 ms	D = 600 ms	Tm1 ≈ 0 ms T = 1625 ms	D = 600 ms	Tm1 = 600 ms T = 2225 ms	D = 600 ms	Tm1 ≈ 0 ms T = 1625 ms	D = 300 ms
HMIPv6 avec RO et RRT avec DAD	Tm1 = 600 ms Tm2 = 2400 ms T = 2225 ms T* = 4025 ms	D = 600 ms D* = 300ms	Tm1 ≈ 0 ms Tm2 = 0 ms T = 1625 ms T* = 1625 ms	D = 300 ms D* = 300 ms	Tm1 = 600 ms Tm2 = 1800 ms T = 3425 ms T* = 3425 ms	D ≈ 0 ms D*≈ 0 ms	Tm1 ≈ 0 ms Tm2 = 1200 ms T = 2825 ms T* = 2825 ms	D = 300 ms D* = 300 ms
HMIPv6 avec RO et RRT avec Optimistic DAD	T3 = 25 ms Tm1 = 600 ms Tm2 = 2400 ms T = 725 ms T* = 2525 ms	D = 600 ms D* = 300ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 0 ms T = 125 ms T* = 125 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm1 = 600 ms Tm2 = 1800 ms T = 1925 ms T* = 1925 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300ms

Lorsque l'optimisation de route est activée (et donc la RRT), on constate des résultats similaires à ceux obtenus pour Mobile IPv6 dans le cas des déplacements 1 et 3. Par contre, le cas du déplacement 2 qui correspond à un cas de micro-mobilité permet de voir l'avantage réel du protocole HMIPv6 : aucun message ne devant traverser le satellite, le temps d'interruption est réduit à T_2+T_3 et le délai de transfert des paquets reste inférieur aux recommandations préconisées pour les applications de VoIP ou de vidéoconférence (< 400 ms). Pour le déplacement 4, la procédure de DAD doit aussi être réalisée contrairement au cas de Mobile IPv6.

Dans le cas de l'utilisation de la méthode d'Optimistic DAD, on peut de nouveau remarquer un gain de temps considérable, particulièrement dans le cas du déplacement 2 où le temps d'interruption est seulement de 125 ms.

III.4.2.3. FMIPv6 en mode prédictif

En mode prédictif, on considère que le déplacement du MN se fait suffisamment lentement pour que celui-ci se trouve dans la zone de chevauchement des réseaux de départ et d'arrivée suffisamment longtemps pour pouvoir réaliser les procédures nécessaires avant que le changement de réseau ait réellement lieu. Dans les cas où la procédure de DAD est utilisée, ce temps de chevauchement doit donc être au moins supérieur au temps nécessaire pour échanger les messages RtSolPr/PrRtAdv, FBU, HI/HACK et FBACK ainsi que le temps nécessaire aux mécanismes de DAD, c'est-à-dire supérieur au temps équivalent à 2 bonds satellite (puisque les messages HI et HACK traversent le satellite) + 1500 ms, donc supérieur à 2100 ms. Si la procédure de DAD n'est pas réalisée, ce temps est réduit à 600 ms. Ces différentes hypothèses correspondent au meilleur des cas pour FMIPv6.

Comme pour les protocoles précédents, on commence d'abord par considérer le cas de l'utilisation de FMIPv6 sans optimisation de route et avec procédure de DAD. Dans ce cas là, si les conditions précédentes sont réunies, le temps d'interruption est toujours réduit à T_2+T_3 = 125 ms (on considère en effet que pour que le MN sache qu'il est bien arrivé dans le nouveau réseau, il doit aussi recevoir un RA, donc T_3 = 25 ms). Pour les 3 premiers déplacements, le tunnel entre le PAR et le NAR doit être conservé au moins pendant T'-T (=600 ms), le temps que le MN prévienne son HA de sa nouvelle adresse. Par contre, après chacun de ses 3 déplacements, le délai de transfert des paquets entre le MN et le CN est d'environ 600 ms puisque le CN continue à envoyer vers la HoA du MN, donc non compatible avec les recommandations sur le délai pour la VoIP et la vidéoconférence. Dans le cas du déplacement 4, le tunnel PAR-NAR doit être maintenu uniquement le temps que le MN échange localement les BU/BACK avec son HA et le délai D revient à 300 ms. Le temps T_{m1} est faible (≈ 0 ms) puisqu'il correspond au temps écoulé entre l'envoi du UNA et la réception du 1er paquet par le tunnel PAR-NAR qui sont échangés localement.

Tableau 8 – Rappel des évaluations concernant FMIPv6 en mode prédictif

	Déplacement 1		Déplacement 2		Déplacement 3		Déplacement 4	
	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau
FMIPv6 Mode prédictif sans RO avec DAD	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms T' = 725 ms	D = 600 ms D' = 600 ms	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms T' = 725 ms	D = 600 ms D' = 600 ms	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms T' = 725 ms	D = 600 ms D' = 600 ms	T3 = 25 ms Tm1 ≈ 0 ms T = 125 ms T' = 125 ms	D = 300 ms D' = 300 ms
FMIPv6 Mode prédictif avec RO et RRT avec DAD	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 2400 ms T = 125 ms T' = 725 ms T* = 2525 ms	D = 600 ms D' = 600 ms D* = 300 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 2400 ms T = 125 ms T* = 2525 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1800 ms T = 125 ms T* = 1925 ms	D = 600 ms D* ≈ 0 ms	T3 = 25 ms Tm1 ≈ 0 ms Tm2 = 1200 ms T = 125 ms T* = 1325 ms	D = 300 ms D* = 300 ms

Lorsque les procédures de RO et RRT sont réalisées, elles permettent, pour les déplacements 1 et 3, de réduire le délai à D* = 300 ms. Mais le cas d'utilisation le plus probant reste le déplacement 2, pour lequel le temps d'interruption T n'est que de 125 ms et le délai D est directement de 300 ms. Le cas du déplacement 4 reste aussi très performant. Les valeurs T' et D' n'ont pas lieu d'être pour les 3 derniers déplacements puisque, après un déplacement du MN, le CN continue d'envoyer ses paquets à destination de l'ancienne CoA du MN et ceux-ci sont interceptés par le PAR et tunnelés vers le NAR ; le fait d'échanger un BU/BACK avec le HA n'y change donc rien. Le tunnel PAR-NAR doit alors être maintenu pendant au minimum T*- T.

Nous n'avons pas intégré au tableau une ligne décrivant FMIPv6 en mode prédictif, avec RO, RRT et Optimistic DAD puisque les résultats résumés dans le tableau seraient exactement les mêmes qu'avec le DAD. En effet, la procédure étant réalisée pendant la communication, elle ne prolonge pas le temps d'interruption. Par contre, dans le cas de l'utilisation de la procédure d'Optimistic DAD, cela va réduire le temps de chevauchement nécessaire pour obtenir une adresse pour le nouveau réseau et mettre en place le tunnel PAR-NAR.

III.4.2.4. FMIPv6 en mode réactif

On considère ici le pire des cas d'utilisation de FMIPv6 dans lequel le MN n'a pas le temps d'envoyer de FBU à son PAR. La procédure du DAD est alors réalisée à la réception du HI comme indiqué dans [15]. De plus, le tunnel PAR-NAR ne peut être mis en place qu'après l'échange des messages FBU/HI/HACK/FBACK qui, mis à part pour le cas du déplacement 2, doivent tous passer par le satellite.

Le cas du déplacement 1 sans RO et avec DAD nous permet tout d'abord de voir l'énorme différence qu'il existe entre les temps d'interruption dans le cas de FMIPv6 en mode réactif (2845 ms) et en mode prédictif (125 ms). Contrairement au mode prédictif, dans le cas réactif, le tunnel PAR-NAR doit en effet être mis en place alors que la communication est interrompue. De plus, pour les déplacements 1, 2 et 3, on retrouve le même problème que dans les cas précédents sans RO : le délai est de 600 ms puisque le CN continue à envoyer à destination de la HoA du MN. Pour ces déplacements, le tunnel PAR-NAR doit aussi être maintenu le temps que le MN prévienne son HA, c'est-à-dire au minimum 600 ms à partir de T.

Tableau 9 – Rappel des évaluations concernant FMIPv6 en mode réactif

	Déplacement 1		Déplacement 2		Déplacement 3		Déplacement 4	
	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau
FMIPv6 Mode réactif sans RO avec DAD	Tm1 = 1800 ms T = 2825 ms T' = 3425 ms	D = 600 ms D' = 600 ms	Tm1 = 600 ms T = 1625 ms T' = 2225 ms	D = 600 ms D' = 600 ms	Tm1 = 1800 ms T = 2825 ms T' = 3425 ms	D = 600 ms D' = 600 ms	T3 = 25 ms Tm1 = 1200 ms T = 1325 ms T' = 1325 ms	D = 300 ms D' = 300 ms
FMIPv6 Mode réactif avec RO et RRT avec DAD	Tm1 = 1800 ms Tm2 = 3600 ms T = 2825 ms T' = 3425 ms T* = 5225 ms	D = 600 ms D' = 600 ms D* = 300 ms	Tm1 = 600 ms Tm2 = 2400 ms T = 1625 ms T* = 4025 ms	D = 600 ms D* = 300 ms	Tm1 = 1800 ms Tm2 = 3000 ms T = 2825 ms T* = 4625 ms	D = 600 ms D* ≈ 0 ms	T3 = 25 ms Tm1 = 1200 ms Tm2 = 2400 ms T = 1325 ms T* = 2525 ms	D = 300 ms D* = 300 ms
FMIPv6 Mode réactif avec RO et RRT avec Optimistic DAD	T3 = 25 ms Tm1 = 1800 ms Tm2 = 3600 ms T = 1325 ms T' = 1925 ms T* = 3725 ms	D = 600 ms D' = 600 ms D* = 300 ms	T3 = 25 ms Tm1 = 600 ms Tm2 = 2400 ms T = 125 ms T* = 2525 ms	D = 600 ms D* = 300 ms	T3 = 25 ms Tm1 = 1800 ms Tm2 = 3000 ms T = 1325 ms T* = 3125 ms	D = 600 ms D* ≈ 0 ms	T3 = 25 ms Tm1 = 1200 ms Tm2 = 2400 ms T = 1325 ms T* = 2525 ms	D = 300 ms D* = 300 ms

Lorsque les procédures de RO et RRT sont activées, le délai de transfert des paquets entre le CN et le MN peut réduire à 300 ms à partir de T* mais on voit que ce temps nécessaire pour que les paquets arrivent en route optimisée est très long (jusqu'à 5.225 s pour le déplacement 1). En fait, cela permet aussi de comprendre que, même si FMIPv6 peut s'avérer être la solution la plus efficace lorsque le mode prédictif est possible, quand ce n'est pas le cas, ce protocole peut aussi devenir la pire des solutions.

Lorsque la procédure d'Optimistic DAD est possible, elle permet, comme pour les protocoles précédents, un gain de 1.5 s pour les 3 premiers déplacements.

III.4.2.5. Mobilité SIP

Dans cette partie, nous étudions le cas de la mobilité SIP en considérant que la communication peut se relancer soit avant l'envoi du ACK (après le OK), soit après et pour chacun des cas, la procédure du DAD ou de l'Optimistic DAD peut être utilisée.

La principale caractéristique qui saute aux yeux est qu'il n'y a pas de passage par une phase non optimisée où le délai est de 600 ms et on a donc, pour tous les cas considérés, T=T* et D=D*. On peut aussi constater que l'utilisation de la procédure d'Optimistic DAD fait aussi gagner un temps précieux et permet de réduire de plus de la moitié le temps d'interruption (dans le cas du déplacement 3, ça le réduit même à environ $T_2 + T_3 = 125$ ms).

On peut aussi constater que, mis à part pour le déplacement 3, le temps d'interruption T (ou T*) est diminué de 600 ms si l'on compare la mobilité SIP avec ACK avec la mobilité SIP sans ACK.

Tableau 10 – Rappel des évaluations concernant la mobilité SIP

	Déplacement 1		Déplacement 2		Déplacement 3		Déplacement 4	
	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau	Temps d'interruption	Délai aller après chgt de réseau
Mobilité SIP après le ACK avec DAD	Tm2 = 1200 ms T = 2825 ms T* = 2825 ms	D = 300 ms D* = 300 ms	Tm2 = 1200 ms T = 2825 ms T* = 2825 ms	D = 300 ms D* = 300 ms	Tm2 ≈ 0 ms T = 1625 ms T* = 1625 ms	D ≈ 0 ms D* ≈ 0 ms	Tm2 = 1200 ms T = 2825 ms T* = 2825 ms	D = 300 ms D* = 300 ms
Mobilité SIP après le ACK avec Optimistic DAD	T3 = 25 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm2 ≈ 0 ms T = 125 ms T* = 125 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm2 = 1200 ms T = 1325 ms T* = 1325 ms	D = 300 ms D* = 300 ms
Mobilité SIP avant le ACK avec DAD	Tm2 = 600 ms T = 2225 ms T* = 2225 ms	D = 300 ms D* = 300 ms	Tm2 = 600 ms T = 2225 ms T* = 2225 ms	D = 300 ms D* = 300 ms	Tm2 ≈ 0 ms T = 1625 ms T* = 1625 ms	D ≈ 0 ms D* ≈ 0 ms	Tm2 = 600 ms T = 2225 ms T* = 2225 ms	D = 300 ms D* = 300 ms
Mobilité SIP avant le ACK avec Optimistic DAD	T3 = 25 ms Tm2 = 600 ms T = 725 ms T* = 725 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm2 = 600 ms T = 725 ms T* = 725 ms	D = 300 ms D* = 300 ms	T3 = 25 ms Tm2 ≈ 0 ms T = 125 ms T* = 125 ms	D ≈ 0 ms D* ≈ 0 ms	T3 = 25 ms Tm2 = 600 ms T = 725 ms T* = 725 ms	D = 300 ms D* = 300 ms

III.4.2.6. Comparaison des solutions

Dans cette partie, nous allons maintenant comparer entre elles les différentes solutions proposées précédemment pour chaque type de déplacement. Deux types de comparaison sont alors nécessaires : une première qui considère seulement les temps d'interruption T, même si les premiers paquets arrivent avec un délai de 600 ms et une deuxième comparaison qui ne considère que le temps à partir duquel le délai est de 300 ms et reste donc compatible avec les recommandations ITU-T sur les applications VoIP ou vidéoconférence.

Pour le 1er type de comparaison, le cas qui se révèle globalement le plus performant est FMIPv6 en mode prédictif puisqu'il permet dans tous les cas de réduire le temps T à 125 ms. Un autre avantage de cette solution est qu'elle reste très efficace même en utilisant la procédure du DAD puisque celle-ci est réalisée dans le nouveau réseau tandis que la communication a encore lieu depuis l'ancien réseau. Cependant, la mise en place d'un tunnel temporaire entre le PAR et le NAR est toujours nécessaire et implique, pour les déplacements 1 et 3, un double bond par le satellite. Pour ce qui est du déplacement 2, HMIPv6 avec RO,

RRT et Optimistic DAD est quand même la solution la plus performante puisqu'elle permet aussi d'obtenir T=125 ms avec un délai D=300 ms (comme FMIPv6 prédictif) mais en plus, les seuls messages qui sont nécessaires sont le LBU/BACK échangés avec le MAP, contrairement aux nombreux messages nécessaires à la mise en place du tunnel FMIPv6. La solution Mobile IPv6 sans RO permet aussi une gestion efficace du déplacement 4, avec T=125 ms et D=300 ms, sans nécessiter d'autres messages que le BU/BA échangé avec le HA. Mais ces deux dernières solutions ne se révèlent efficaces que dans le cas d'un déplacement bien précis au détriment des autres types de déplacements.

Pour le 2ème type de comparaison, nous allons comparer chaque type de déplacement l'un après l'autre en ne tenant compte que des solutions avec RO et RRT:

- Pour le déplacement 1, si l'on compare les solutions en utilisant la procédure d'Optimistic DAD, on remarque que les solutions basées sur la mobilité SIP sont les plus performantes et permettent de diminuer de plus d'une seconde les temps d'interruption, en comparaison des solutions basées sur Mobile IPv6.
- Pour le déplacement 2, les solutions HMIPv6 (avec Optimistic DAD) et FMIPv6 prédictif restent les plus efficaces, pour les mêmes raisons que pour le 1er type de comparaison.
- Pour le déplacement 3, les solutions de mobilité SIP sont aussi les plus performantes que l'on considère les mécanismes de DAD ou d'Optimistic DAD. Ils permettent en effet de rétablir la communication directement entre le MN et le CN en 125 ms environ lorsque la procédure d'Optimistic DAD est utilisée.
- Pour le déplacement 4, la solution FMIPv6 prédictif est la plus efficace.

III.4.2.7. Recommandations concernant la mobilité dans un système satellite

La première constatation concernant Mobile IPv6 et ses extensions est que, dans le cadre d'un système satellite, il est indispensable que les procédures de RO et RRT puissent être réalisées pour pouvoir être compatible avec les recommandations ITU-T concernant les applications de VoIP ou de vidéoconférence. D'autre part, la phase de tunnel bidirectionnel implique des encapsulations IPv6/IPv6 qui ajoutent un overhead important au système (on le quantifiera dans le chapitre IV) et le fait de devoir traverser 2 fois le système satellite au lieu d'une a aussi pour conséquence une consommation plus importante des ressources globales du système satellite, ce qui confirme le caractère indispensable des procédures de RO et RRT.

La deuxième constatation concernant Mobile IPv6 et HMIPv6 avec RO et RRT est que la phase de tunnel bidirectionnel n'est utilisable que pour un seul type de déplacement : lorsque le MN quitte son réseau mère. Cela signifie qu'aucune application (y compris celle n'ayant pas de contrainte sur le délai) ne peut profiter de la phase de tunnel bidirectionnel pour tous les autres types de déplacements, ce qui réduit fortement l'intérêt de cette solution pour un utilisateur se déplaçant entre différents réseaux visités.

On peut aussi constater que le temps nécessaire pour réaliser les mécanismes de DAD est bien trop long. Des mécanismes similaires à l'Optimistic DAD doivent donc être mis en place.

FMIPv6 se révèle très efficace pour les déplacements sans changement de ST (type déplacement 2) et les déplacements avec changement de ST depuis le réseau du CN vers un autre réseau (type déplacement 4) dès lors que le mode prédictif peut être réalisé. Cependant, cette solution est beaucoup moins efficace dans les autres types de déplacement principalement parce que les procédures de RO et RRT, nécessaires pour réduire le délai à un bond satellite, sont longues à réaliser. De plus, si le mode prédictif ne peut être réalisé, les temps d'interruption deviennent carrément prohibitifs et sont mêmes supérieurs dans tous les cas à ceux obtenus avec Mobile IPv6. L'utilisation de FMIPv6 doit donc être faite en s'assurant au maximum que les conditions soient réunies pour que le mode prédictif puisse avoir lieu.

HMIPv6 se révèle efficace dans les déplacements de micro-mobilité (type déplacement 2) pour lequel il a d'ailleurs été conçu. Mais vu ses résultats pour les autres types de déplacements, HMIPv6 doit absolument être combiné avec un ou plusieurs autres protocoles de mobilité.

Ensuite, on constate que Mobile IPv6 (avec Optimistic DAD) se révèle aussi efficace que ses extensions dans le cas des déplacements 1 et 3. Pour ces déplacements, la longue procédure de RO+RRT est très pénalisante et constitue la majeure partie du temps d'interruption, il serait donc intéressant que des mécanismes tels que [40] soient implémentés pour le réduire, ces améliorations étant aussi valables pour HMIPv6 et FMIPv6.

Enfin, dans le cas d'une gestion de la mobilité pour des applications interactives basées sur SIP dans un système satellite DVB-S2/RCS, la mobilité SIP se révèle être une solution particulièrement efficace. En effet, elle présente l'énorme avantage de pouvoir relancer la communication directement entre les deux entités concernées, contrairement aux solutions précédentes, ce qui lui permet aussi de gérer efficacement tous les types de déplacements sans dépendre d'une topologie précise et même d'être la solution la plus efficace pour les déplacements 1 et 3. Et le fait de ne pas utiliser de ACK (ou du moins de relancer la communication juste après le OK) permet encore d'améliorer ses performances. Enfin, elle n'ajoute aucun overhead lorsque le MN se trouve dans un réseau visité.

La mobilité SIP se révèle donc être une bonne alternative à Mobile IPv6 ou ses extensions dans le cas de la gestion de la mobilité dans un système satellite pour les applications ayant de fortes contraintes temporelles, d'une part pour ses performances purement liées à la mobilité mais aussi de part son lien direct avec la configuration de la QoS. Par contre, Mobile IPv6 (ainsi que ses extensions) reste une solution efficace pour la gestion de la mobilité des applications ayant moins de contraintes, notamment sur le délai de transmission.

III.5. Propositions d'architectures de mobilité et de QoS pour les systèmes DVB-S2/RCS

A partir des outils de QoS proposés et des évaluations précédentes sur les différentes solutions de mobilité que nous avons considérées, nous allons maintenant tenter de définir des architectures permettant une gestion efficace à la fois de la mobilité et de la QoS dans un système satellite DVB-S2/RCS. Pour cela, il est donc nécessaire de déterminer les besoins

spécifiques de chaque solution en tenant compte des conclusions faites dans la partie II.5.3. Nous nous appliquerons donc dans un premier temps à définir une solution permettant la gestion de la QoS dans le cadre de Mobile IPv6. Puis, nous verrons comment cette solution peut être utilisée dans le cas de FMIPv6. Enfin, nous proposerons une solution basée sur SIP. Nous n'étudierons pas le cas HMIPv6 puisque, dans les cas de mobilité globale, la gestion de la QoS est identique à celle de Mobile IPv6 et dans le cas d'une mobilité locale (déplacement 2 sur la Figure 39), la mobilité est complètement transparente pour les STs et la QoS ne nécessite donc aucun changement au niveau du système satellite.

III.5.1. Mobile IPv6 couplé au QoS Agent mobile

Pour le cas de Mobile IPv6, nous voulons définir une solution ne nécessitant pas l'ajout d'une nouvelle option IPv6. De plus, étant donné que l'architecture de QoS d'un système satellite DVB-S2/RCS est basée sur DiffServ, il va de soi que l'utilisation de RSVP ne peut être envisagée. L'idée d'utiliser le QoS Agent peut alors s'avérer intéressante pour la gestion de la QoS pour un utilisateur mobile, mais cela implique de modifier quelque peu son fonctionnement.

Considérons la mise en place d'une communication VoIP entre un CN et le MN depuis son réseau d'origine. L'usage du proxy SIP conscient de la QoS permet d'obtenir une session avec QoS tant que le MN reste dans son réseau, mais, dès qu'il change de réseau, Mobile IPv6 va permettre de relancer la communication depuis le nouveau réseau mais sans garantie de QoS et sans libérer les ressources de l'ancien réseau. Nous cherchons donc à ce que le QoS Agent puisse réaliser ces différentes fonctions. Mais pour cela, plusieurs problèmes se posent pour cet outil :

- La découverte du QoS Server (son adresse IP) est statique. L'utilisateur doit la configurer lui-même. Mais dans un contexte de mobilité, on ne peut concevoir que l'utilisateur doive reconfigurer cette adresse à chaque changement de réseau.
- Le QoS Agent n'est pas prévu pour re-réserver la QoS au niveau du nouveau ST ni pour libérer la QoS au niveau de l'ancien ST.
- Le QoS Agent tient seulement compte des modifications de QoS qu'il a lui-même réalisé. Si la QoS a été configurée par d'autres applications (SIP par exemple), le QoS Agent n'en sera pas conscient et ne pourra pas la rétablir dans le nouveau réseau.

Si nous voulons mettre en œuvre une architecture de QoS compatible avec Mobile IPv6 et basée sur l'architecture QoS Agent/QoS Server, il faut donc les modifier en conséquence et leur ajouter les fonctionnalités suivantes :

- Détection automatique de l'adresse du QoS Server par le QoS Agent lors de la connexion à un réseau (ou lors d'un changement de réseau), ce qui revient dans le système satellite à détecter automatiquement l'adresse du ST auquel est raccordé le nouveau réseau.
- Enregistrement du QoS Agent auprès du QoS Server dès que la détection a été effectuée, cet enregistrement comprend les adresses IPv4 et IPv6 du QoS Agent et le numéro de port sur lequel le QoS Agent attend les messages du QoS Server.
- Désenregistrement lors de la déconnexion du réseau (ou d'un changement de réseau).

- Ajout d'un module au QoS Agent permettant de parcourir toutes les applications (connexions) ayant subi une modification de la QoS, de renégocier la QoS avec le nouveau QoS Server et de libérer la QoS auprès de l'ancien QoS Server.
- Ajout d'un module au QoS Server pour stocker la liste des QoS Agents enregistrés.
- Ajout d'un module au niveau du QoS Server qui lui permet, lorsqu'il reçoit une demande de réservation/libération de ressource qui ne provient pas d'un QoS Agent, de parcourir la liste des QoS Agents qui se sont enregistrés auprès de lui, de vérifier si la réservation/libération concerne l'un d'eux et, si c'est le cas, de prévenir le QoS Agent concerné. Le QoS Agent enregistre alors l'information et dans le cas d'un changement de réseau, il pourra automatiquement renégocier la QoS pour les applications concernées.

Enfin, reste le problème déjà énoncé à la fin de la partie III.3.5.1 de la configuration de la QoS dans les deux sens. Pour le cas d'une communication de VoIP, il est en effet nécessaire de mettre en place de la qualité de service au niveau des deux STs concernés par la communication. Mais si l'on considère qu'un QoS Agent est aussi présent au niveau du correspondant (ou CN pour Mobile IPv6), il permet donc aussi d'enregistrer toutes les réservations de QoS qui concernent ce CN. Il faudrait donc mettre en œuvre un moyen de communication entre Mobile IPv6 et le QoS Agent qui permettrait, dès que le MN change de réseau et envoie au CN un BU contenant sa nouvelle CoA, que le module Mobile IPv6 du CN prévienne le QoS Agent pour que celui-ci modifie les réservations de QoS auprès de son QoS Server en remplaçant l'ancienne adresse du MN par la nouvelle.

Bien sûr, toutes les modifications qui viennent d'être présentées supposent l'utilisation de l'optimisation de route car dans le cas contraire, la QoS serait plus compliquée à gérer et, surtout, cela impliquerait une consommation importante et sous optimale des ressources du système satellite. En effet, si l'on considère le cas où le MN, le CN et le HA sont tous connectés à des STs différents (par ex. après le déplacement 1 sur la Figure 39) et que tous les paquets passent par le HA, la QoS devra être configurée au niveau des 3 STs.

De même, la QoS ne peut être assurée pendant la phase transitoire d'optimisation de route. En effet, le temps que nécessiterait la mise en place de la QoS pour la phase de tunnel bidirectionnel serait le plus souvent supérieur à la phase elle-même. Il est donc préférable de se focaliser sur la réservation de ressources uniquement pour la phase optimisée.

III.5.2. FMIPv6 couplé au QoS Agent mobile

Dans le cas de FMIPv6, les mêmes fonctionnalités doivent être ajoutées au niveau du QoS Agent et du QoS Server. On considère aussi, pour les mêmes raisons que précédemment, que la QoS n'est pas gérée pour les différentes phases transitoires (tunnel entre le PAR et le NAR et phase de tunnel bidirectionnel par le HA) et qu'elle n'est gérée que pour la phase de communication directe entre le CN et le MN.

Compte tenu des caractéristiques de FMIPv6, d'autres fonctionnalités peuvent être envisagées pour optimiser les mécanismes liés à la réservation et à la libération de QoS. Ainsi, dans le cas prédictif, lorsque le MN obtient des informations sur le futur réseau auprès du PAR, il peut en profiter pour découvrir l'adresse du futur QoS Server associé et réaliser un

pré-enregistrement ainsi qu'une pré-réservation des ressources, avant même d'avoir changé de réseau. Par contre, dans le cas réactif, le fonctionnement restera le même que pour Mobile IPv6.

III.5.3. SIP pour la gestion de la mobilité et de la QoS pour les applications interactives

Dans ce cas, nous souhaitons définir une solution permettant la gestion de la mobilité et de la QoS dans un système DVB-S2/RCS uniquement par SIP, de façon indépendante de tout autre protocole de mobilité. Cette solution permettrait alors aux applications SIP de mettre en place la QoS et de gérer les changements de réseaux dans un système satellite de façon totalement automatique. Elle s'adresse donc plus spécifiquement aux applications interactives ayant des contraintes temporelles fortes (VoIP, vidéoconférence, etc…).

Concernant la QoS, un proxy SIP amélioré, comme spécifié dans la partie III.3.5.3, doit être localisé derrière chaque ST pour pouvoir intercepter les messages SIP et transmettre les informations de QoS au QoS Server associé ainsi qu'à l'ARC lorsque cela est nécessaire (pour des flux audio et vidéo). Dans ce cas, il n'est pas absolument nécessaire d'implémenter un module de découverte dynamique du QoS Server ou de l'ARC puisque le proxy SIP est fixe. L'adresse du QoS Server, son port d'écoute ainsi que l'adresse et le port d'écoute de l'ARC peuvent donc être configurés statiquement par l'administrateur réseau. Dans le cas de l'initiation de session, les réservations de QoS sont réalisées selon la procédure décrite par la Figure 31 en suivant les recommandations de la RFC 3312 [115] mais la ré-initiation de session et la re-réservation de ressources après un changement de réseau se fait selon le schéma de la Figure 38. Dans le cas d'une fin de session classique (messages BYE/OK), la libération (au niveau du QoS Server et de l'ARC) se fait au niveau du BYE. Par contre, dans le cas d'un changement de réseau, aucun message BYE n'est échangé avec l'ancien proxy SIP ; le seul message qui lui est envoyé est le message REGISTER (si l'ancien SIP proxy est aussi le proxy mère) ou DEREGISTER (pour les autres cas). La libération de ressource doit donc être initiée au moment de la réception de ce message qui indique dans tous les cas que le MN est maintenant localisé dans un autre réseau. Cette nouvelle fonctionnalité doit aussi être ajoutée au proxy SIP. De plus, l'échange de message REGISTER/DEREGISTER entre les différents proxies SIP n'est pas une fonctionnalité décrite dans la RFC et doit donc aussi être ajouté.

Enfin, si l'échange de message REGISTER/DEREGISTER permet de libérer les ressources au niveau de l'ancien ST en charge du MN, ce n'est pas le cas pour celui qui est en charge du CN. La libération de ressources doit alors être réalisée au niveau de la ré-initiation de la session. Lorsque le proxy SIP en charge du CN reçoit un message re-INVITE avec le même Call-Id que celui d'une session déjà en cours, il comprend alors que cela correspond à une modification de session, il analyse alors l'adresse du client SIP émetteur de ce message :

- Si l'adresse ne correspond à aucune des deux adresses des clients SIP participant à la session, il comprend que cela correspond à une modification de session avec changement de réseau et libère donc les ressources associées.
- Si l'adresse correspond à l'une des deux adresses des clients SIP participant à la session, cela correspond à une modification de session sans changement de réseau. Il

ne libère donc pas les ressources et attend de recevoir le message OK (ou Session Progress) pour modifier la réservation. En effet, dans ce cas, la session n'est pas interrompue et effectuer une libération pourrait dégrader la qualité de la communication jusqu'à la re-réservation.

Ensuite, lorsque le proxy SIP côté CN a détecté qu'il s'agissait d'une modification de session avec changement de réseau, deux possibilités sont envisageables lorsqu'il reçoit le message OK:

- Le MN s'est déplacé derrière un ST différent de celui du CN : dans ce cas là, le proxy SIP côté CN procède à une re-réservation de ressources auprès du QoS Server en précisant la nouvelle adresse et éventuellement les nouveaux paramètres de la session.
- Le MN s'est déplacé derrière le même ST que celui du CN : dans ce cas là, le proxy SIP repère que la session ne va plus traverser le système satellite et n'échange donc aucun message avec le QoS Server associé à son ST.

Toutes ces fonctionnalités doivent donc aussi être implémentées au niveau du proxy SIP et du QoS Server.

Concernant la mobilité, un client SIP, comme spécifié dans la partie III.4.1, doit être implémenté. Différentes fonctionnalités doivent donc y être ajoutées :

- Un module de détection de changement de réseau (et de découverte dynamique du proxy SIP local) interfacé avec le client SIP pour le prévenir qu'il doit lancer les procédures de réenregistrement et de ré-initiation de session.
- Envoi d'un message REGISTER spécifique pour indiquer la nécessité de prévenir d'autres proxies SIP.
- Mise à jour des paramètres de la session pour envoyer un message re-INVITE comprenant la nouvelle adresse du MN ainsi que les éventuels nouveaux paramètres SDP.

Il est à noter que peu de modifications sont ici nécessaires au niveau du CN, contrairement au cas de Mobile IPv6 où ce protocole doit y être aussi implémenté. Pour la solution de mobilité SIP, le CN voit effectivement un message re-INVITE comme une modification classique de la session existante et a donc juste besoin de mettre à jour les paramètres de la session. La seule modification nécessaire pour notre architecture est de lui indiquer que, dans le cas d'un re-INVITE avec changement de réseau, il doit répondre directement par un message OK (INVITE) sans passer par les messages 183 Session Progress, etc…

III.6. Conclusion

Dans ce troisième chapitre, nous avons donc pu concevoir deux types d'architecture de mobilité et de QoS pour les réseaux satellites DVB-S2/RCS: la première se base sur Mobile IPv6 ou ses extensions pour la partie mobilité et sur le système QoS Agent mobile/QoS Server en ce qui concerne la QoS alors que la deuxième est essentiellement construite autour de SIP en étendant les capacités des clients SIP pour les rendre conscients de la mobilité ainsi

qu'en donnant de nouvelles fonctionnalités de QoS et de gestion de la mobilité aux proxies SIP.

Pour cela, nous avons d'abord expliqué brièvement le fonctionnement d'un système satellite DVB-S2/RCS en présentant ces caractéristiques essentielles ainsi que les principaux éléments qui le composent.

Ensuite, après avoir défini les objectifs du projet européen SATSIX dans le cadre duquel la majeure partie de nos travaux ont été réalisés, nous avons pu présenter les thèmes majeurs de notre mémoire.

Nous avons donc commencé par aborder la qualité de service dans un réseau DVB-S2/RCS en présentant tout d'abord les mécanismes du DAMA, les différentes recommandations proposées pour les mécanismes de niveau MAC et IP ainsi que l'architecture de QoS spécifique au projet SATSIX. Ensuite, nous avons présenté les outils qui nous serviront de base pour nos implémentations: le QoS Agent, le QoS Server et le proxy SIP en précisant les différentes fonctionnalités qui doivent leur être ajoutées.

Ensuite, nous avons étudié le cas de la mobilité dans les réseaux DVB-S2/RCS en ne conservant que les solutions présentant, selon nous, le plus d'intérêt et ayant le plus de chance de voir le jour : Mobile IPv6, FMIPv6, HMIPv6 et mobilité SIP. Les solutions de la famille de Mobile IPv6 étant déjà spécifiées précisément par différents standards, nous nous sommes concentrés sur les différentes spécifications nécessaires à la mise en place de la mobilité SIP dans un système DVB-S2/RCS. Ainsi, différentes méthodes de réenregistrement /désenregistrement, puis de ré-initiation de session sont comparées pour pouvoir définir une solution optimale aussi bien en termes de temps nécessaires pour les réaliser que de nombre de messages à envoyer.

Une fois que les solutions de mobilité ont bien été définies, nous présentons une évaluation théorique des temps d'interruption et des délais de transfert de paquets pour chacune d'elles et pour différents cas de déplacements impliquant un système satellite DVB-S2/RCS. Cette évaluation nous permet alors avant tout de comparer ces solutions entre elles, mais aussi d'établir une série de recommandations concernant la mobilité dans un système satellite, en s'attachant particulièrement à vérifier la compatibilité de ces solutions avec des applications de type VoIP ou vidéoconférence.

Enfin, la dernière partie a été consacrée à l'objectif final de notre thèse : le couplage de la mobilité et de la QoS dans un système satellite DVB-S2/RCS. L'étude des deux architectures proposées (les solutions Mobile IPv6 + QoS Agent mobile et FMIPv6 + QoS Agent mobile étant considérées comme une seule architecture) permet alors de déterminer les dernières fonctionnalités à ajouter aux outils précédemment présentés.

Le quatrième chapitre de ce mémoire aura donc pour premier objectif de présenter les implémentations réalisées sur la base des spécifications établies dans ce chapitre afin de valider la pertinence de notre architecture. Ensuite, nous présenterons les différents résultats obtenus lors des évaluations réalisées sur PLATINE, la plateforme d'émulation satellite mise en œuvre au cours du projet SATSIX après avoir été initiée dans le projet SATIP6.

IV. Implémentations et Evaluations

Afin de valider les deux architectures de QoS et de mobilité pour les réseaux DVB-S2/RCS définies précédemment, une série de tests a été menée sur une plateforme d'émulation nommée PLATINE. Nous avons décidé de valider nos travaux dans le cadre de l'émulation d'une part parce qu'une plateforme devait en effet être mise en œuvre dans le cadre du projet SATSIX et d'autre part parce que nous jugeons les résultats obtenus plus proches de la réalité que dans le cadre de la simulation ; par exemple l'émulation permet d'échanger réellement de vrais paquets entre différentes machines physiques. D'autre part, les outils développés pour la réalisation de nos expériences peuvent directement être utilisés pour des tests sur des réseaux réels.

Pour bien comprendre les expériences menées et l'analyse des résultats obtenus, la première partie de ce chapitre est alors consacrée à la description de la plateforme d'émulation elle-même ainsi que des différents outils qui ont été utilisés et/ou implémentés pour nos différents tests en fonction des spécifications décrites dans le chapitre III.

Ensuite, les parties suivantes décrivent les résultats obtenus lors de nos campagnes de tests en envisageant différents types de scénarios que ce soit pour les tests de mobilité, de QoS ou des deux à la fois.

IV.1. PLATINE : la plateforme d'émulation

Comme nous l'avons précisé dans le paragraphe III.2, l'un des objectifs du projet SATSIX est de développer une plateforme capable d'émuler précisément les différentes caractéristiques d'un réseau satellite DVB-S2/RCS afin de pouvoir mesurer l'impact qu'auraient les nouvelles applications IPv6 et les services liés aux NGN (mobilité, QoS, sécurité, mulitcast, etc…) si on les intégrait à ce type de réseau.

Pour cela, la plateforme PLATINE [148], initiée dans le cadre du projet SATIP6, se veut pleinement compatible avec l'architecture adoptée par le groupe ETSI BSM [147] ainsi qu'avec les standards DVB-S2 et DVB-RCS et suit les recommandations du groupe SatLabs en matière de QoS (voir paragraphe III.3.4). Elle est implémentée sur un système Linux Fedora Core 5, choisi d'une part pour son support natif d'IPv6 (en plus d'IPv4) et d'autre part pour ses caractéristiques permettant une configuration précise des paramètres réseaux ou encore de la QoS.

IV.1.1. Environnement de développement de la plateforme d'émulation

Le cœur de l'architecture de PLATINE est basé sur un moteur d'exécution C++ appelé Margouilla [149] qui fournit un environnement de développement et d'exécution indépendant du système d'exploitation utilisé, des outils de synchronisation, une série de protocoles déjà implémentés (IP, ATM, Ethernet, …) ainsi que des fonctionnalités permettant par exemple de générer des traces ou des messages d'erreur. Les blocs Margouilla développés pour les besoins de la plateforme sont les suivants :

- Le bloc *Satellite Carrier*, responsable de l'émulation des porteuses (DVB-RCS, DVB-S2 et canaux de signalisation) ainsi que de l'émulation du délai et des taux d'erreur sur le lien satellite.
- Le bloc *DVB-RCS*, qui implémente une structure de trame compatible avec les standards DVB-S2 et DVB-RCS et remplit les trames DVB-RCS avec les paquets des couches supérieures (ATM ou MPEG2-TS) provenant du bloc *Encapsulation/Desencapsulation*. Afin de mettre en œuvre correctement les mécanismes de QoS, ce bloc gère la synchronisation et les files d'attente en fonction du bloc *DAMA*.
- Le bloc *DAMA*, qui implémente les algorithmes du DAMA utilisés pour gérer l'allocation des ressources satellite à la couche MAC.
- Le bloc *Encapsulation/Desencapsulation*, qui implémente les mécanismes d'encapsulation AAL5 et ULE et qui est en charge des fonctions de segmentation et de réassemblage.
- Le bloc *IP QoS*, qui est en fait une couche intermédiaire entre le noyau Linux et PLATINE. Ce bloc récupère les paquets IP classifiés par TC (*Traffic Control*) et les transmet aux blocs spécifiques à PLATINE. Ce bloc repose sur la notion de CQD (*Classful Queuing Discipline* ou gestionnaire de mise en file d'attente basé sur des classes) qui permet de traiter différemment des flux selon leur classe, et pour implémenter le CQD, l'algorithme d'ordonnancement est basé sur HTB (*Hierarchical Token Bucket*).

La Figure 40 présente comment les différents blocs s'articulent entre eux au niveau des trois entités essentielles du système satellite : ST, Satellite et GW/NCC.

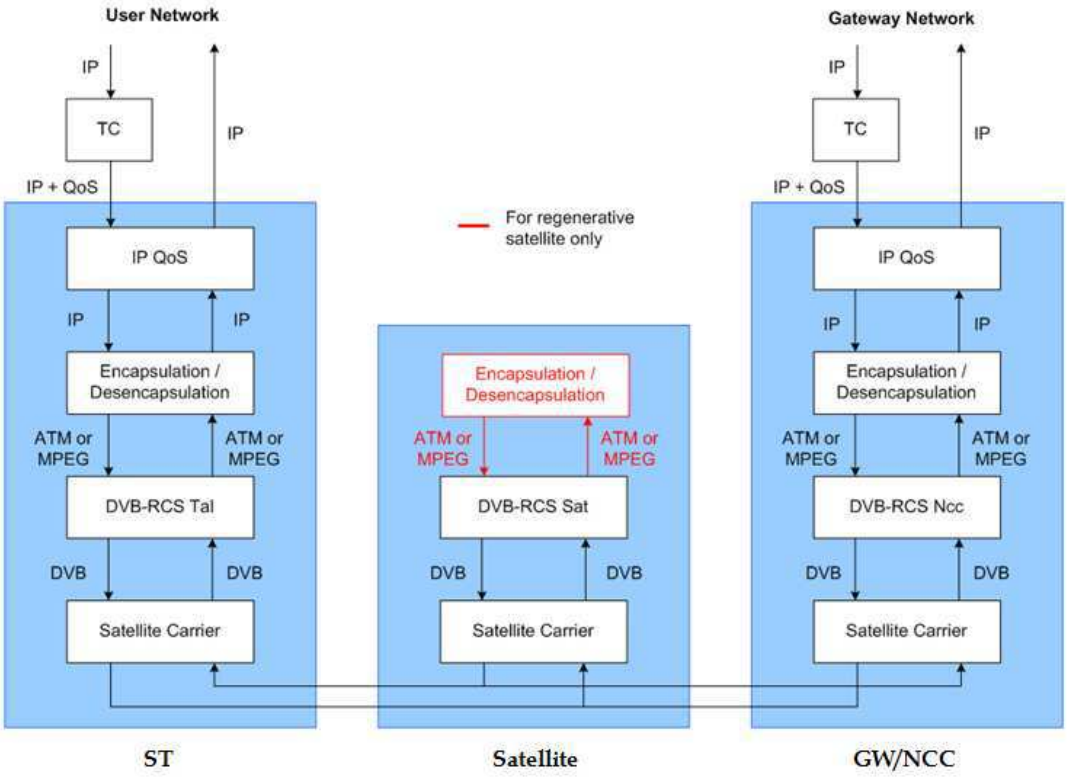


Figure 40 – Les différents blocs nécessaires à la plateforme PLATINE

IV.1.2. Les différents éléments de la plateforme

Pour permettre l'émulation de scénarios divers, chaque élément d'un réseau satellite DVB-S2/RCS est émulé sur un nœud dédié, comme on peut le voir sur la Figure 41. L'élément central en est l'émulateur satellite (SE). Cet émulateur a été initié dans le cadre du projet européen BRAHMS précédant SATIP6 puis SATSIX. Il permet l'émulation de différents types de système satellite grâce à une possible configuration du délai, de la gigue, des taux d'erreur, du nombre de spots, du type de satellite (transparent, régénératif ou mixte) et du type de connectivité (maillée, étoilée). Les délais et les modèles d'erreurs peuvent être basés sur des modèles statistiques classiques ou basés sur des distributions pré-calculées, ces dernières étant obtenues à partir de mesures réelles. De plus, les liaisons montantes et descendantes sont émulées au dessus d'Ethernet (choisi essentiellement pour ses capacités en termes de bande passante). Ainsi, chaque trame DVB est encapsulée dans une trame Ethernet et chaque canal satellite correspondant à une porteuse est associé à une adresse multicast. Un spot est alors composé de quatre canaux :

- Un canal dédié aux données DVB-S2 sur une liaison descendante.
- Un canal dédié aux données DVB-RCS sur une liaison montante.
- Deux canaux dédiés à la signalisation de contrôle et de gestion (e.g TBTP), un pour la liaison montante et l'autre pour la descendante.

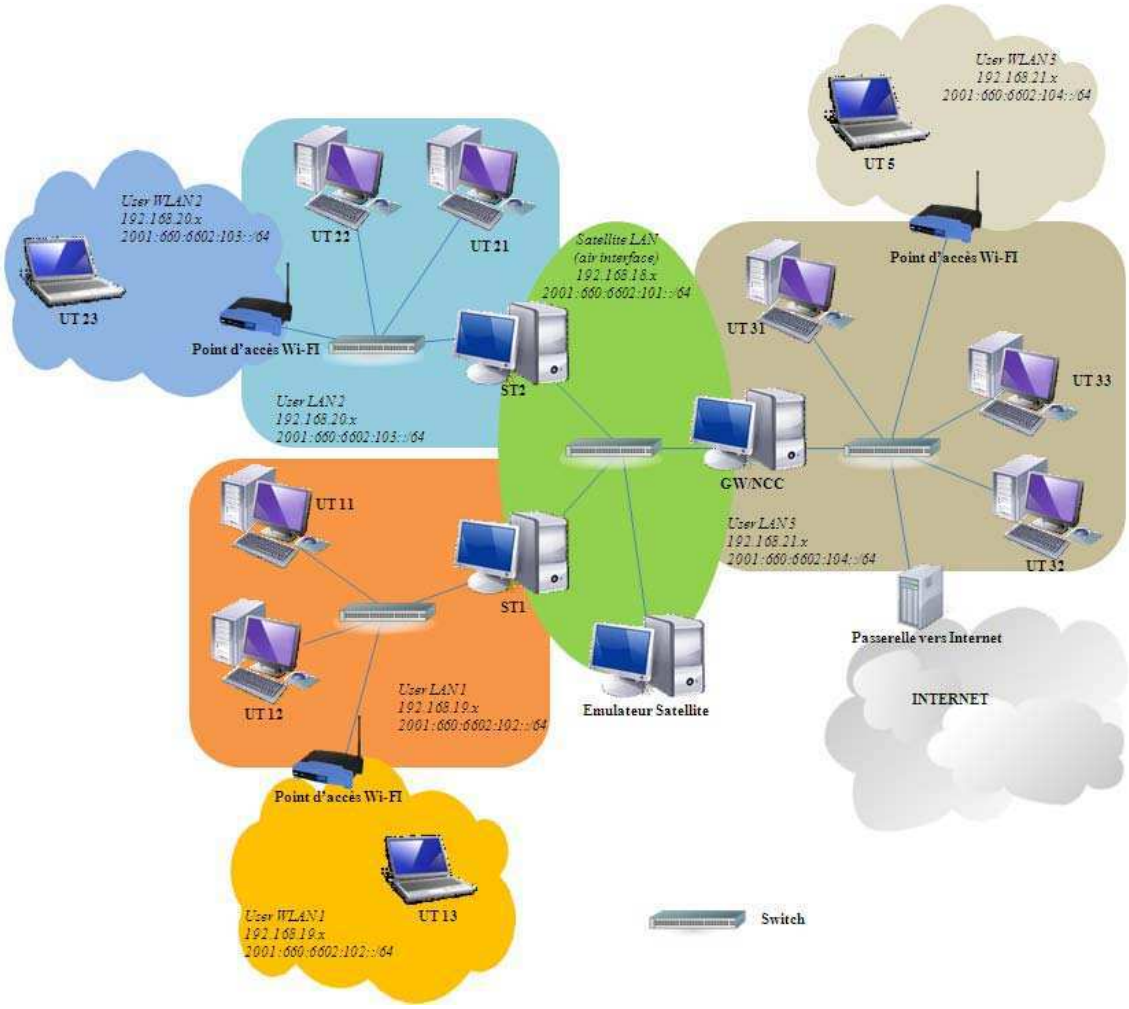


Figure 41 – Eléments de la plateforme d'émulation

Le réseau Ethernet, émulant le lien satellite permet alors d'interconnecter le SE avec les STs et la GW/NCC qui, eux-mêmes, sont reliés aux terminaux utilisateurs (UT) par l'intermédiaire de réseaux Ethernet et/ou d'un réseau de type Wi-Fi par exemple. Le nombre de terminaux utilisateurs peut varier dans chaque *User LAN* et *User WLAN* selon les besoins de l'expérience et aucun système d'exploitation n'est imposé.

IV.2. Les outils nécessaire à l'évaluation

Pour réaliser nos expériences de mobilité et/ou de QoS ou tout simplement mettre en œuvre un réseau satellite réaliste, différents outils ont dû être implémentés, modifiés ou tout simplement utilisés. L'objectif de cette partie est de présenter ces outils.

IV.2.1. Les outils de base

Pour pouvoir réaliser des expériences au plus proche de la réalité, il est tout d'abord important d'installer au niveau de chaque LAN utilisateur (derrière chaque ST ou GW) : un serveur DNS, un serveur DHCP (pour IPv4) et de mettre en œuvre les mécanismes d'autoconfiguration d'adresse IPv6. En effet, on considère que derrière chaque ST se situe un domaine (ou sous-domaine) à part entière. Pour implémenter ces fonctionnalités, nous avons choisi d'utiliser les outils suivants :

- Simple DNS Plus [151] v4.00.06 qui supporte aussi bien IPv4 qu'IPv6 et qui permet de gérer les principaux enregistrements DNS dont nous avons besoin pour nos expériences : les classiques « A record » (nom d'hôte → adresse IPv4) et « AAAA record » (nom d'hôte → adresse IPv6) et le « SRV record » permettant de connaître l'adresse du proxy SIP en charge d'un domaine (_sip._udp.domaine.org → adresse du proxy SIP en charge de domaine.org).
- ISC (*Internet Systems Consortium*) DHCP Server [152] v3.0.5 qui nous permet de définir le masque de sous-réseau du client, l'adresse IP du routeur présent sur le sous-réseau du client (le ST dans notre cas), le nom de domaine, l'adresse du serveur DNS et enfin l'option permettant d'attribuer une adresse IPv4 au client.
- Linux IPv6 Router Advertisement Daemon (radvd) [153], un démon fonctionnant sous Linux et permettant d'envoyer des messages *Router Advertisement*, comme spécifié dans [23], indiquant le préfixe réseau et l'adresse MAC du routeur (le ST dans notre cas).
- NTP Daemon v4.2.4, un démon NTP (*Network Time Protocol*) [154] fonctionnant sous Linux et permettant de synchroniser temporellement différentes machines sur le principe client-serveur avec possibilité d'avoir plusieurs niveaux de hiérarchie.

IV.2.2. Les outils pour la capture, le rejeu et l'analyse des flux

Dans le cadre de certaines évaluations, la même expérience doit être réalisée un certain nombre de fois en changeant certains paramètres du système satellite (bande passante totale allouée, quantité de CRA, de RBDC, etc…). Pour être bien sûr que les différences que nous observons sont bien dues aux variations du système et non aux variations de certains paramètres des flux, il est donc nécessaire de rejouer exactement les mêmes flux pour chaque

expérience. En effet, dans le cas d'une vidéoconférence par exemple, le débit correspondant au flux vidéo peut beaucoup varier d'une session à l'autre en fonction des images à transmettre. Pour cela, nous avons du utiliser différents outils de capture, de rejeu et d'analyse de trafic que nous allons maintenant présenter.

IV.2.2.1. FLOC : un outil de capture de trafic

L'outil FLOC (*FLOw Capturer*) a été développé au LAAS et fait partie d'un ensemble de 3 logiciels appelés (FL3 (*FLOC : FLOw Capturer, FLORE : FLOw REplayer, FLAN : FLow ANalyzer*) [155] permettant la capture, le rejeu et l'analyse de flux. Ce logiciel a été conçu pour permettre de capturer à la volée, pendant une durée donnée, l'ensemble des données IPv4 ou IPv6 véhiculées par toutes les connexions provenant d'une application. Cet outil utilise les mêmes fonctionnalités que le traceur de connexion implémenté pour le QoS Agent et permet d'obtenir le nombre de connexions ouvertes, le protocole de transport associé à chaque connexion (UDP ou TCP), la date d'émission de chaque paquet capturé et la taille du paquet (charge utile du niveau transport). Ces deux dernières informations permettent d'obtenir deux fichiers contenant les informations nécessaires au rejeu de ces flux : le délai inter-paquet et la taille des paquets.

IV.2.2.2. JTG : un outil de génération, de rejeu et d'analyse de trafic

Pour la génération, le rejeu et l'analyse de trafic, nous avons choisi d'utiliser l'outil JTG (*Jugi's Traffic Generator*) [156] développé par l'université d'Helsinki. Ce générateur de trafic simple et précis permet d'échanger entre un émetteur et un récepteur des flux UDP et TCP totalement configurables et notamment permet de rejouer des flux capturés à partir de deux informations obtenues grâce à l'outil FLOC : le délai inter-paquet et la taille des paquets.

JTG présente aussi l'avantage de pouvoir analyser les flux à partir des fichiers log, au format MGEN (*Multi-GENerator*) [157], obtenus au niveau du récepteur et de tracer les courbes correspondantes à différentes caractéristiques du flux : la gigue, le délai, la variation du délai inter-paquet ainsi que les pertes. Ces courbes sont réalisées en utilisant l'outil jtg_calc de jtg puis gnuplot. C'est principalement pour ces raisons que nous avons choisi cet outil plutôt que les parties *FLORE* et *FLAN* de Fl3.

Dans le cadre de nos expériences, nous avons dû effectuer certaines modifications sur cet outil :

- Pour permettre la gestion de flux IPv6, il existe une version JTG6 mais cette dernière ne fonctionne que sous certains systèmes d'exploitation n'incluant pas Fedora Core. Il a donc fallu modifier le code source pour ajouter cette fonctionnalité.
- Nous avons aussi ajouté la possibilité d'analyser et de tracer la courbe du débit correspondant à un flux. Plusieurs méthodes de calcul de débit ont ainsi été ajoutées.
- Enfin, dans le cadre de nos expériences sur Mobile IPv6, nous avons modifié le code source pour que le flux généré par un terminal mobile puisse continuer pendant et après un changement de réseau sans s'interrompre.

Il est important de noter que pour obtenir des résultats concluants, il est indispensable de synchroniser temporellement l'émetteur et le récepteur concerné. Pour cela, nous avons utilisé

la synchronisation NTP en utilisant un modèle hiérarchique. Sur la Figure 41, l'émulateur satellite sert de référence temporelle surlaquelle se synchronisent les STs et GW. De même, les clients se synchronisent sur leur ST/GW respectif, ce qui nous permet d'obtenir une précision inférieure à la milliseconde sur un réseau LAN et de l'ordre de quelques millisecondes sur un réseau WLAN.

IV.2.2.3. ORENETA

Dans le cadre des expériences correspondant aux évaluations de QoS ou de Mobile IPv6 et de ses extensions, il est possible de réaliser les tests en rejouant les différents flux mais dans le cas de la mobilité SIP, pour tester les performances de notre architecture, il nous faut utiliser les logiciels SIP puisque ce sont eux qui initient les mécanismes de mobilité et de QoS et qui permettent donc aux flux de reprendre depuis le nouveau réseau.

Cependant, pour pouvoir comparer la mobilité SIP à Mobile IPv6, il est nécessaire de pouvoir obtenir des fichiers log permettant d'analyser les flux et de tracer les mêmes courbes que celles obtenues avec JTG. C'est pour cette raison que nous avons choisi d'utiliser l'outil ORENETA (*One way delay REal-time NETwork Analyser*) [158] qui permet de capturer à la volée différents flux identifiés par leur quadruplet {port source, adresse source, port destination, adresse destination} au niveau de deux entités appelées *meters* et localisées au niveau de l'émetteur et du récepteur des flux que l'on cherche à étudier. Ces deux entités transmettent également à la volée les informations concernant les paquets capturés vers un serveur central appelé *analyzer* qui permet d'une part de visualiser en direct certaines caractéristiques concernant les flux (débit, délai, pertes, variation du délai) mais aussi de créer des fichiers log au format MGEN à la fin de la communication. Ces fichiers peuvent ensuite être analysés par JTG de la même manière que précédemment si bien sûr les différents *meters* sont synchronisés temporellement (pour cela, nous avons utilisé la synchronisation NTP).

Le désavantage d'un tel outil est qu'il génère lui-même du trafic pour transmettre les données concernant les paquets capturés. Pour ne pas fausser nos évaluations, il a donc fallu mettre en œuvre un réseau spécialement dédié à ce trafic. La Figure 42 résume les interactions existant entre nos différents outils de mesures.

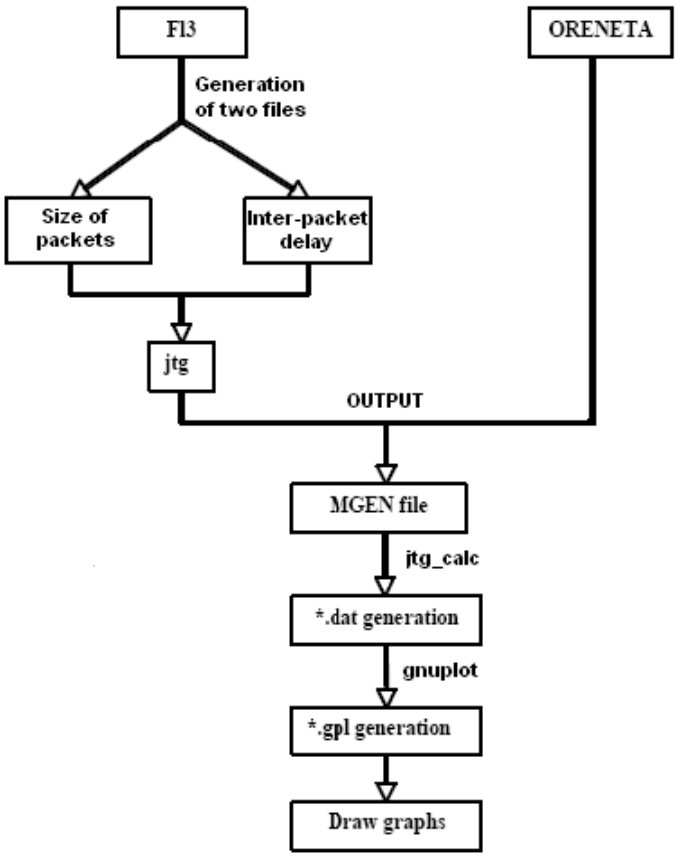


Figure 42 – Interactions entre les différents outils de mesure utilisés

IV.2.3. Les outils pour la mobilité

Comme on a pu le voir dans le chapitre précédent, en ce qui concerne la mobilité, nous nous sommes principalement intéressés à la famille des protocoles basés sur Mobile IPv6 (essentiellement Mobile IPv6 et FMIPv6) ainsi qu'à la solution de mobilité SIP. Nous allons donc présenter les différents outils qui nous ont permis d'évaluer ces solutions.

IV.2.3.1. Mobile IPv6 et FMIPv6

Pour ce qui est de Mobile IPv6, nous avons tout d'abord utilisé la pile UMIP (USAGI-patched Mobile IPv6 for Linux) v0.4 [159], une pile Mobile IPv6 open-source, distribuée sous licence GPLv2 pour les systèmes d'exploitation Linux. Elle est basée sur la pile MIPL2.0.2 et est maintenue pour fonctionner sur les derniers noyaux. Elle supporte les standards [31] et [36]. Pour permettre les mécanismes d'optimisation de route, indispensables dans le cadre d'un système satellite, il est donc nécessaire d'installer la pile UMIP sur 3 entités au minimum : le HA, le MN et le CN et configurer proprement les différents fichiers de configuration.

Nous avons tout d'abord utilisé cette pile UMIP v0.4 puisqu'elle présentait l'avantage d'être utilisée conjointement avec la pile FMIPv6 [160] proposée par le projet fmipv6.org qui fournit une implémentation totalement compatible avec la RFC 4068 [161] qui a très récemment été remplacée par la RFC 5568 [15]. Cette pile, qui fonctionne elle aussi sur un noyau Linux permet donc d'améliorer le temps de latence dû aux mécanismes de Mobile IPv6. Pour fonctionner correctement, la pile FMIPv6 doit être installée sur au moins deux

routeurs d'accès (un PAR et un NAR) ainsi que sur le MN. Le HA et le CN ne nécessitent pas son installation puisqu'ils sont non conscients des mécanismes spécifiques à FMIPv6 ; par contre, ils requièrent évidemment la pile UMIP.

Cependant, certains bugs de la pile UMIPv0.4 allongeaient fortement le temps de handover et nous ont donc conduit à finalement utiliser la pile MIPL2.0.2 sur un noyau 2.6.16 pour nos tests Mobile IPv6.

De plus, concernant FMIPv6, nous avons pu mettre en œuvre un banc de test correctement configuré et avons pu observer les échanges des messages spécifiques à FMIPv6 (FBU, FBACK, HI, HACK et UNA) et la mise en œuvre de tunnel mais pour des raisons inconnues (peut être des problèmes de compatibilité avec notre plateforme d'émulation satellite), la communication ne reprenait pas par les tunnels FMIPv6. Les évaluations de cette solution seront donc partielles par rapport à celles obtenues avec Mobile IPv6 et la mobilité SIP.

IV.2.3.2. La mobilité SIP

IV.2.3.2.a Le client SIP : VisioSIP

Pour le cas de la mobilité SIP, nous avons choisi d'implémenter la partie mobilité pure par nous même sur un client SIP initialement développé par le LAAS et Silogic [162] lors du projet EuQoS et nommé VisioSIP (voir Figure 43). Ce client SIP, développé en JAVA pour pouvoir être utilisé aussi bien sous Linux que sous Windows utilise les librairies JAIN-SIP 1.2 [163] comme base de développement et a été implémenté pour être pleinement compatible avec les RFC 3261 [69] et 3312 [115]. Il est aussi compatible IPv4/IPv6.

Figure 43 – VisioSIP lors de l'enregistrement d'un utilisateur

Cependant, les fonctionnalités spécifiées dans le paragraphe III.5.3 doivent y être ajoutées. Tout d'abord, **un module de détection de changement d'adresse doit être ajouté** et doit fonctionner indépendamment du protocole utilisé (IPv4 ou IPv6) et du système d'exploitation (Linux ou Windows). Pour cela, nous avons tout d'abord choisi de sonder périodiquement les adresses IP courantes du client (avec une période très faible pour que cela soit négligeable au niveau du temps d'interruption). Dès qu'un changement de réseau a lieu, ce module interroge le serveur DNS local (dont il a obtenu l'adresse par l'intermédiaire des mécanismes de DHCP) pour connaître l'adresse du proxy SIP en charge du domaine courant

et ainsi lui envoyer une requête REGISTER puis, presque simultanément, un message re-INVITE.

Cependant, dans le cas de Linux, nous avons implémenté une méthode « cross-layer » plus efficace basée sur le principe du « debugfs » qui permet de créer des fichiers dans lesquels les drivers ou les programmes utilisateurs pourront écrire et lire des valeurs. Ainsi, lorsque le driver Wi-Fi détecte un évenement « Link Down », il peut, par l'intermédiaire du « debugfs », d'une part permettre d'initier la procédure de DHCP et/ou l'envoi d'un message *Router Sollicitation* et d'autre part indiquer au client SIP qu'il va y avoir un changement de réseau (et potentiellement d'adresse IP). Le client SIP sonde donc les adresses IP, les met à jour si nécessaire et attend de nouveau jusqu'au prochain évènement « Link Down ». Cela peut permettre de gagner un peu de temps pour la reconfiguration d'adresse et cela évite au client SIP de sonder en permanence les adresses et donc de consommer des ressources inutilement. Enfin, dans le cas d'IPv6, il n'est alors plus nécessaire d'envoyer des RA toutes les 50 ms environ, ce qui libère aussi des ressources importantes au niveau du lien local.

En ce qui concerne le message REGISTER, nous avons décidé de n'envoyer qu'un seul message, comme cela est précisé dans le paragraphe III.4.1.1, mais il est tout de même nécessaire d'envoyer un message qui diffère d'un message classique pour pouvoir indiquer au proxy SIP local à qui il doit transférer tel ou tel message.

Ainsi, pour **différencier un enregistrement classique d'un enregistrement avec nécessité de prévenir le proxy SIP du réseau mère**, nous avons utilisé la possibilité d'avoir un champ « From » différent du champ « To » dans le message REGISTER comme autorisé par la norme [69]: «The From header field contains the address-of-record of the person responsible for the registration. The value is the same as the To header field unless the request is a third-party registration ». On considère ici que l'utilisateur, bien qu'il soit la même personne physique, représente une « third party » vu qu'il a changé de domaine et d'adresse. Ainsi, le proxy SIP reçoit un message REGISTER avec un champ « From » comportant l'adresse « user@DomaineMere.org » et un champ « To » avec l'adresse « user@DomaineVisité.org ». En voyant ces deux champs différents, le proxy SIP local envoie automatiquement une requête DNS de type SRV permettant de connaître le proxy SIP en charge du domaine correspondant au champ « From » (s'il ne le connaît pas déjà) et prévient ce dernier que c'est maintenant lui qui se charge des requêtes de l'utilisateur mobile. Le proxy SIP en charge du réseau mère doit alors transmettre toutes les requêtes à destination de « user@DomaineMere.org » vers « user@DomaineVisité.org ». Nous étudierons plus précisément cette fonctionnalité dans le paragraphe IV.2.3.2.b.

Enfin, dans le cas où le client SIP mobile doit aussi **prévenir l'ancien proxy SIP pour qu'il le désenregistre et qu'il libère les ressources** qui lui étaient allouées, nous avons utilisé le champ «Message Body » inutilisé pour les messages REGISTER pour y inscrire le nom de domaine correspondant au dernier réseau visité. Le contenu de ce message Body est décrit dans le champ « Content-Type ».

La Figure 44 donne un exemple de ce type de message REGISTER en considérant que l'utilisateur *kiaso* a pour réseau mère *laas.org* (et donc comme SIP URI, *sip:kiaso@laas.org*). Le message REGISTER présenté ci-après est envoyé par le client SIP alors qu'il vient de se déplacer du réseau *satsix.org* vers le réseau *iptel.org* (il avait donc déjà effectué un

déplacement de *laas.org* vers *satsix.org* auparavant). Les parties encadrées de ce message présentent alors les différences avec un REGISTER classique. On voit effectivement que les champs « From » et « To » correspondent respectivement à l'URI mère et à l'URI courante du client, ce qui indique que le proxy SIP en charge du domaine *iptel.org* doit envoyer un message REGISTER au proxy SIP mère (en charge du domaine *laas.org*). Ce message aura alors pour champ « From », *sip:proxy.iptel.org* (l'adresse du proxy SIP local) et pour champ « To » *sip:kiaso#laas.org@iptel.org* ce qui indiquera au proxy SIP mère qu'il doit désormais transférer toutes les requêtes à destination de *kiaso@laas.org* vers *kiaso@iptel.org* (nous voyons ce point plus en détail dans le paragraphe suivant IV.2.3.2.b).

De plus, on peut remarquer la présence d'un champ « Content-Type » indiquant *lastdomain/satsix.org*, *satsix.org* étant donc le contenu du « Message Body ». Ceci permet alors au proxy SIP local de savoir qu'il doit envoyer un message DEREGISTER au proxy SIP en charge du dernier domaine visité, *satsix.org*. Ce message aura alors pour champ « From », *sip:proxy.iptel.org* et pour champ « To » *sip:kiaso@satsix.org* et aura le champ « Expires » à 0.

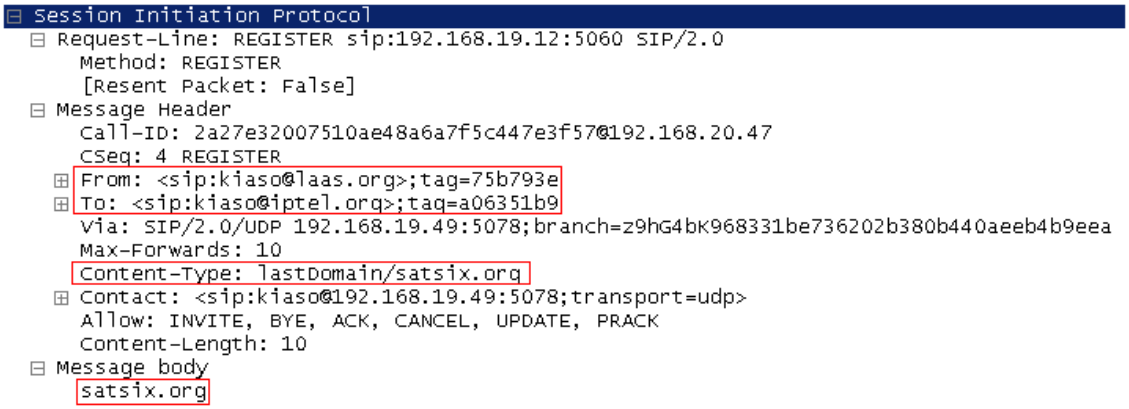


Figure 44 – Format spécial d'un message REGISTER après changement de réseau

En plus des implémentations nécessaires à la mise en œuvre de ce message REGISTER spécifique, d'autres modifications ont été nécessaires pour la ré-initiation de session. En effet, comme cela est précisé dans le paragraphe III.4.1.2, nous avons choisi de ré-initier la session uniquement en échangeant les messages re-INVITE/OK/ACK. Il faut donc :

- Au niveau du client SIP mobile : lorsqu'un changement de réseau est détecté en cours de communication, le client SIP doit envoyer à son correspondant un message re-INVITE ayant le même call-ID que le message INVITE initial (pour que le correspondant puisse comprendre que c'est juste une modification d'une session déjà existante et pas une nouvelle session). De plus, il doit modifier son adresse dans les différents champs qui la comprennent (« Contact », « Via » ainsi que dans les SDP).
- Au niveau du client SIP correspondant : lorsqu'il reçoit un nouveau message re-INVITE ayant le même call-ID qu'une session SIP en cours, il modifie les paramètres de la session et répond directement par un message OK.

IV.2.3.2.b Le proxy SIP : JAIN-SIP-PRESENCE-PROXY

Les nouvelles fonctionnalités qui doivent être mise en place au niveau d'un proxy SIP en ce qui concerne la mobilité ont déjà été abordées dans le paragraphe précédent :

- A la réception d'un message REGISTER spécifique, le proxy SIP doit construire un ou plusieurs messages REGISTER pour prévenir le ou les autres proxies SIP concernés par le changement de réseau.
- Lorsque le proxy SIP mère reçoit un message REGISTER de mise à jour de l'enregistrement d'un client SIP envoyé par un autre proxy SIP, il doit l'accepter et automatiquement transférer les messages SIP à destination de ce client vers le nouveau proxy SIP qui en a la charge.
- Le proxy SIP doit pouvoir accepter les requêtes de désenregistrement de la part d'autres proxies SIP (en plus des messages de désenregistrement classiques envoyés par un client SIP lui-même).

Par souci de cohérence avec notre choix concernant le client SIP, nous avons là aussi choisi d'utiliser un proxy SIP utilisant les mêmes librairies JAIN-SIP 1.2. Nous avons alors choisi le JAIN-SIP-PRESENCE-PROXY (ou NIST-SIP PROXY) [164] qui implémente les principales fonctionnalités d'un proxy SIP et d'un serveur d'enregistrement (voir Figure 45). Il est compatible avec IPv4 et IPv6 et permet aussi de fonctionner sous Windows comme sous Linux.

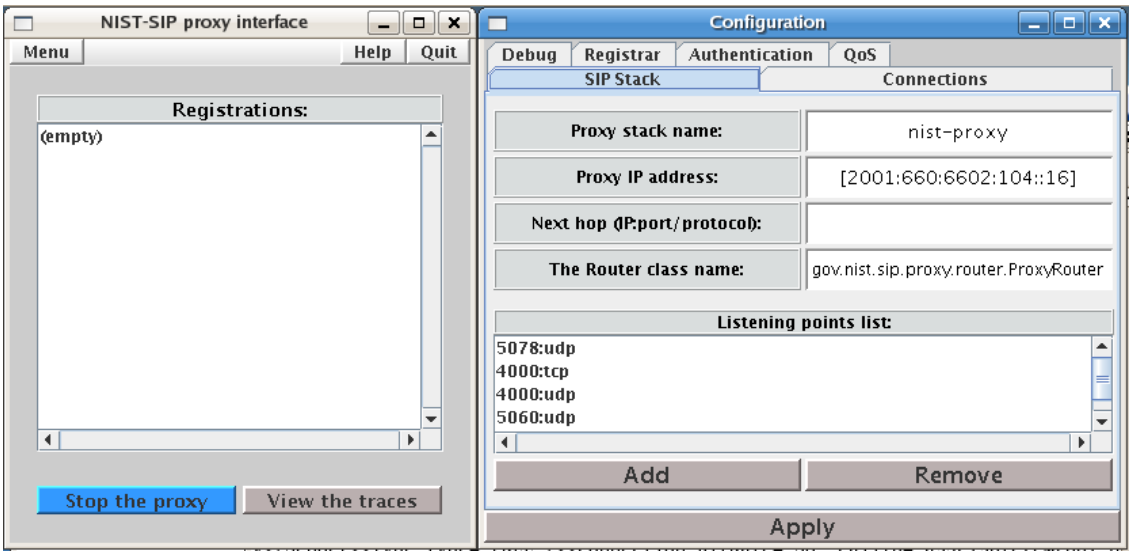


Figure 45 – le proxy SIP après enregistrement de l'utilisateur bat@iptel.org

Concernant les nouvelles implémentations à réaliser sur le proxy SIP, il faut tout d'abord lui permettre de **repérer un message REGISTER spécifique** lui indiquant qu'il doit prévenir un ou plusieurs autres proxies SIP. Pour cela, à la réception d'un message REGISTER, s'il repère que les champs « From » et « To » sont différents et que les deux SIP URI sont de la forme *sip:nom_user@nom_domaine* avec le même *nom_user* mais avec deux *nom_domaine* différents, il sait qu'il doit envoyer un message REGISTER au proxy SIP en charge du domaine mère correspondant au *nom_domaine* du champ « From ». Ce message aura les spécificités indiquées dans le paragraphe IV.2.3.2.a.

De plus, lorsqu'il reçoit un message REGISTER, il vérifie aussi la présence d'un champ « Message Body ». Si c'est le cas, il doit alors envoyer un message DEREGISTER, comme indiqué dans le paragraphe IV.2.3.2.a, après éventuellement avoir envoyé une requête DNS au serveur local pour connaître l'adresse du proxy SIP en charge du domaine indiqué dans le « Message Body ».

Ensuite, il faut que le proxy SIP mère en charge d'un client mobile soit capable de **transférer les messages SIP qui lui sont destinés vers son domaine courant** (vers le proxy SIP en charge de son domaine courant). Pour cela, le proxy SIP vérifie tout d'abord si le REGISTER provient d'un client SIP ou d'un autre proxy SIP : si le message REGISTER contient un champ « From » et un champ « To » différents, avec un champ « From » n'étant pas de la forme classique *sip:nom_user@nom_domaine* (mais juste *sip:nom*), il sait que le message provient d'un autre proxy SIP. Il vérifie alors si le message d'enregistrement concerne un utilisateur qui est déjà enregistré dans sa base de données en analysant le *nom_user* du champ « To » : si le *nom_user* est de la forme mot1#mot2 avec mot1 et mot2 respectivement équivalents au *nom_user* et au *nom_domaine* d'un utilisateur présent dans sa base, le proxy SIP met alors à jour sa nouvelle localisation et à partir de ce moment là, transfère tous les messages SIP qui sont destinés à cet utilisateur vers le proxy SIP correspondant au *nom_domaine* du champ « To ».

Pour reprendre l'exemple du paragraphe précédent, le proxy SIP mère (en charge du domaine *laas.org* et ayant dans sa base de données un utilisateur enregistré sous la SIP URI *sip:kiaso@laas.org*) reçoit un message REGISTER avec un champ « From » de la forme *sip:proxy.iptel.org*. Il sait alors que la requête provient d'un proxy SIP et analyse alors le champ « To » qui est de la forme *sip:kiaso#laas.org@iptel.org*. Il compare alors *kiaso#laas.org* avec les utilisateurs enregistrés auprès de lui et trouve effectivement un *sip:kiaso@laas.org*. Il sait alors qu'il doit transférer toutes les requêtes à destination de *kiaso@laas.org* vers *kiaso@iptel.org*.

Enfin, pour la question d'un **désenregistrement initié par un autre proxy SIP**, un proxy SIP désenregistre un utilisateur lorsqu'il voit arriver un message REGISTER avec :

- Un champ « From » de la forme *sip:nom*.
- Un champ « To » correspondant exactement à l'URI d'un utilisateur enregistré dans sa base.
- Un champ « Expires » à 0.

IV.2.4. Les outils pour la QoS

Dans cette partie, nous décrivons successivement les outils que nous avons dû développer ou utiliser pour réaliser nos expériences concernant la QoS.

IV.2.4.1. Le QoS Server amélioré

L'outil communément utilisé par nos deux architectures (SIP d'un côté et MobileIPv6 ou FMIPv6 de l'autre) est le QoS Server puisque sa fonction principale est de recevoir et d'analyser les messages de réservation et de libération en provenance du proxy SIP ou du QoS Agent (seulement dans les cas de mobilité + QoS ; nous n'utiliserons pas la possibilité

qu'à un utilisateur de configurer par lui-même la QoS de ses applications avec le QoS Agent pour les raisons indiquées dans le paragraphe III.3.5.2). De plus, des messages RESV et FREE pourraient aussi être envoyés par d'autres entités améliorées pour la QoS, spécifiques à d'autres types d'application.

Pour recevoir les messages RESV/FREE, le QoS Server est donc configuré pour écouter sur un port spécifique (12000 pour nos expériences) et, lorsqu'il reçoit un message de type XML, il l'analyse et agit en conséquence. Cependant, un certain nombre de modifications doit être ajouté comme cela a été précisé dans les paragraphes III.3.5.1 et III.5.1.

Tout d'abord, il a fallu ajouter les fonctionnalités lui permettant **l'enregistrement de un ou plusieurs QoS Agents**. Dorénavant, lorsque le QoS Server reçoit un message XML, il doit donc aussi pouvoir interpréter les messages de type REGISTER ou DEREGISTER ayant le format présenté sur la Figure 46. Ainsi, lorsqu'il reçoit un message de ce type, il enregistre les paramètres <AddressIPv4, PortToUse et AddressIPv6> dans la base de QoS Agent.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<XMLQoSMessage>
 <Sender>QoSAgent</Sender>
 <Type type="REGISTER" >
  <Parameters
            AddressIPv4="192.168.19.44"
            PortToUse="12001"
            AddressIPv6="2001:660:6602:102:218:8bff:fea7:579c" />
 </Type>
</XMLQoSMessage>
```

Figure 46 – Format XML d'un message REGISTER envoyé par le QoS Agent au QoS Server

Par la suite, à chaque fois qu'il reçoit une requête de réservation/libération en provenance d'une entité autre qu'un QoS Agent, il compare les adresses du flux concerné, comprises dans le message de réservation/libération, avec les adresses des QoS Agents enregistrés dans sa base. Si l'un d'entre eux correspond, il le prévient de la réservation/libération réalisée respectivement par un message RESV/FREE en indiquant les paramètres correspondant au flux. Ces derniers messages ont le même format que celui décrit par la Figure 47 (le paramètre *Sender* devenant « QoS Server »).

Ensuite, comme on l'a dit auparavant, sa principale fonctionnalité est de recevoir des messages RESV et FREE en provenance des Proxies SIP et des QoS Agents et de **configurer la QoS en conséquence**. La Figure 47 montre alors un exemple de message RESV envoyé par un QoS Agent. Il comprend différentes informations permettant d'identifier le flux et de connaître le service auquel le flux doit être associé ainsi que le débit (dans le cas de messages envoyés par le proxy SIP, on a en plus des informations concernant la gigue maximale, le délai maximal et le taux de perte maximal).

```
<?xml version = "1.0" encoding = "UTF-8"?>
<XMLQoSMessage>
 <Sender>QoSAgent</Sender>
 <Type type="RESV" >
  <Connection
            Command="iperf"
            PortSrc="33496"
            PortDst="5001"
            Service="1"
            Pid="13420"
            BitRate="50"
            IPSrc="2001:660:6602:102:218:8bff:fea7:579c"
            IPDst="2001:660:6602:103::14"
            IPVersion="IPv6"
            Protocol="UDP" />
 </Type>
</XMLQoSMessage>
```

Figure 47 – Format XML d'un message RESV envoyé par le QoS Agent au QoS Server

Pour la configuration de la QoS au niveau IP (et donc le bloc *IP QoS* de la plateforme d'émulation), l'outil TC (*Traffic Control*), fonctionnant sous Linux, a été choisi. Cet outil utilise essentiellement les éléments suivants :

- Un choix de gestionnaires de mise en file d'attente par flux réseau.
- Un choix de règles de classification des paquets avant leur mise en file d'attente.
- Un choix d'ordonnanceurs pour la mise en forme du trafic en sortie d'une interface.

Comme cela a déjà été expliqué dans le paragraphe IV.1.1, TC nous permet d'utiliser les classes HTB. De plus, TC nous permet d'ajouter ou de supprimer un gestionnaire de mise en file d'attente (*queuing discipline* ou *qdisc*) par l'intermédiaire de la commande « tc qdisc » et aussi de créer ou détruire des classes de trafic avec « tc class ». Ainsi, pour la mise en œuvre de la QoS au niveau de la couche IP, on utilise la hiérarchie décrite sur la Figure 48, en se basant sur 3 catégories DiffServ principales dont une, AF, avec trois sous catégories. De plus, chaque catégorie ou sous catégorie est associée à deux valeurs : *rate* (la bande passante garantie) et *ceil* (la bande passante maximale en « burst »). Enfin, on peut voir que chacune des 5 sous classes définies (EF, AF31, AF32, AF33 et BE) est associée à un champ DS spécifié au niveau des ovales comportant le terme *dsmark*.

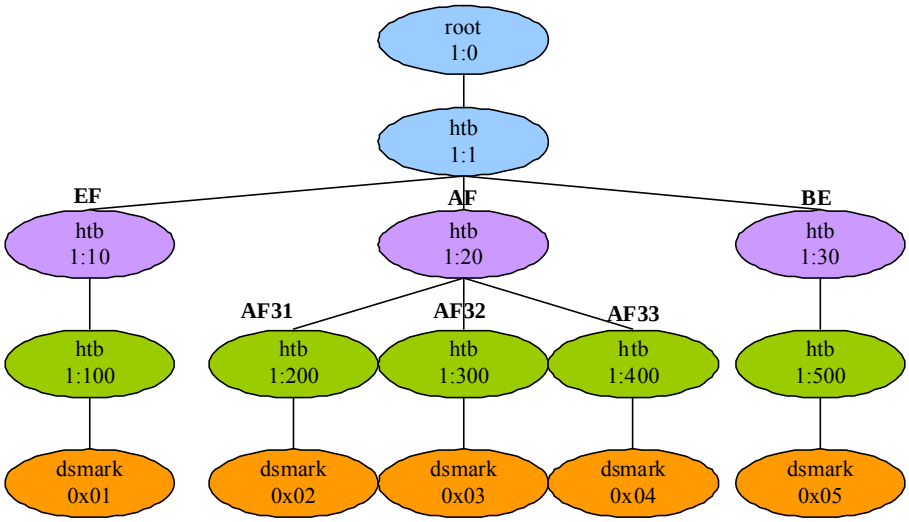


Figure 48 – Configuration des classes de service DiffServ par TC

Les différentes commandes TC nécessaires à la création et à la configuration de ces classes sont décrites précisément dans [169].

Lorsque le QoS Server reçoit une requête de réservation (voir Figure 47), il doit alors **modifier les paramètres des différentes files concernées et ajouter un nouveau filtre correspondant au flux** devant faire l'objet d'une réservation, par l'intermédiaire de la commande « tc filter ». Dans ce cas, le flux identifié par le quadruplet (< adresse_src | port_src ; adresse_dest | port_dest >), < 2001:660:6602:102:218:8bff:fea7:579c | 33496 ; 2001:660:6602:103::14 | 5001 >, doit être ajouté dans la file EF (Service="1" sur la Figure 47) et son débit est évalué à 50 kbit/s.

Pour implémenter cette fonctionnalité, nous avons donc modifié le QoS Server pour qu'il augmente la bande passante allouée à la file EF de 50kbit et qu'il diminue d'autant celle de la file BE, si cela est possible bien évidemment. Dans le cas contraire, la réservation (et donc les modifications TC) ne peut être effectuée.

Les commandes TC nécessaires aux modifications correspondant à une réservation sont détaillées dans [169].

Dans le cas d'une libération de ressource, le QoS Server agira de façon inverse en diminuant la bande passante allouée à la file concernée et en augmentant celle de la file BE.

Ce fonctionnement explique donc la nécessité d'ajouter au QoS Agent la possibilité de définir le débit de l'application pour laquelle on veut effectuer une réservation, car, sans cela, même si le flux est marqué pour passer dans la file EF et y passe effectivement, la reconfiguration de la taille des files ne pourra pas se faire et donc la qualité des flux de cette file ne sera plus assurée.

Cependant, il est clair que l'utilisateur n'est pas toujours capable d'évaluer le débit de ses applications. Donc un outil tel que le proxy SIP qui connaît les débits associés aux applications qu'il permet d'initier est bien évidemment plus adapté pour la mise en œuvre de la QoS dans un réseau, qu'il soit satellite ou non. D'ailleurs, nous utiliserons le QoS Agent uniquement pour reréserver ou libérer la QoS après un changement de réseau, en se basant sur

les informations liées aux réservations/libérations effectuées par le proxy SIP et transmises par le QoS Server.

IV.2.4.2. Le QoS Agent mobile

En ce qui concerne les modifications réalisées au niveau du QoS Agent, une partie d'entre elles a déjà été définie dans le paragraphe précédent (IV.2.4.1) concernant le QoS Server. Elles sont les suivantes :

- L'enregistrement/désenregistrement auprès du QoS Server qui implique la définition de nouveaux messages (REGISTER/DEREGISTER).
- La possibilité de définir le débit associé à une application pour laquelle on veut effectuer une réservation et la modification du message XML correspondant.

En plus de sa capacité d'envoyer des messages RESV/FREE au QoS Server, le QoS Agent doit aussi être capable de **recevoir ces mêmes messages de son QoS Server lorsque ce dernier reçoit des messages de réservation/libération qui proviennent d'une autre entité** que ce QoS Agent, mais qui concernent des applications tournant sur le même terminal.

Pour illustrer cette fonctionnalité, prenons un exemple concret : un utilisateur démarre sur son terminal le QoS Agent qui s'enregistre automatiquement auprès du QoS Server associé ; il décide ensuite de lancer une communication de VoIP initiée par SIP ; son client SIP s'enregistre auprès du proxy SIP associé et initie la session ; le proxy SIP intercepte les messages SIP, interprète les SDP, en déduit les codecs et leurs caractéristiques associées puis envoie un message RESV au QoS Server associé ; le QoS Server reçoit ce message et constate qu'il provient d'une entité autre qu'un QoS Agent ; il regarde donc sa base de données répertoriant les QoS Agents qui sont enregistrés auprès de lui et voit que le flux de VoIP concerne l'un de ses QoS Agent qu'il prévient donc aussitôt par un message RESV.

Le QoS Server envoie les messages RESV et FREE à destination de l'adresse et du port précisés dans le message REGISTER envoyé par le QoS Agent concerné. Chaque QoS Agent écoute donc sur le port qu'il a indiqué dans son message REGISTER. Dés qu'il reçoit un message, il modifie en conséquence sa table des réservations qui répertorie les flux avec QoS.

Une autre fonctionnalité que nous avons ajoutée au QoS Agent est la **découverte dynamique de son QoS Server associé** (et donc du ST associé). Pour cela, lors de sa connexion à un réseau, le terminal sur lequel tourne le QoS Agent va amorcer les mécanismes de DHCP pour IPv4 et d'autoconfiguration d'adresse pour IPv6. Ces échanges de messages lui permettent alors de connaître les adresses IPv4 et IPv6 du ST, et donc du QoS Server auprès duquel le QoS Agent peut donc directement s'enregistrer.

De plus, pour gérer l'enregistrement auprès du QoS Server courant ainsi que les réservations/libérations dans un environnement mobile, d'autres fonctionnalités doivent aussi être ajoutées au QoS Agent :

- Module de détection de changement de réseau qui sonde périodiquement les adresses IP courantes du client avec une période très faible pour que les réservations soient réalisées le plus tôt possible. Cependant, on pourrait aussi utiliser le principe cross-layer basé sur « debugfs », défini dans le paragraphe IV.2.3.2.a, pour faciliter cette détection.

- Lorsqu'un changement de réseau est détecté, le QoS agent doit :
  - Se réenregistrer auprès du QoS Server courant
  - Réserver les ressources pour les flux présents dans la table des réservations auprès du nouveau QoS Server en tenant compte du changement d'adresse.
  - Libérer les ressources pour les flux présents dans la table des réservations auprès de l'ancien QoS Server
  - Se désenregistrer auprès de l'ancien QoS Server.
  - Mettre à jour la table des réservations (changer l'adresse source des flux).

La Figure 49 résume le fonctionnement du QoS Agent mobile.

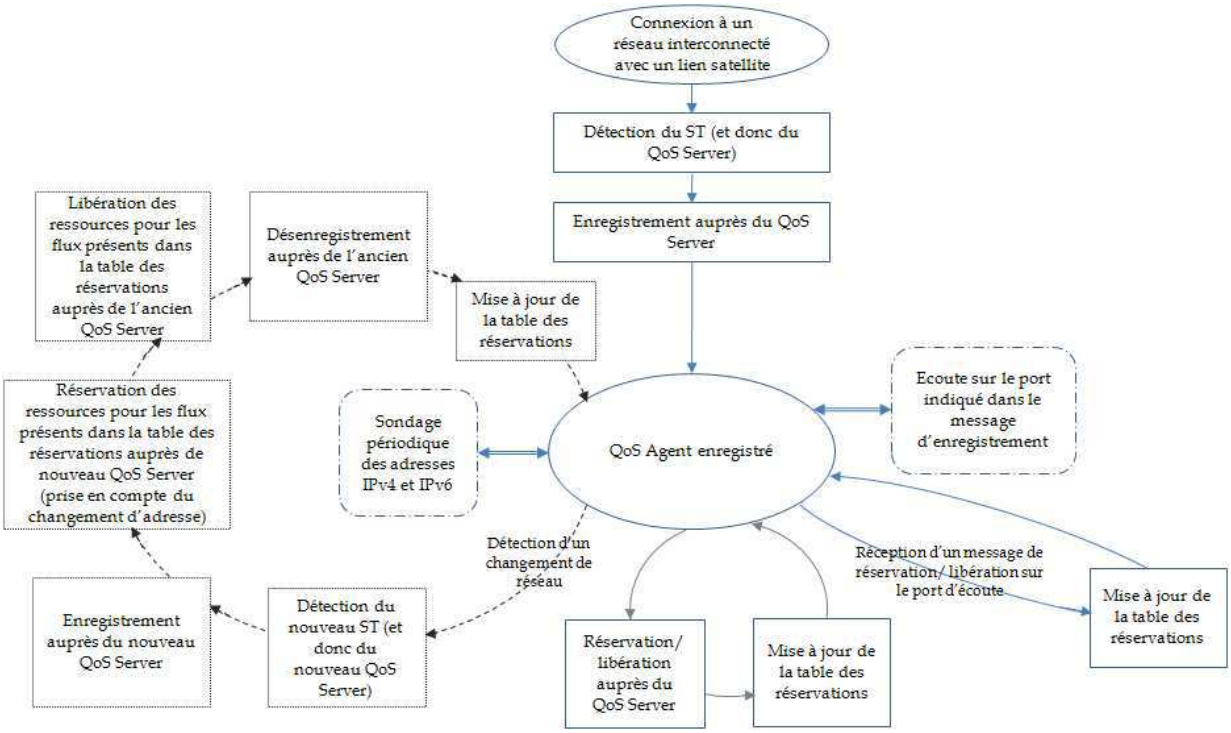


Figure 49 – Fonctionnement du QoS Agent mobile

IV.2.4.3. L'ARC

L'ARC, défini dans le paragraphe III.3.5.4 et localisé au niveau de l'entité GW/NCC de la Figure 41, a un fonctionnement quelque peu similaire à celui du QoS Server. En effet, il est aussi configuré pour écouter sur un port spécifique (5334 pour nos expériences) sur lequel il reçoit les messages XML de type RESV ou FREE (identiques à ceux envoyés au QoS Server), en provenance des proxies SIP. Il est de plus équipé d'une table contenant les identifiants (adresses IP) des abonnés ayant souscrit à ce service et étant donc autorisés à demander une augmentation de la quantité de CRA allouée à leur ST en plus de la reconfiguration des files DiffServ réalisée par le QoS Server associé. Si le ou les abonnés concernés par la communication sont autorisés à utiliser ce service, une augmentation de CRA est alors effectuée auprès du serveur DAMA en tenant compte des informations sur le débit contenues dans le message XML ainsi que des caractéristiques du ST (quantité de CRA déjà allouée et quantité de CRA maximale autorisée).

A l'inverse, lors d'une fin de session, le ou les proxies SIP concernés enverront un message FREE indiquant à l'ARC qu'il faut reconfigurer le serveur DAMA pour qu'il diminue la quantité de CRA allouée aux STs correspondants.

IV.2.4.4. Le JAIN-SIP-PROXY conscient de la QoS

Dans cette partie, nous allons décrire comment ont été implémentées les différentes fonctionnalités que nous avons ajoutées au proxy SIP et qui ont déjà été spécifiées dans le paragraphe III.3.5.3, concernant la QoS.

La première fonctionnalité à ajouter est d'extraire et d'analyser les descripteurs de session des messages SIP que le proxy reçoit. Ce dernier va donc tout d'abord extraire du message INVITE les premières informations concernant la session (Call-Id de la session et adresse de l'appelant). Ensuite, il extrait les autres informations (adresse de l'appelé et SDP), soit du message Session Progress, soit du message OK (INVITE).

Une fois qu'il a toutes les informations nécessaires concernant les types de média (audio, vidéo, …) et les codecs utilisés, il consulte sa base de données (ou table des codecs) concernant les codecs dont il connaît déjà les caractéristiques. Si les codecs sont connus, il commence par mettre à jour une table d'association permettant d'associer une session (Call-Id) avec les médias qui y sont impliqués, chaque média étant lui-même associé à une série de caractéristiques obtenues des SDP (IP_src, Port_src, IP_dest, Port_dest) ou de la table des codecs (débit maximum, gigue maximum, délai maximum et taux de perte maximum). Il peut alors envoyer au QoS Server et à l'ARC un message de réservation (ou plusieurs si différents médias sont impliqués dans la session) dont le format est le même que celui envoyé par le QoS Agent (voir Figure 47), avec les paramètres supplémentaires MaxJitter, MaxLoss et MaxDelay, ces derniers pouvant par exemple être utilisés au niveau du QoS Server pour décider dans quelle file les flux vont être redirigés en fonction des politiques adoptées.

Si les codecs sont inconnus localement, le proxy SIP fait appel au MTR que nous décrivons dans la partie suivante IV.2.4.5.

L'autre fonctionnalité importante que nous avons ajoutée au proxy SIP consiste à libérer les ressources associées à un utilisateur ayant changé de réseau lorsque le proxy reçoit un message REGISTER/DEREGISTER le concernant, en provenance d'un autre proxy SIP. Dans ce cas, la procédure d'analyse du message reçu permettant de connaître l'utilisateur concerné, déjà décrite dans le paragraphe IV.2.3.2.b, permet aussi de savoir si une session concernant cet utilisateur était en cours avant son changement de réseau et donc si elle doit faire l'objet de libération de ressource. Si c'est le cas, un message classique FREE sera envoyé au QoS Server concerné et à l'ARC.

Concernant la configuration de la QoS, nous avons enfin ajouté au niveau du proxy SIP une fonctionnalité qui lui permet d'associer un niveau de QoS à chaque type de média utilisé pour la session. Ainsi, pour l'instant, la configuration est statique (et réalisée par l'opérateur ou le fournisseur d'accès par exemple) et nous tenons compte uniquement de trois types de médias ; les médias « audio » et « vidéo » sont associés, comme on l'a déjà dit, à un service de type EF et une demande d'allocation CRA auprès de l'ARC tandis que le média désigné par « text » qui correspond à de la messagerie instantanée par exemple est associé à un service de type AF sans demande d'allocation CRA supplémentaire puisque les contraintes sur le

délai sont moins importantes que dans le cas d'une vidéoconférence. Cependant, cette configuration pourrait se faire à l'aide d'un autre service Web qui permettrait à un opérateur ou FAI de configurer les proxies SIP selon sa propre politique et selon le contrat de l'utilisateur.

IV.2.4.5. Le MTR : service Web pour l'amélioration de la QoS

Conformément aux spécifications du paragraphe III.3.5.5, nous avons développé un service Web pour permettre au proxy SIP d'obtenir les caractéristiques associées à un codec lorsqu'il ne les connaît pas localement (dans sa table des codecs). Pour mettre en œuvre cette architecture client/serveur orientée services, nous avons choisi de nous baser sur deux projets open source *d'Apache Software Foundation* : Apache Axis [165] qui est un package JAVA permettant d'implémenter les spécifications SOAP et fournissant un certain nombre d'outils permettant la création de Web Services ainsi que leurs déploiements et leurs invocations ; et Apache Tomcat [166] qui implémente les spécifications des technologies *Java Servlet* (application permettant la création dynamique de données au sein d'un serveur HTTP) et *JavaServer Pages* (JSP) (technique permettant de générer dynamiquement du code HTML, XML, etc…) et qui fait aussi office de serveur HTTP.

Le service Web mis en place, nommé QoS_Service, est alors composé de différentes méthodes qui ont été définies pour pouvoir gérer et utiliser ce service Web. Du point de vue gestion du service Web, pour permettre aux administrateurs de réseaux de s'adapter aux évolutions de la vidéo et de l'audio (et d'autres médias), deux méthodes ont été créées : la méthode *addMediaType* pour pouvoir ajouter de nouveaux codecs et la méthode *deleteMediaType* pour pouvoir en supprimer. Du point de vue utilisateur, une méthode *getQoSParameters* est fournie pour permettre au client d'obtenir les paramètres associés au codec demandé. C'est cette méthode qui est invoquée par le proxy SIP. Elle a pour entrées le type de média, le nom du codec demandé et la version d'IP utilisée et pour sorties, les 4 paramètres *peakBitRate* (en kbps), *maxJitter* (en ms), *maxLoss* (en %) et *maxDelay* (en ms).

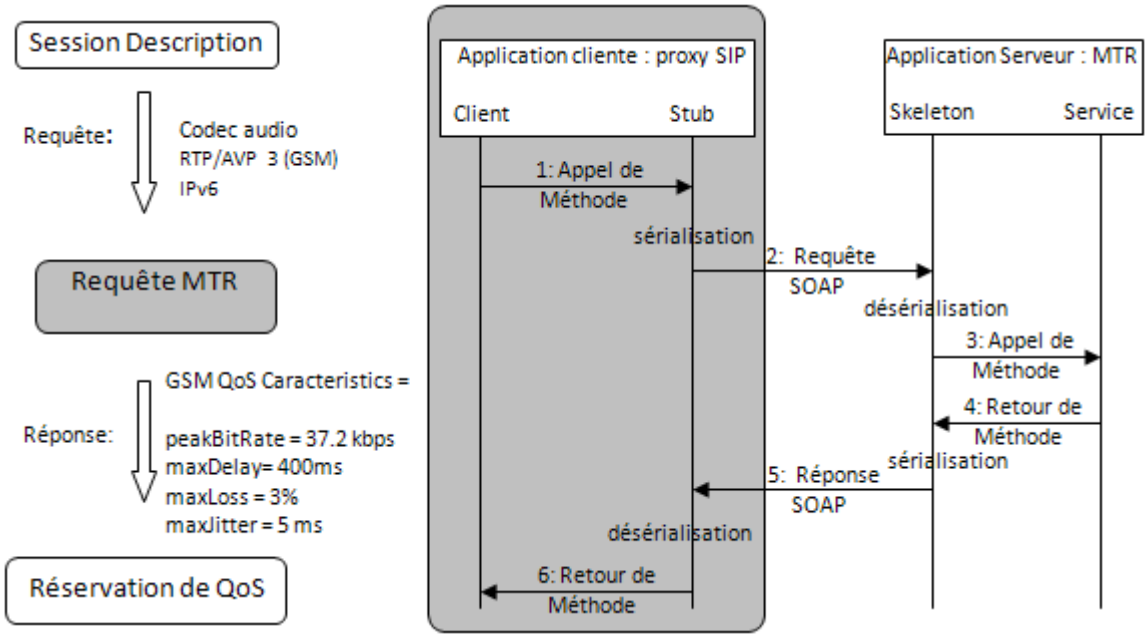


Figure 50 – Principe de fonctionnement du service Web MTR

La Figure 50 résume le principe de fonctionnement du service Web MTR que nous avons développé. Sur cette figure, seuls les messages SOAP sont indiqués sans tenir compte des messages échangés pour l'établissement et la terminaison de la connexion TCP (puisque les messages SOAP utilisent HTTP/TCP).

IV.3. Evaluation de l'architecture de QoS dans un système DVB-S2/RCS

Dans cette partie, nous allons expliciter les différentes évaluations de performance de l'architecture de QoS que nous avons effectuées sur la plate-forme PLATINE, essentiellement dans le cadre du projet SATSIX. Toutes les expériences présentées ici ont été réalisées en IPv6 en considérant un système satellite maillé avec satellite régénératif.

De plus, pour la mise en place de session de visioconférence avec QoS, nous utilisons notre architecture SIP basée sur le client VisioSIP et le proxy JAIN-SIP-PROXY conscient de la QoS. Pour cela, un codec audio (de type GSM mais avec une charge utile de 99 octets plutôt que 33) et un codec vidéo (de type H263) sont choisis, avec respectivement 15.1 kbps et entre 80 et 85 kbps de débit de données utiles moyen. Si on reprend le schéma de la Figure 41, on considère des communications entre l'UT 11 et l'UT 21 qui sont tous deux équipés du client VisioSIP, un proxy SIP conscient de la QoS étant aussi présent derrière chaque ST au niveau des UTs 12 et 22. Enfin, un QoS Server se situe au niveau de chacun des STs 1 et 2 concernés par ces communications.

Les courbes obtenues sont issues de l'utilisation des outils FLOC, JTG et ORENETA qui ont permis de capturer, de générer et de rejouer exactement les mêmes flux lorsque cela était nécessaire afin de ne tenir compte pour nos analyses que des changements dans les conditions expérimentales et non pas des changements dans les caractéristiques des flux. Cependant, pour chaque évaluation, l'expérience est réalisée au moins une fois avec l'architecture SIP+QoS Server+ARC que nous avons développé ainsi que les vrais flux issus d'une session de vidéoconférence (qui nous sert souvent à effectuer la capture pour la suite de l'évaluation). Dans le cas de rejeu, nous nous contentons d'initier la session avec les messages SIP pour que les proxies puissent interpréter les SDP et envoyer un message XML au QoS Server et à l'ARC, lorsque cela est nécessaire, pour que ces derniers puissent configurer respectivement les files DiffServ au niveau des STs et le serveur DAMA au niveau de la GW/NCC.

IV.3.1. Impact de la gestion des files d'attente : EF vs BE

IV.3.1.1. Perturbations UDP

Pour étudier l'impact de la gestion des files d'attente sur la voie retour d'un ST, nous allons analyser le comportement de celle du ST1 (voir Figure 41). Ce ST est configuré pour avoir une capacité totale de 1000 kbps dont 150 kbps sont alloués statiquement (CRA) et 850 kbps peuvent être alloués en RBDC. Nous avons configuré le CRA à 150 kbps car cela correspond à peu près à la somme des débits d'émission des flux audio (21,6 kbps) et vidéo

(130 kbps) en tenant compte des en-têtes RTP, UDP et IPv6, sachant que pour le flux vidéo, nous tenons compte du débit crête.

Pour cette expérience, nous démarrons à t=10 s une session de visioconférence initiée par SIP, puis à t=60 s, t=120 s et t=180 s, nous ajoutons à chaque fois un flux UDP concurrent de 500 kbps. A t=250 s, les flux UDP sont stoppés et enfin à t=300 ms la visioconférence se termine. Tous ses flux sont envoyés depuis l'UT 11 en direction de l'UT 21 (les flux audio et vidéo sont aussi échangés dans l'autre sens mais, le fonctionnement étant le même, nous nous concentrons sur ce sens de la communication) et passent donc par le ST1 qui les transmet au ST2. Le but de cette évaluation est de réaliser cette expérience avec et sans mécanisme de QoS SIP pour valider le fonctionnement de notre architecture de QoS.

Nous allons pour cela analyser les graphiques de la Figure 51 correspondants à la courbe du délai moyen (*Moving average delay*) du flux vidéo dans le cas sans QoS (courbe A, tous les flux sont en BE) et avec QoS (courbe B, les flux audio et vidéo sont en EF et les autres en BE). La même analyse pourrait être faite pour le flux audio.

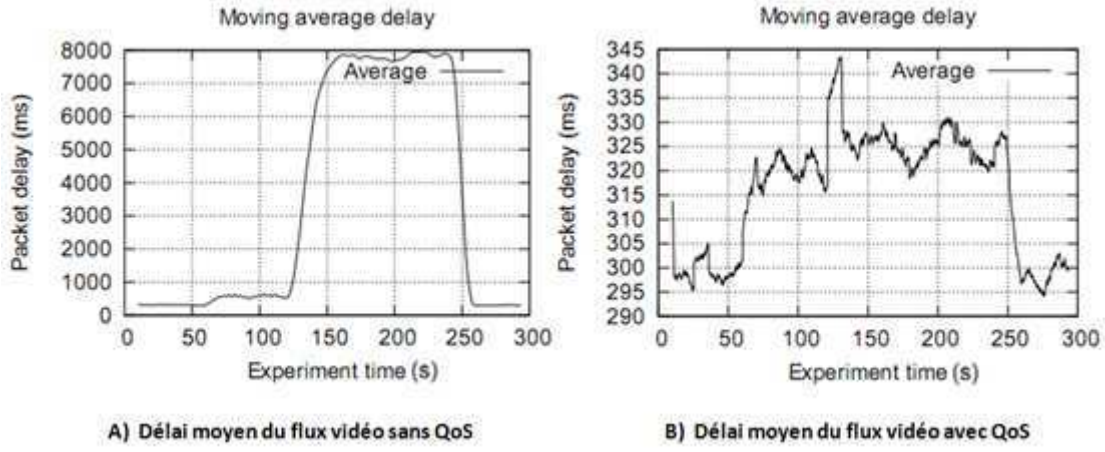


Figure 51 – Impact des mécanismes de QoS initiés par SIP sur le délai moyen

On constate que jusqu'à t=60 s, lorsqu'aucun flux UDP concurrent n'est présent, le délai moyen du flux vidéo est environ égal à 300 ms dans les deux cas. Ceci s'explique par le fait que toutes les ressources CRA sont utilisées par les flux multimédias et donc qu'aucune capacité à la demande (RBDC) n'est nécessaire.

A t=60 s, lorsqu'un premier flux UDP concurrent intervient, on remarque que le délai moyen du flux vidéo augmente dans les deux cas, mais tandis qu'il n'augmente que très légèrement dans le cas où la QoS est activée (il passe de 300 ms à 320 ms), il monte jusqu'à environ 500 ms dans le cas sans QoS. Bien que la capacité du canal soit suffisante pour les trois flux (audio, vidéo, 1 flux UDP concurrent), les ressources CRA ne suffisent plus et des requêtes RBDC sont alors nécessaires. Le délai du flux vidéo (de même que celui du flux audio) augmente alors beaucoup plus sensiblement dans le cas sans QoS puisque tous les flux utilisent la même file MAC et donc le même PVC (*Permanent Virtual Circuit*). En effet, le schéma d'allocation de capacité implique que tous les flux observent la même augmentation de délai. Au contraire, lorsque la QoS est activée, à l'initiation de la session, le proxy SIP a envoyé un message de réservation pour les flux audio et vidéo au QoS Server qui reconfigure alors les files à l'aide de TC ; les paquets correspondants aux flux audio et vidéo passent alors

par la file EF et utilisent une autre file MAC et un autre PVC, prioritaires par rapport à ceux de la file BE. Il en découle que les flux audio et vidéo sont protégés et que leur délai moyen n'augmente que très faiblement. On peut d'ailleurs constater que dans le cas sans QoS, les délais ressentis par les flux audio et vidéo sont déjà supérieurs aux recommandations de l'ITU-T (< 400 ms) alors que la capacité totale du lien n'est même pas atteinte. Ceci indique l'importance de la mise en place de mécanismes de QoS au sein des systèmes satellites.

A t=120 s, un deuxième flux UDP concurrent vient s'ajouter au premier. La somme des débits des flux en cours dépasse alors la capacité du lien, ce qui se traduit, dans le cas sans QoS, par une augmentation très importante du délai moyen du flux audio (et des autres flux) qui atteint jusqu'à 8000 ms (la capacité totale de la file est alors atteinte). Dans le cas avec QoS, on peut voir que les flux audio et vidéo restent protégés malgré une légère augmentation du délai à l'arrivée de la perturbation. De plus, leur délai moyen reste compatible avec les recommandations de l'ITU-T.

Les mêmes commentaires peuvent être faits pour la phase pendant laquelle un troisième flux UDP concurrent intervient. En effet, le délai observé par le flux vidéo reste inférieur à 360 ms dans le cas avec QoS. Ceci indique que, même lorsque la capacité du lien est largement surchargée, la séparation en classe de service permet une protection efficace des flux prioritaires. Cette première expérience montre donc clairement l'apport que représentent les mécanismes de QoS mis en place au travers de notre architecture SIP + QoS Server.

Nous avons ensuite réalisé la même expérience en configurant les flux vidéo et audio en EF et les flux de perturbations en AF. Dans ce cas là, les flux audio et vidéo sont aussi protégés.

IV.3.1.2. Perturbations TCP

Pour cette expérience, nous avons considéré les mêmes conditions expérimentales sauf que les flux concurrents sont des flux TCP au lieu d'UDP.

On peut alors remarquer sur la Figure 52.A qu'une fois de plus, la vidéo (ainsi que l'audio) est bien protégée par le service EF. Le délai moyen reste compris entre 300 et 330 ms pendant toute la session avec de légères augmentations ressenties à l'arrivée des flux de perturbations TCP (à t=60 s, t=120 s et t=180 s).

La Figure 52.B permet de constater que le délai moyen ressentie par le 1er flux TCP reste raisonnable tout au long de l'expérience tandis que les Figure 52.C et D permettent de voir que la bande passante est partagée de façon équitable entre les différents flux TCP. En effet, lorsque les 3 flux TCP sont simultanément en cours entre t=180 s et t=240 s (ce qui correspond à la période entre t=60 s et t=120 s sur la Figure 52.C et à la période entre t=0 et t=60 s pour la Figure 52.D), on peut constater que les 2ème et 3ème flux TCP ont tous deux un débit moyen aux alentours de 250-275 kbps (de même que le 1er flux iperf). La somme des débits des 3 flux correspond donc bien à peu de choses près à la bande passante disponible pour la file BE, obtenue en soustrayant les 150 kbps de la vidéoconférence à la bande passante totale (1000 kbps).

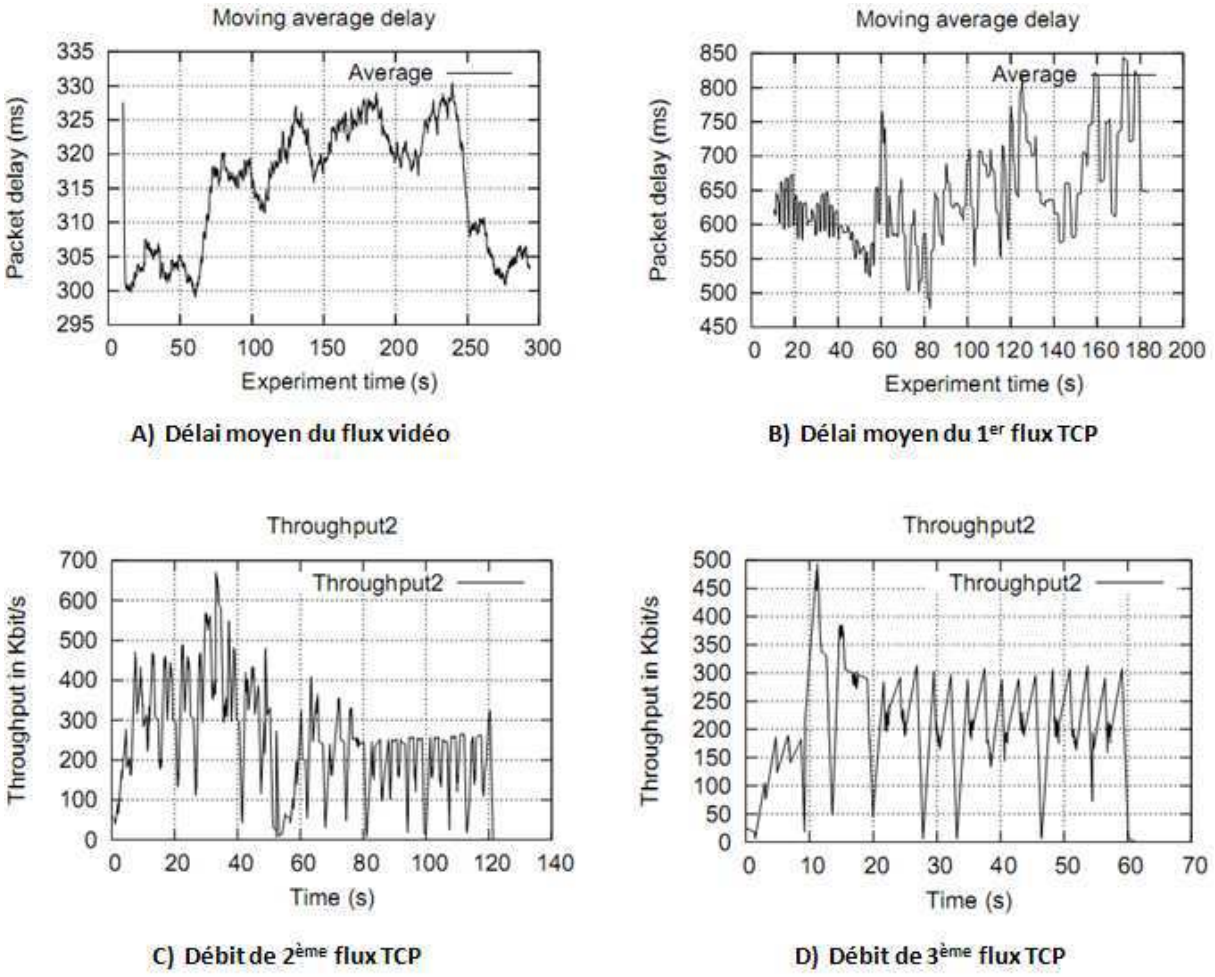


Figure 52 – Vidéoconférence en EF puis perturbations TCP

IV.3.2. Impact des mécanismes liés aux requêtes RBDC

Pour cette expérimentation, nous voulons tester l'impact de l'algorithme du DAMA sur une application de vidéoconférence. Le protocole expérimental est le même que pour la partie avec QoS du paragraphe précédent (mêmes codecs pour la vidéoconférence, les flux audio et vidéo sont marqués pour être dirigés dans la file EF et les flux UDP concurrents restent en Best Effort et interviennent aux mêmes moments) mais cette fois, aucun CRA n'est initialement alloué au ST1 et toute la capacité (1000 kbps) peut être obtenue par l'intermédiaire des requêtes RBDC. De plus, dans le cadre du projet SATSIX, nous avons choisi d'utiliser l'algorithme DAMA de l'ESA.

Sur les Figures 53.A et 53.C, on peut constater que le délai moyen des flux vidéo et audio varie entre 600 et 700 ms avant que le 1er flux concurrent UDP n'intervienne à t=60 s. A ce moment là, le délai moyen diminue pour se stabiliser entre 400 et 450 ms et on remarque de nouveau une diminution lorsque le 2ème flux UDP concurrent intervient : le délai moyen des flux vidéo et audio se stabilise alors à environ 325 ms. Finalement, lorsque les flux concurrents se terminent, le délai moyen revient à sa valeur initiale entre 600 et 700 ms. On peut donc constater en premier lieu que l'algorithme du DAMA de l'ESA fonctionne correctement mais que les requêtes RBDC dues aux seuls flux audio et vidéo ne sont pas suffisantes pour que ceux-ci répondent aux recommandations sur le délai de l'ITU-T (<400

ms). Le 2ème élément intéressant de cette expérience réside dans le fait que le délai diminue lorsque des flux concurrents interviennent. En fait, cela s'explique par le fait que, l'application de vidéoconférence étant considérée comme un trafic prioritaire, elle prend avantage des requêtes RBDC faite pour les flux UDP concurrents et les utilise pour ses propres flux.

Sur la Figure 53.B, le délai moyen reste inférieur à 1 s tant que la capacité du lien n'est pas atteinte. Mais, lorsque c'est le cas, à partir de t=60 s (ce qui correspond à t=120 s sur les Figures 53.A et 53.C), le délai augmente de façon importante pour les flux de la classe BE.

En conclusion de cette expérience, on peut constater que les requêtes RBDC, même si elles permettent d'obtenir d'assez bons résultats, ne sont pas suffisantes pour assurer une qualité de service adaptée aux flux audio et vidéo puisque, dans le cas où le trafic de fond n'est pas élevé, leur délai est trop important (> 400 ms) pour satisfaire les exigences de QoS de ce type d'application. Lorsque les ressources CRA initialement allouées au ST concerné ne suffisent pas, il est donc nécessaire de les modifier dynamiquement lors de la mise en place de la session SIP. Nous allons voir dans le paragraphe suivant les mécanismes permettant cette modification dynamique du CRA.

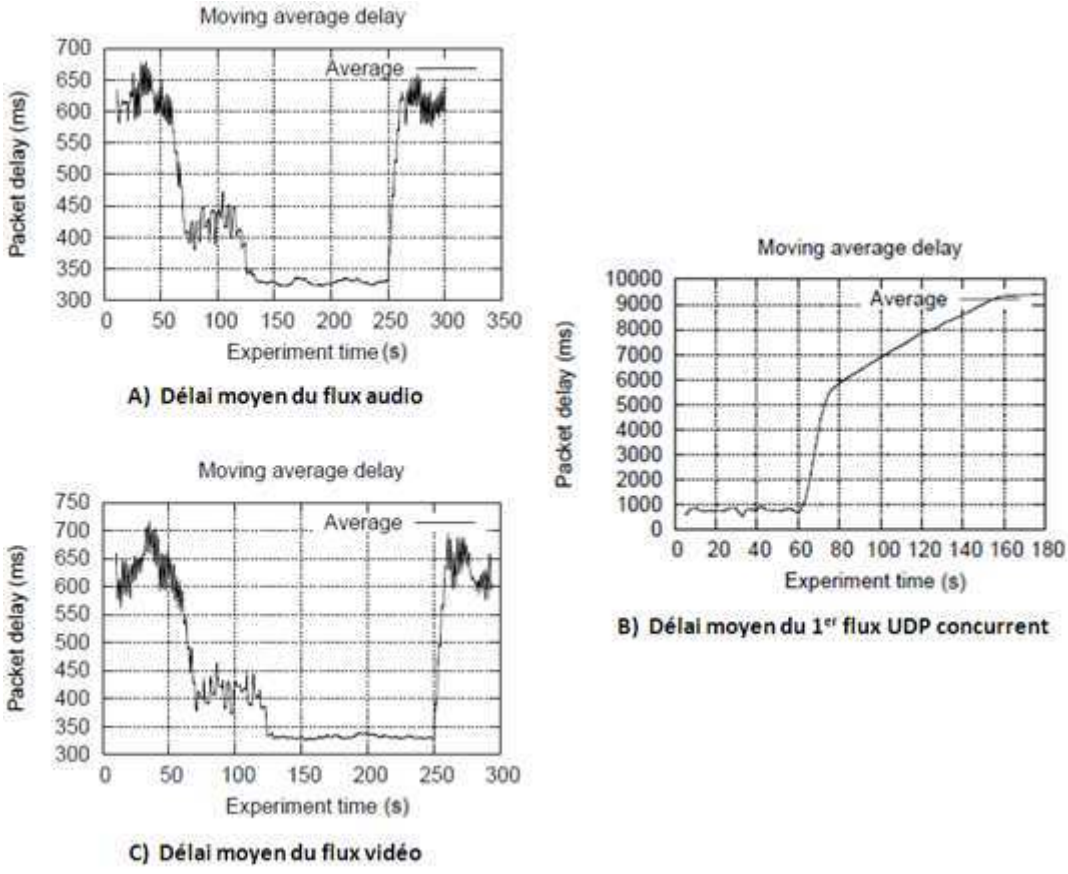


Figure 53 – Impact de l'algorithme du DAMA sur le délai moyen

IV.3.3. Modification dynamique des ressources CRA

Pour cette expérience, on considère le même protocole expérimental que pour le paragraphe précédent : les mêmes flux démarrent aux mêmes moments et aucun CRA n'est initialement alloué au ST. Cependant, nous avons ajouté à notre architecture la possibilité

d'interagir avec l'ARC (voir paragraphe III.3.5.4) situé au niveau du NCC et nous cherchons à étudier l'impact qu'il peut avoir sur le délai.

Lors de l'initiation de session SIP, le proxy SIP analyse les SDP contenus dans les différents messages et détermine les types de médias impliqués dans la session et leur QoS associée (voir paragraphe IV.2.4.4). Pour nos expériences, nous considérons que, dans le cas de communication audio ou vidéo, l'augmentation de la quantité de CRA allouée est nécessaire en plus de l'association au service EF. Le proxy SIP envoie alors simultanément au QoS Server ainsi qu'à l'ARC un message XML classique RESV. L'ARC peut alors prévenir le serveur DAMA qui va ainsi augmenter la quantité de CRA allouée au(x) ST(s) concerné(s) par la communication pendant que le QoS Server va reconfigurer les files DiffServ directement au niveau des STs.

On obtient alors la courbe du délai moyen du flux vidéo, représentée par la Figure 54 qui, comme on peut le constater est quasiment identique à celle de la Figure 51.B ce qui indique que l'augmentation de CRA allouée a bien eu lieu parallèlement à la configuration des files DiffServ. En effet, le délai moyen reste toujours inférieur à 340 ms contrairement au cas précédent (voir Figure 53.C). De plus, lorsque le message BYE est échangé, le proxy SIP envoie bien un message XML FREE au QoS Server ainsi qu'à l'ARC.

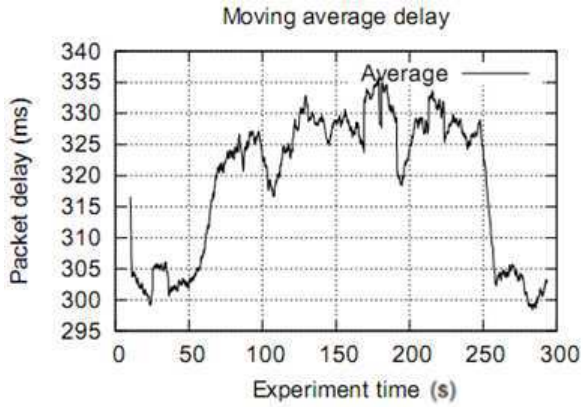


Figure 54 – Impact de l'ARC sur le délai moyen

IV.3.4. Configuration de la file EF pour codec à débit variable

Lors des expériences précédentes, pour la configuration des files TC, nous avons considéré le débit IP du flux audio (21.6 kbps) puisqu'un codec à débit constant est utilisé mais nous avons utilisé le débit IP crête estimé du flux vidéo (130 kbps) qui utilise un codec à débit variable (débit IP entre 60 kbps et 130 kbps, avec un débit IP moyen entre 85 et 90 kbps).

Cependant, même si, effectivement, se baser sur le débit moyen impliquerait des délais de transmission de paquets trop importants particulièrement lorsque le lien satellite se trouverait surchargé, se baser uniquement sur le débit crête serait aussi extrêmement pénalisant en termes d'efficacité globale du système puisque, la plupart du temps, le débit réel serait bien moins important.

Pour cette expérience, nous considérons donc que, dans le cas d'un flux vidéo, le débit moyen ET le débit crête estimés correspondants au codec utilisé sont obtenus depuis la table de codecs locale ou le MTR, le but étant alors d'étudier comment la quantité de CRA allouée

à un ST donné et la taille des files DiffServ peuvent être configurées de façon optimale à partir des débits moyen et crête, dans le cas d'une vidéo utilisant le codec H 263 (issue d'une session de vidéoconférence).

Pour cela, nous étudions le taux de paquets dont le délai est inférieur à 400 ms en fonction d'une part, du débit considéré (qui varie entre le débit moyen et le débit crête) pour la reconfiguration des files DiffServ et la quantité de CRA requise auprès de l'ARC et, d'autre part, de la charge du ST considéré. Les résultats sont résumés sur la Figure 55 et montrent que le choix du débit considéré est important par rapport à la charge du lien satellite correspondant au ST concerné qui n'influence que légérement le délai de transmission des paquets.

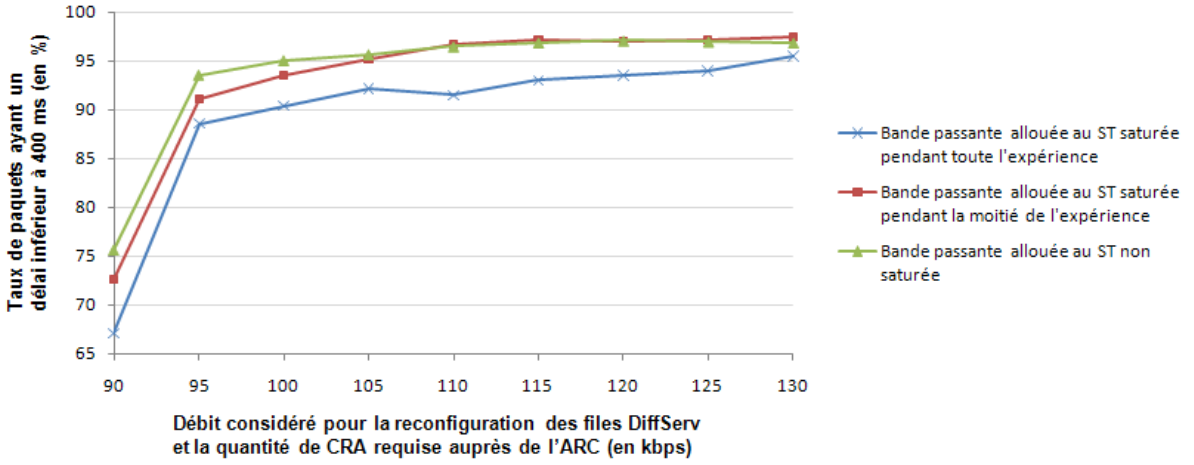


Figure 55 – Evolution du taux de paquets ayant un délai inférieur à 400 ms en fonction du débit considéré et de la charge du ST

En effet, on voit que, si la file EF et la quantité de CRA sont configurées en tenant compte du débit moyen (90 kbps), si on tient compte du pire des cas (lien surchargé en permanence), on aura 67 % des paquets seulement qui respecteront les recommandations de l'ITU-T. Par contre, dès qu'on les configure en tenant compte d'une valeur un peu supérieure au débit moyen (95 kbps), on voit que cela permet d'augmenter de façon non négligeable le nombre de paquets conformes (88,6 % dans le pire des cas). Ensuite, le taux augmente progressivement avec l'augmentation du débit considéré pour la quantité de CRA et la reconfiguration de la file EF. On peut d'ailleurs noter que le taux de paquets dont le délai est inférieur à 400 ms semble saturer aux alentours de 97,5 %, ce qui semble anormal mais en regardant de plus près les résultats obtenus, on peut repérer que le délai des premiers paquets correspondants au flux vidéo est supérieur à 400 ms pendant les premiers instants de l'expérience, certainement à cause d'un problème de configuration de la plateforme d'émulation satellite. Si l'on ajoute ces 2,5% à chacune des valeurs du graphique, pour s'assurer dans tous les cas que par exemple 95% des paquets ont un délai inférieur à 400 ms, il faudra configurer la file à 115 kbps, ce qui correspond à environ **0,6*(débit crête – débit moyen) + débit moyen** ou encore à **0,6*débit crête + 0,4*débit moyen**.

Nos expériences indiquent aussi qu'en cas de sous-estimation du débit considéré pour un flux vidéo, la qualité ressentie au niveau de l'utilisateur en sera grandement pénalisée. Par

exemple, dans le cas du codec H263, configurer la file EF et la quantité de CRA avec une valeur de 80 kbps fait chuter le taux de paquets dont le délai est inférieur à 400 ms à 0,07 %.

Jusqu'à présent, nous avons considéré que les valeurs prises pour l'allocation de CRA et la configuration de la taille de la file EF (ainsi que la diminution de la taille de la file BE) étaient les mêmes. Mais, dans ce cas là, la file EF ne peut pas utiliser de requêtes RBDC en plus de l'allocation CRA. Nous allons donc maintenant étudier le cas où la quantité de CRA est fixe mais où la taille de la file EF peut être supérieure et donc utiliser des requêtes RBDC si nécessaire. Le Tableau 11 présente alors l'évolution du taux de paquets dont le délai est supérieur à 400 ms en fonction de la taille de la file EF en considérant que 90K de CRA sont alloués.

Tableau 11 – Etude de l'influence des requêtes RBDC sur le taux de paquets dont le délai est inférieur à 400 ms

	CRA =90K et taille de la file EF=90K	CRA =90K et taille de la file EF=100K	CRA=90K et taille de la file EF=110K	CRA =90K et taille de la file EF=120K	CRA=90K et taille de la file EF=130K	CRA=90K et taille de la file EF=150K	CRA=90K et taille de la file EF=170K
Taux de paquets dont le délai est inférieur à 400 ms (en %)	75.6	80.1	84,3	85.5	86,7	87,3	87,8

Dans le cas d'une allocation CRA de 90K, l'utilisation de requêtes RBDC n'est donc pas suffisante pour atteindre des taux supérieurs à 90%, même si l'on configure la taille de la file EF avec une valeur supérieure au débit crête de l'application, mais elle permet tout de même une amélioration significative du taux de paquets conformes aux recommandations de l'ITU-T.

Finalement, la quantité de CRA allouée au ST et la configuration de la taille de la file EF doivent donc dépendre de la politique de l'opérateur et des engagements qu'il a pris vis à vis de ses clients. Ainsi dans le cas d'une vidéoconférence, l'opérateur peut proposer différents types d'abonnements variant par exemple entre le service de faible qualité offrant une quantité de CRA (et une reconfiguration des files DiffServ) correspondante au débit moyen estimé du codec utilisé (indiqué dans la base du MTR) et le service le plus cher offrant une quantité de CRA correspondante au débit crête estimé du codec utilisé (indiqué dans la base du MTR), avec entre les deux, un service offrant, par exemple, un débit égal à 0,6*débit crête + 0,4*débit moyen et un autre offrant une quantité de CRA égal au débit moyen mais permettant en plus l'utilisation par la file EF de requêtes RBDC (la quantité étant aussi fixée par l'opérateur).

Ensuite, l'opérateur doit globalement définir quels types de services doivent être associés à une allocation CRA (nous avons par exemple considéré que les applications audio et vidéo nécessitent du CRA tandis que les applications de type messagerie instantanée n'en nécessite pas) et quelle est la politique adoptée en cas de congestion si une nouvelle demande d'allocation CRA intervient (refus du nouveau flux, acceptation du flux mais en RBDC, acceptation du flux et renégociation d'autres flux, etc…). Dans tous les cas, ces politiques d'acceptation d'admission de connexion peuvent être mise en œuvre dans le cadre de notre architecture de QoS basée sur des proxies SIP améliorés.

IV.3.5. Priorisation de la signalisation SIP

Un autre problème peut intervenir lorsque, contrairement à nos expériences précédentes, une session de vidéoconférence doit être mise en place alors que le lien satellite concerné est déjà saturé par d'autres trafics concurrents. En effet, dans ce cas là, rien ne permet d'assurer que les messages SIP ne seront pas perdus et donc que la session pourra effectivement avoir lieu. Deux solutions sont alors possibles pour mettre en œuvre une politique de QoS au niveau IP et MAC permettant un ordonnancement prioritaire de ce type de flux :

- Soit on considère que ce type de flux passe par la file EF.
- Soit on met en place une nouvelle classe de service de niveau 3 pour traiter exclusivement ce type de flux en y associant un débit garanti et une priorité élevée (supérieure ou équivalente à celle d'EF selon la politique de l'opérateur) que l'on mappe sur une nouvelle file MAC. Cela correspondrait aux files LNM et INM de niveau IP et à la file NM de niveau MAC définies dans le paragraphe III.3.3.

Pour illustrer ce problème, nous avons réalisé une expérience pour évaluer le temps nécessaire pour l'initiation d'une session SIP d'une part et pour la ré-inititation d'une session SIP d'autre part en comparant à chaque fois le cas où la signalisation SIP est priorisée avec celui où elle ne l'est pas. Pour toutes ces expériences, la session SIP se fait entre deux clients situés chacun derrière un ST distinct et on considère que l'un des deux STs est surchargé. Le temps considéré pour l'initiation d'une session SIP correspond au temps nécessaire pour que les clients SIP échangent les messages INVITE, Session Progress, PRACK, OK(PRACK), UPDATE et OK (UPDATE), ce temps ne prenant donc pas en compte les derniers messages échangés (OK(INVITE) et ACK) puisque ceux-ci dépendent du moment où l'appelé répond à l'appel. Enfin, le temps considéré pour la ré-initiation de session SIP correspond au temps nécessaire pour que les clients SIP échangent les messages re-INVITE et OK. Pour chaque expérience, une série de 20 mesures est réalisée et les résultats obtenus sont résumés sur le Tableau 12.

Tableau 12 – Temps d'établissement nécessaire pour une initiation ou une ré-inititation de session SIP en fonction de la priorisation de la signalisation SIP

	Temps d'établissement moyen	Temps d'établissement minimum	Temps d'établissement maximum
Initiation de session SIP avec signalisation SIP priorisée	2.18 s	2.04 s	2.34 s
Initiation de session SIP avec signalisation SIP non priorisée	38.17 s	19.94 s	65.53 s
Ré-initiation de session SIP avec signalisation SIP priorisée	0.75 s	0.65 s	0.88 s
Ré-initiation de session SIP avec signalisation SIP non priorisée	10.48 s	3.61 s	25.72 s

On peut constater que la priorisation de la signalisation SIP influence considérablement les temps d'établissements, que ce soit dans le cas d'une initiation ou d'une ré-initiation de

session SIP. Elle permet effectivement d'améliorer significativement les temps d'établissement moyens et de les rendre très peu variables. Au contraire, lorsque la signalisation n'est pas priorisée et passe donc par la même file que les autres flux au niveau du ST surchargé, le temps est non seulement fortement allongé en moyenne mais surtout il est très dépendant du délai à l'intérieur de la file ainsi que des pertes éventuelles de messages SIP, ce qui explique les grandes différences obtenues entre les valeurs minimales et les valeurs maximales

La priorisation de la signalisation SIP est donc primordiale, particulièrement dans le cas d'une ré-initiation de session SIP après qu'un client SIP mobile est changé de réseau.

IV.3.6. Evaluation de notre service Web : le Media Type Repository

Pour déployer notre architecture basée sur le service Web MTR dans le système satellite, deux solutions sont possibles :

- La solution centralisée dans laquelle le MTR est localisé au niveau de l'entité GW/NCC de la Figure 41. L'avantage essentiel de cette architecture est qu'une seule base de données est alors nécessaire pour offrir les informations de QoS à tous les proxies SIP du système satellite. Il est donc plus facile d'y effectuer des mises à jour, modifications ou suppressions. En contre partie, les requêtes et réponses échangées entre le MTR et un proxy SIP localisé derrière un ST (autre que la GW/NCC) subiront le délai dû au lien satellite.
- La solution distribuée dans laquelle un MTR est localisé au niveau de chaque ST ou GW. Les mises à jour, les modifications et les suppressions deviennent un peu plus complexes à gérer, mais les requêtes et réponses SOAP sont échangées localement ce qui permet un gain de temps important.

Pour nos évaluations, nous comparons le temps de réponse obtenu pour les deux solutions précédentes en considérant qu'une session SIP est mise en place entre deux clients localisés derrière deux STs différents. Les messages SIP sont interceptés par les proxies qui envoient leur requête auprès du MTR après réception du message Session Progress (mais cela pourrait être fait de la même manière après le OK(INVITE)). Plus de détails sont fournis dans [167].

Pour chaque scénario, l'expérience est réalisée 25 fois et les résultats obtenus sont présentés sur le Tableau 13. On constate que dans le cas centralisé, on obtient un temps moyen de 1270 ms qui correspond bien à peu de choses près à la somme du temps d'établissement de la connexion TCP (les 3 messages du three-way handshake) et de l'échange requête-réponse SOAP, c'est-à-dire un peu plus de 4*300 ms (4 bonds satellites) puisque le dernier message du three-way handshake et la requête SOAP sont envoyés quasi simultanément. Le temps de réponse est donc particulièrement long dans le cas centralisé. En effet, si l'on considère que les deux proxies SIP utilisés pour la mise en place de la session sont en mode « Assured » (voir paragraphe II.3.3), le temps de mise en place de la session va être allongé de 2,5 s environ ; et même dans le mode « Enabled », en considérant que seul les messages INVITE/OK et ACK sont échangés, la session pourrait débuter avant même que les informations de QoS aient été reçues par le proxie SIP. Enfin, si l'on considère le cas d'un utilisateur SIP mobile, ce délai supplémentaire devient très pénalisant en termes de temps d'interruption et une architecture MTR distribuée s'avère indispensable.

En effet, le temps de réponse moyen est seulement de 17,75 ms en mode distribué ce qui convient beaucoup mieux à notre architecture SIP couplant mobilité et QoS et plus généralement, à toute architecture nécessitant une optimisation du temps de mise en place d'une session SIP. Par contre, le volume de données nécessaires pour la mise à jour sera plus important puisqu'il faudra transmettre les données à tous les MTRs au lieu d'un seul mais les mises à jour étant peu fréquentes, cela ne risque pas de diminuer de façon notable les performances du système satellite.

Tableau 13 – Temps de réponse entre un proxy SIP et le MTR

	Temps de réponse moyen	Temps de réponse minimum	Temps de réponse maximum
Scénario centralisé	1270 ms	1229 ms	1335 ms
Scénario distribué	17.75 ms	12 ms	29 ms

Il reste cependant la possibilité d'utiliser UDP pour transporter les messages SOAP et ainsi supprimer le temps nécessaire à l'établissement de la connexion TCP dans le cas centralisée.

IV.4. Evaluation des solutions de mobilité dans un système DVB-S2/RCS

Dans cette partie, nous allons présenter les différentes évaluations de performance que nous avons effectuées sur la plate-forme PLATINE, pour pouvoir comparer les différentes solutions de mobilité que nous avons prises en compte dans notre étude, à savoir, Mobile IPv6, la mobilité SIP et partiellement FMIPv6 (pour les raisons indiquées dans le paragraphe IV.2.3.1). Cette comparaison porte essentiellement sur les temps d'interruption obtenus pour différents cas de mobilité intervenant dans le cadre d'un système satellite DVB-S2/RCS, en considérant un utilisateur mobile en cours de session de vidéoconférence. Cependant, nous étudierons aussi les problèmes d'overhead liés à chaque solution.

IV.4.1. Comparaison des temps d'interruption

Pour comparer les différents temps d'interruption, nous étudions la même topologie et les mêmes types de déplacement que ceux qui ont été évalués théoriquement dans la partie III.4.2, en supprimant toutefois le déplacement 2 puisque nous n'avons pas évalué expérimentalement HMIPv6 dans le cadre de ce mémoire. En ce qui concerne la technologie de niveau 2 utilisée (à savoir le Wi-Fi), pour diminuer au maximum le temps de réassociation avec le nouveau point d'accès, nous forçons la carte à travailler sur un canal prédéfini pour chaque point d'accès, ce qui permet d'obtenir des temps d'interruption de niveau 2 situés essentiellement entre 0,1 et 0,2s. Pour pouvoir comparer les différentes solutions de la manière la plus juste possible, nous ne prendrons d'ailleurs en compte pour nos résultats que les temps d'interruption pour lesquels le temps de niveau 2 est situé dans cette fourchette. De même, pour calculer la moyenne des temps d'interruption obtenus pour chaque solution et

chaque déplacement, dans un premier temps, nous tenons uniquement compte des cas les plus courants où les échanges de messages de mobilité ont lieu comme défini dans le paragraphe III.4.2. Nous nous intéresserons ensuite aux cas particuliers pour lesquels les temps d'interruption sont plus élevés ou plus faibles que pour les cas les plus courants pour en expliquer les raisons.

IV.4.1.1. Cas courants

Dans le paragraphe III.4.2, nous nous intéressions plus spécifiquement au temps d'interruption dans le cas où le MN est récepteur alors qu'ici, puisque nous étudions le cas d'une communication bidirectionnelle de vidéoconférence, nous allons aussi nous intéresser au temps d'interruption dans le cas où le MN est émetteur, c'est-à-dire le temps écoulé entre le dernier paquet émis depuis l'ancien réseau (et reçu par le correspondant) et le premier paquet émis depuis le nouveau réseau (et reçu par le correspondant). En effet, ces deux temps peuvent différer selon le type de déplacement et le type de solution prise en compte. De plus, en ce qui concerne la phase d'optimisation de route de Mobile IPv6, nous ajoutons aussi le cas où le BACK n'est pas nécessaire entre le MN et le CN, puisque cette option est autorisée dans la pile MIPL2.0.2. Cela permet au MN de recommencer à émettre des paquets directement au CN jusqu'à 600 ms plus tôt dans les cas où le MN et le CN sont situés derrière un ST différent.

Pour les temps d'interruption correspondants à FMIPv6, bien que nous n'ayons pas pu observer la réception et l'envoi de message à travers le tunnel PAR-NAR, nous avons tout de même pu observer l'échange de tous les messages FMIPv6. En tenant compte des observations faites sur Mobile IPv6, nous pouvons donc raisonnablement considérer que le premier message émis depuis le nouveau réseau est envoyé juste après l'envoi du message UNA (on considérera + 0,05s après) et que le premier message reçu l'est 0,1 s après l'envoi de l'UNA (pour établir ces temps, on se base sur d'autres expériences similaires).

Les Figures 56 et 57 nous présentent alors les temps d'interruption moyens (20 mesures pour chaque déplacement et chaque solution) correspondants aux différentes solutions suivantes en étudiant les cas avec DAD ou avec Optimistic DAD (en activant l'option CONFIG_IPV6_OPTIMISTIC_DAD) et en ne tenant compte que des temps à partir desquels le délai de transmission des paquets entre le CN et le MN est inférieur à 400 ms :

- Mobilité SIP avec ACK (le CN envoie des données juste après avoir reçu le ACK).
- Mobilité SIP sans ACK (le CN envoie des données juste après avoir envoyé le OK (INVITE)).
- Mobile IPv6 avec BACK (le MN envoie des données directement au CN juste après avoir reçu le BACK envoyé par le CN).
- Mobile IPv6 sans BACK (le MN envoie des données directement au CN juste après avoir envoyé le BU à ce dernier).
- FMIPv6 en mode prédictif en considérant, comme pour Mobile IPv6, les cas avec ou sans BACK échangé avec le CN.

Toutefois, dans le cas de Mobile IPv6 et de FMIPv6, le temps d'interruption ressenti par le MN en tant que récepteur est le même qu'il y ait envoi de BACK ou non puisque le CN commence dans tous les cas à envoyer des paquets lorsqu'il reçoit le BU envoyé par le MN.

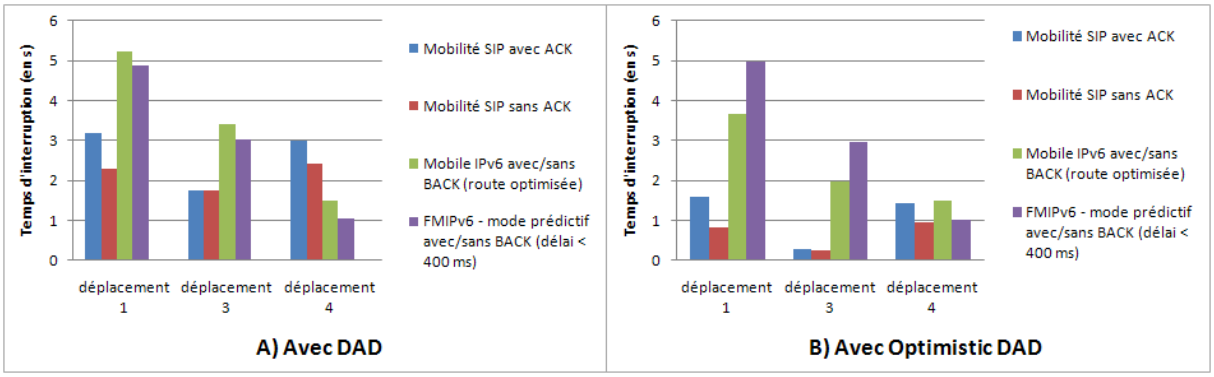


Figure 56 – Temps d'interruption ressenti par le MN en tant que récepteur

Par rapport aux temps calculés théoriquement dans le paragraphe III.4.2, on peut noter les différences importantes suivantes :

- Dans le cas du déplacement 1 pour Mobile IPv6 et FMIPv6, l'échange de BU/BACK entre le MN et le HA prend une seconde de plus que prévu, certainement à cause d'un bug de la pile MIPL 2.0.2.
- Dans le cas de FMIPv6, l'envoi du message UNA prend en moyenne 1,2 s mais varie entre un minimum de 0,26 s et un maximum de 2,47 s, ce qui allonge considérablement le temps d'interruption et surtout le rend très variable. Dans le cas où la méthode d'Optimistic DAD est utilisée, FMIPv6 devient alors la solution avec les temps d'interruption les plus longs pour les déplacements 1 et 2.

Comparons maintenant les solutions pour chaque déplacement :

- Pour le déplacement 1, même en tenant compte des défauts d'implémentation précédemment décrits, les solutions basées sur SIP sont globalement les plus efficaces en réception ainsi qu'en émission, surtout lorsque les mécanismes d'Optimistic DAD sont utilisés. A titre d'indication, dans le cas de FMIPv6, le message UNA est envoyé en moyenne à t=1,35 s pour les mesures effectuées pour ce déplacement. De même, pour la solution Mobile IPv6 avec DAD, le premier message est envoyé par le tunnel bidirectionnel (uniquement valable pour ce déplacement pour les raisons indiquées dans le paragraphe III.4.2.1) à t=3,36 s en moyenne (mais ce temps souffre aussi du défaut d'implémentation).
- Pour le déplacement 3, les solutions SIP sont aussi les plus efficaces.
- Pour le déplacement 4, les solutions SIP sont moins efficaces lorsque le DAD est activé puisque les autres solutions n'ont pas à effectuer ces mécanismes lorsque le MN rejoint son réseau mère, mais cette lacune est comblée lorsque les mécanismes d'Optimistic DAD sont utilisés.

Concernant Mobile IPv6 et FMIPv6, on peut effectivement constater que l'option désactivant l'usage du BACK dans la phase d'optimisation de route permet d'envoyer des paquets directement au CN jusqu'à 600 ms plus tôt. Cette option nous permet aussi de découvrir la phase communément appelée « routage triangulaire » pendant laquelle le MN

envoie des paquets directement au CN tandis que le CN continue à émettre par l'intermédiaire du HA. Cette phase intervient entre le moment où le MN envoie le BU au CN et le moment où le CN reçoit ce BU.

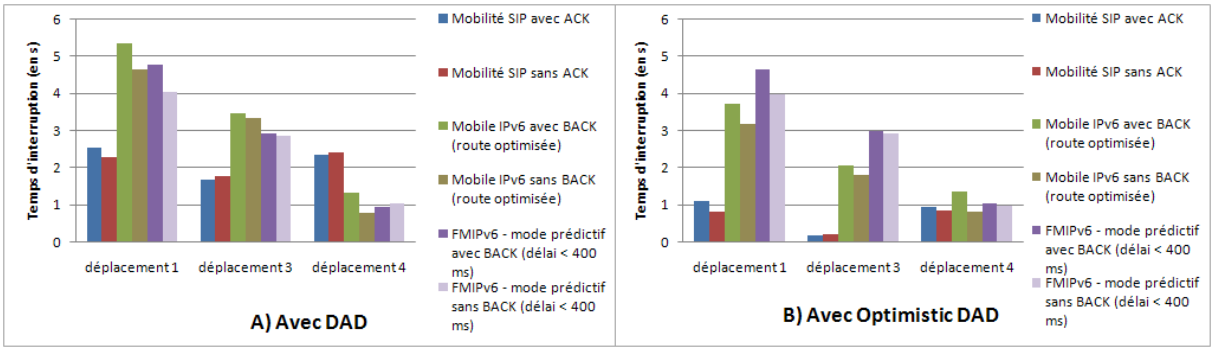


Figure 57 – Temps d'interruption ressenti par le MN en tant qu'émetteur

Plus généralement, si l'on considère un utilisateur mobile en cours de communication VoIP ou vidéconférence, dans le cadre d'un système satellite DVB-S2/RCS, on peut conclure que :

- Pour Mobile IPv6 avec RO et RRT, la phase de tunnel bidirectionnel (uniquement utilisable lorsque le MN quitte son réseau mère) ne permet pas d'être conforme aux recommandations de l'ITU-T. Le temps d'interruption est donc fortement allongé dès lors que les messages de Mobile IPv6 doivent traverser le système satellite.
- Pour FMIPv6, le tunnel PAR-NAR ne permet d'être conforme aux recommandations de l'ITU-T que dans le cas où le MN quitte le réseau du CN vers un autre réseau quelconque. Les procédures de RO et RRT spécifiques à Mobile IPv6 sont donc là aussi nécessaires et donc pénalisantes en terme de temps d'interruption.
- Les solutions basées sur SIP sont plus efficaces dans le cas général, sauf pour un retour dans le réseau mère où les mécanismes de DAD, lorsqu'ils sont activés, les pénalisent. Mais ce problème peut être supprimé par l'utilisation de mécanismes tels que l'Optimistic DAD. En effet, la communication peut reprendre directement entre le CN et le MN tout en limitant le nombre de messages devant traverser le système satellite.

IV.4.1.2. Cas particuliers

Au cours de nos expériences, nous avons tout de même pu constater des comportements particuliers dans certaines situations, surtout au niveau de la procédure de RRT liée à Mobile IPv6. Ainsi, dans le cas du déplacement 3, il est arrivé que seuls les messages CoTi/CoT soient échangés avec le CN (réduisant ainsi de environ 1,2 s le temps d'interruption). De même, dans le cas du déplacement 4 où seul l'échange de HoTi/HoT est habituellement nécessaire, il est arrivé qu'aucun message de la procédure de RRT ne soit échangé (réduisant ainsi de 0,6 s le temps d'interruption).

La RFC 3775 permet d'expliquer ces comportements spécifiques en indiquant que, dans le cas où un MN se déplace rapidement et régulièrement entre différents réseaux, il peut

parfois réutiliser une *keygen* encore valide. Ceci s'avère alors particulièrement efficace dans le cas du déplacement 3 par exemple, puisque, du coup, aucun message de la procédure de RRT n'est échangé à travers le système satellite.

Cependant, nous ne pouvons pas considèrer ces cas particuliers comme des possibilités d'améliorations puisque cela reviendrait à remettre en cause les mécanismes de RRT.

IV.4.2. Problème d'overhead

Un autre avantage de la mobilité SIP est qu'elle n'ajoute pas d'overhead aux paquets échangés entre le MN et le CN lorsque le MN est dans un réseau visité, contrairement aux solutions basées sur Mobile IPv6. En effet, Mobile IPv6, lorsque le MN communique directement avec le CN depuis un réseau visité, ajoute un champ supplémentaire de 24 octets à l'entête de chaque paquet IPv6 pour indiquer la HoA du MN et ainsi rendre transparent la mobilité du MN : le champ « Destination Option Header » dans le sens MN vers CN et le champ « Routing Header » dans le sens CN vers MN.

L'entête IPv6 passe donc de 40 à 64 octets, ce qui peut s'avérer pénalisant particulièrement pour une communication de VoIP qui utilise généralement des paquets UDP de petite taille. Si l'on prend l'exemple d'un codec GSM qui a pour caractéristique d'envoyer toutes les 20 ms un paquet dont la charge utile est de 33 octets (ce qui donne un débit utile de 13,2 kbps), le débit au niveau IPv6 (en tenant compte des entêtes RTP, UDP et IPv6) passe de 37,2 kbps à 46,8 kbps ce qui constitue une augmentation d'environ 26% (trois flux audio utilisant l'entête mobile IPv6 sonts équivalent à quatre flux sans entête Mobile IPv6).

Cet overhead est encore plus important pendant la phase de tunnel bidirectionnel puisqu'on a alors une encapsulation IPv6/IPv6. Si l'on reprend l'exemple du codec GSM, le débit IP passe alors à 53,2 kbps et on obtient le même résultat pour toutes les phases où un tunnel est mis en place comme pour FMIPv6, PMIPv6 ou HMIPv6 (pour cette dernière solution, dans le cas d'un changement de domaine MAP, on a même une phase avec double encapsulation IPv6/IPv6/IPv6). Cela implique donc une consommation des ressources plus importantes qui peut se révéler pénalisante, particulièrement pour le lien Wi-Fi ou le lien satellite.

Ce problème d'overhead confirme le fait que dans un système satellite où les ressources sont limitées, l'utilisation de l'optimisation de route est indispensable non seulement en termes de délai de transmission de paquet mais aussi en termes de consommation de bande passante. Mais dans tous les cas, une solution basée sur SIP est moins coûteuse en utilisation de ressources pour une quantité de données utiles équivalente puiqu'elle n'ajoute aucun overhead supplémentaire dans les paquets de données, et même si la taille des messages SIP est plus importante que celle des messages issus de Mobile IPv6 et de ses extensions [170], cela reste négligeable par rapport à la taille des paquets issus des flux audio et vidéo.

IV.4.3. Conclusion sur la mobilité

Ces différentes comparaisons montrent que les solutions basées sur Mobile IPv6, même si elles présentent l'incontestable avantage de pouvoir gérer la mobilité de façon transparente

pour **toutes** les applications, sont globalement moins efficaces que les solutions basées sur SIP en ce qui concerne la gestion de la mobilité des applications interactives de type vidéoconférence (mais aussi VoIP). Les expériences réalisées sur la plateforme d'émulation satellite sont donc concordantes avec les évaluations théoriques, les conclusions et les recommandations faites dans la partie III.4.2. En effet, l'efficacité des solutions basées sur Mobile IPv6 en termes de temps d'interruption est réduite à des cas particuliers de mobilité (essentiellement le retour au réseau mère et la micro-mobilité, non étudiée expérimentalement) tandis que la mobilité SIP reste plus constante dans la gestion globale des différents types de déplacements.

Ceci est encore plus vrai dans le cas d'un réseau à ressources limitées, comme c'est le cas pour un système satellite, puisque les solutions basées sur Mobile IPv6 ajoutent un overhead qui peut s'avérer important dès lors que le MN n'est plus dans son réseau mère.

IV.5. Evaluations des architectures couplant QoS et mobilité

Dans cette partie, nous allons présenter les différentes expérimentations qui nous ont permis de valider le fonctionnement des architectures présentées dans le paragraphe III.5 permettant la gestion de la mobilité et de la QoS dans un système DVB-S2/RCS. Les expériences ont aussi été réalisées sur la plateforme PLATINE en déployant une topologie maillée (avec satellite régénératif). De plus, nous considérons aussi pour chaque expérience qu'une session SIP de vidéoconférence (vidéo : de type H263 + audio : de type GSM) a lieu entre le MN et un CN et pour vérifier que ces flux sont correctement priorisés, on les soumet à des flux de perturbation qui viennent saturer le lien satellite. Si on reprend le schéma de la Figure 41, les éléments suivants sont communs aux 3 architectures que nous allons évaluer par la suite:

- Une communication SIP entre l'UT 13 et l'UT 33 qui sont tous deux équipés du client VisioSIP. L'UT 13 joue le rôle du MN et se déplace entre différents réseaux IEEE802.11 (WLAN 1, 2 et 3 sur la Figure 41), tous distincts les uns des autres (SSID différente) et donc un changement d'adresse IPv6 est nécessaire.
- Un proxy SIP conscient de la QoS (voir IV.2.4.4), présent derrière chaque ST/GW au niveau des UTs 12, 22, 32 et 4.
- Un QoS Server (voir IV.2.4.1) se situe au niveau de chacun des STs/GW du système satellite.
- Un ARC (voir IV.2.4.3) se situe au niveau de la GW/NCC.
- Un MTR (voir IV.2.4.5) se situe derrière chaque ST/GW au niveau des UTs 11, 21 et 31.

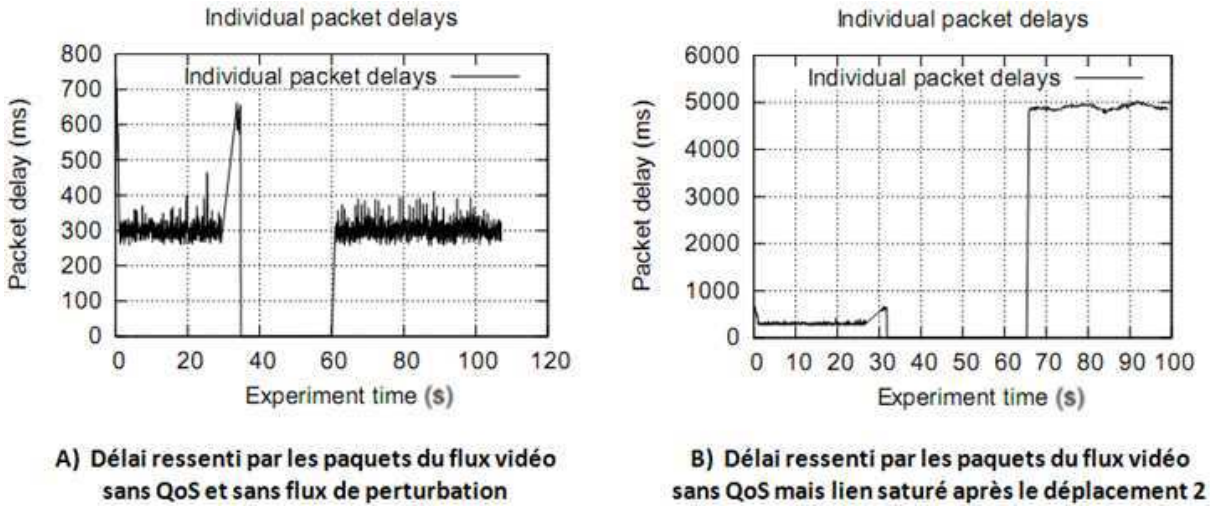
IV.5.1. Mobile IPv6 couplé au QoS Agent mobile

Dans le cas de mobile IPv6, en ce qui concerne la configuration des entités de la plateforme, en plus des éléments généraux présentés au début de la partie IV.5, nous considérons que les UTs 13 et 33 sont équipés de la pile MIPL2.0.2 (voir IV.2.3.1) ainsi que d'un QoS Agent mobile (voir IV.2.4.2). De plus, l'UT 12 joue le rôle de HA et est donc aussi équipé de la pile MIPL2.0.2.

Pour nos expériences, nous considérons le cas où une session SIP est mise en place entre l'UT13 et l'UT 33. Dans le cas où les mécanismes de QoS sont utilisés, la QoS est donc configurée au niveau du ST1 et de la GW par l'intermédiaire des proxies SIP, des QoS Server - qui préviennent ensuite les QoS Agents mobiles des réservations effectuées. On considère ici que la quantité de CRA allouée au niveau de chaque ST est suffisante pour les flux audio et vidéo. En cours de communication, l'UT 13 se déplace tout d'abord vers le réseau de son correspondant (déplacement 1) puis retourne dans son réseau initial (déplacement 2). Nous allons alors valider notre architecture en la comparant à une solution basée sur Mobile IPv6 mais sans QoS. Pour cela, nous analysons différents cas : tout d'abord, nous réalisons un test sans QoS Agent mobile et sans que le lien satellite soit saturé ; ensuite, nous réalisons un test sans QoS Agent mobile mais en considérant que le lien satellite est saturé lorsque le MN (UT13) revient dans son réseau initial (déplacement 2) ; enfin, nous reprenons ce dernier test mais cette fois ci en utilisant notre architecture de QoS.

La Figure 58.A permet d'observer les différentes étapes du scénario décrit précédemment :

- Entre t=0 et t=29,2 s, le MN et le CN communiquent à travers le système satellite : le délai est bien environ égal à 300 ms, puisque la quantité de CRA allouée est suffisante pour les 2 flux interactifs (vidéo + audio) et qu'aucun flux concurrent n'est présent.
- A t=29,2 s, le MN se déplace dans le réseau de son correspondant. On a alors une période d'interruption de 4,3 s pendant laquelle le MN doit s'associer à un nouveau point d'accès Wi-Fi, s'auto-configurer une nouvelle adresse IPv6 (sa CoA), réaliser les mécanismes du DAD et échanger les messages BU/BACK avec son HA.
- A partir de t=33,5 s, on peut alors observer une phase rapide (un peu moins de 200 ms) où le délai est environ égal à 600 ms. Cela correspond à la phase de tunnel bidirectionnel de Mobile IPv6 où tous les paquets échangés entre le MN et le CN passent par le HA, bien qu'ils soient localisés dans le même réseau et ce, jusqu'à ce que le MN ait réalisé la procédure de RRT puis échangé directement avec le CN les messages BU/BACK. Le temps d'interrupion total jusqu'à la phase de route optimisée est alors de 5,6 s.
- Entre t=34,8 s et t=60 s, le CN et le MN communiquent en route optimisée, sans passer par le HA. Le délai est alors faible, de l'ordre de quelques millisecondes.
- A t=60 s, le MN se déplace de nouveau pour retourner dans son réseau d'origine. On observe ici un temps d'interruption environ égal à 0,99 s.
- A partir de t=61 s, le délai est de nouveau proche de 300 ms puisque la communication repasse par le satellite.



A) Délai ressenti par les paquets du flux vidéo sans QoS et sans flux de perturbation

B) Délai ressenti par les paquets du flux vidéo sans QoS mais lien saturé après le déplacement 2

Figure 58 – Mobile IPv6 sans QoS Agent mobile

Ensuite, la Figure 58.B. présente la même expérience mais cette fois, le lien satellite est saturé lorsque le MN revient dans son réseau mère, après le déplacement 2. Les premières étapes de l'expérience sont donc les mêmes : d'abord la phase où le délai est d'environ 300 ms avant le 1er déplacement ; ensuite une période d'interruption d'environ 3,7 s, une rapide phase de tunnel bidirectionnel puis une phase de route optimisée. La différence se situe au niveau du 2ème déplacement après la période d'interruption, ici égale à 0,69 s du point de vue du MN mais sachant que le CN ne reçoit les premiers paquets qu'environ 5 s après qu'ils aient été envoyés. On constate en effet qu'à partir de t=65,9 s, la communication reprend mais le délai ressenti par les paquets est très élevé, proche de 5 s. On retrouve alors les résultats obtenus dans le paragraphe IV.3.1.1. Ces valeurs élevées du délai sont dûes au fait que des flux concurrents sont intervenus pendant le 1er déplacement et ont saturé la bande passante allouée au ST considéré et, puisqu'aucun mécanisme de QoS n'est utilisé, les flux audio et vidéo partagent les mêmes files IP et MAC et le même PVC que les flux concurrents et subissent donc tous une forte augmentation du délai.

Pour résoudre ce problème, nous réalisons maintenant la même expérience en considérant que le lien satellite MN→CN (ou HA→CN) est congestionné dès le début et jusqu'à la fin de l'expérience mais cette fois-ci, les mécanismes de QoS que nous avons développé sont utilisés. La Figure 59.A présente tout d'abord le délai de transmission des paquets du flux vidéo, mais pointe deux problèmes que nous allons détailler : le problème de la libération des ressources et le problème de transmission des messages MobileIPv6 en milieu saturé.

Dans ce cas, la session SIP est mise en place avec QoS, c'est-à-dire que les proxies SIP procèdent à la réservation de QoS auprès des QoS Server concernés. Les QoS Server notent que le message RESV provient d'un proxy SIP et donc préviennent les QoS Agent mobiles concernés (ceux du MN et du CN) qui enregistrent cette réservation. La session SIP commence donc et le délai des communications reste proche de 300 ms jusqu'à t=23 s où le MN se déplace dans le réseau de son correspondant.

On observe alors un temps d'interruption plus important qu'habituellement (environ 11,5 s) avant une phase de tunnel bidirectionnel anormalement longue (14,5 s) et pendant laquelle le délai de transmission y est très élevé (environ 5 s). Ceci s'explique pour deux raisons :

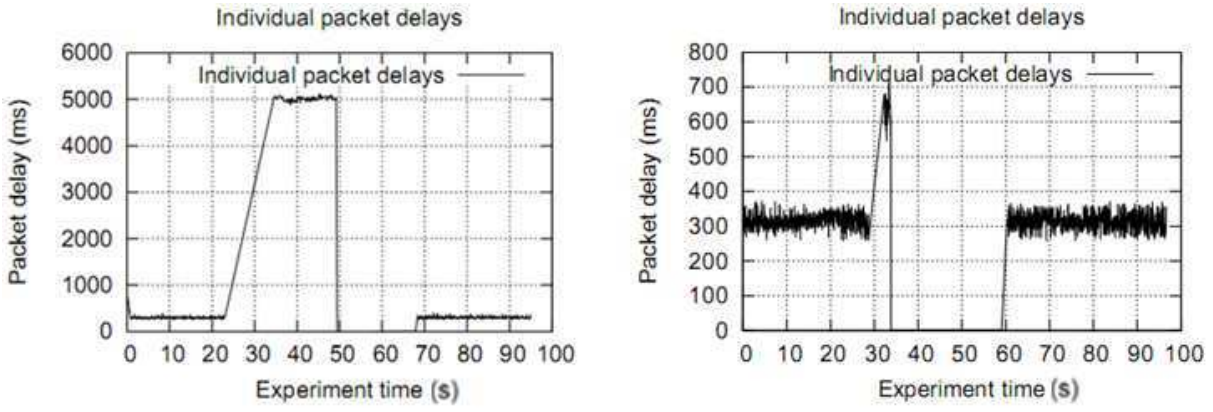
- D'une part, les messages Mobile IPv6 ne sont pas priorisés et donc en cas de congestion, ils vont subir le même délai que le reste du trafic et risquent aussi d'être perdus. La phase de tunnel bidirectionnel met donc beaucoup plus de temps à se mettre en place. De même, le message HoTi doit aussi traverser le lien congestionné, ce qui a pour conséquence de retarder considérablement l'optimisation de route.
- D'autre part, après que le MN se soit déplacé, le QoS Agent mobile détecte le changement de réseau : il doit alors libérer les ressources dans l'ancien réseau, ce qu'il fait aussitôt, mais, puisque l'optimisation de route n'a pas encore été réalisée, cela a pour conséquence d'augmenter considérablement le délai de transmission des paquets du flux vidéo qui subissent alors le même délai que le reste du trafic.

Le MN doit parallèlement se réenregistrer auprès de son nouveau QoS Server (celui de la GW) avant d'envoyer les messages de réservation auprès de ce dernier. Cependant, en analysant les messages RESV, le QoS Server voit que les flux concernés ne traversent pas le système satellite puisque le MN et le CN sont dans le même réseau : aucune réservation n'est donc réalisée.

Par contre, après le déplacement 2, la QoS peut être reconfigurée normalement au niveau du ST considéré et le trafic reprend avec un délai de 300 ms malgré la congestion du lien.

Cette expérience permet de comprendre l'importance de la **priorisation des messages de mobilité** puisque, dans le cas contraire, le temps d'interruption est fortement allongé.

En ce qui concerne le problème de libération de ressources, il s'agit en fait de permettre au QoS Agent mobile de savoir quand il doit libérer les ressources au niveau de l'ancien ST. Un mécanisme de timer se basant sur le temps moyen nécessaire pour réaliser l'optimisation de route pourrait faire l'affaire, mais il serait beaucoup plus efficace que la pile MIPL2.0.2 prévienne le QoS Agent mobile dès que l'optimisation de route a été réalisée.



A) Délai de transmission de paquets du flux vidéo avec perturbations et QoS pour flux audio et vidéo mais libération des ressources trop rapide

B) Délai de transmission de paquets du flux vidéo avec perturbations et QoS pour flux audio, vidéo et messages Mobile IPv6

Figure 59 – Mobile IPv6 avec QoS Agent mobile

La Figure 59.B nous montre alors la même expérience en tenant compte des remarques précédentes : les messages Mobile IPv6 sont priorisés de la même manière que les messages SIP (voir paragraphe IV.3.5) et la libération des ressources au niveau de l'ancien ST est faite après l'optimisation de route (par un mécanisme de timer).

Quelques remarques importantes peuvent être faites concernant cette architecture couplant Mobile IPv6 au QoS Agent mobile :

- Il faut permettre la communication entre la pile MIPL2.0.2 et le QoS Agent mobile pour que le MN puisse savoir précisement quand libérer les ressources allouées au niveau de l'ancien ST.
- Après un déplacement du MN, la libération de ressources doit aussi être faite dans le sens CN→MN ainsi qu'éventuellement la re-réservation en mettant à jour l'adresse du MN. Pour cela, la pile MIPL2.0.2 au niveau du CN doit aussi prévenir son QoS Agent mobile lorsqu'elle reçoit un message BU de son MN
- Enfin, pour une architecture plus complète, des mécanismes doivent être mis en place pour que le QoS Agent mobile puisse aussi communiquer avec l'ARC. En effet, lors de ces expériences, nous considérions que la quantité de CRA allouée au niveau des STs était suffisante, mais cela ne couvre pas tous les scénarios possibles.

IV.5.2. Gestion de la QoS et de la mobilité pour les applications interactives par SIP

Pour cette solution, en plus des éléments généraux présentés au début de la partie IV.5, nous considérons que :

- Les UTs 13 et 33 sont constitués du client SIP amélioré pour la mobilité (voir IV.2.3.2.a). Nous testerons uniquement le cas où seul l'UT 13 est mobile. On considère aussi que le réseau d'origine du MN est le WLAN 1 ; donc son proxy SIP mère est au niveau de l'UT 12.
- Les proxies SIP (localisés au niveau des UTs 12, 22, 32) se voient ajouter les fonctionnalités concernant la gestion de la mobilité (voir IV.2.3.2.b).
- Une file DiffServ et une file MAC sont dédiées à la signalisation SIP au niveau de chaque ST.

On reprend ici le cas où une communication SIP avec QoS est mise en place entre les UTs 13 et 33 tandis que des flux concurrents saturent le lien satellite dans le sens MN→CN (et donc la bande passante disponible au niveau du ST1). Les QoS Servers ont donc configuré la QoS au niveau des STs tandis que l'ARC a validé l'augmentation de la capacité de CRA allouée aux STs concernés. Ensuite, au cours de la communication, l'UT 13 se déplace tout d'abord vers le réseau de son correspondant au niveau du *User WLAN 3* (déplacement 1) puis retourne dans son réseau initial (déplacement 2).

La Figure 60 permet d'observer les différentes étapes du scénario décrit précédemment sur la courbe du délai de transmission des paquets (Figure 60.A) ainsi que sur la courbe du débit moyen du flux vidéo (Figure 60.B).

Ainsi, entre t=0s et t=29,6s, malgré la présence de flux concurents, on retrouve le fait que les flux audio et vidéo sont protégés et ont donc un délai aux alentours de 300 ms. Ensuite, le MN se déplace dans le réseau de son CN et on observe bien que la communication reprend à t = 31,7s (voir Figure 60.B) directement entre les 2 correspondants (le délai est de quelques millisecondes seulement). Lorsque le proxy SIP en charge du nouveau réseau du MN, localisé au niveau de l'UT 32, reçoit le message REGISTER spécifique envoyé par le MN, il prévient

aussitôt le proxy SIP du réseau mère, localisé sur l'UT 12, en lui indiquant le changement de localisation du mobile, comme indiqué dans le paragraphe III.4.1.1. L'UT 12 procède alors à la libération de la QoS au niveau du ST1 et prévient l'ARC qui va ainsi modifier la quantité de CRA allouée au ST1 et à la GW/NCC. Le MN ré-initie alors la session en échangeant les messages re-INVITE/OK et ACK avec son CN. Lorsque le proxy SIP en charge du CN reçoit le message re-INVITE avec un Call-Id identique à une session en cours mais une adresse différente de celles des clients SIP associés à cette session, il comprend que c'est une modification de session avec changement de réseau et procède donc à la libération des ressources (reconfiguration de la taille des files DiffServ et annulation du marquage des paquets correspondant à cette session) auprès de la GW/NCC. Lorsqu'il reçoit ensuite le message OK, il analyse les adresses des correspondants et en déduit que la communication ne traversera plus le satellite : aucune re-réservation au niveau du système satellite n'est alors nécessaire.

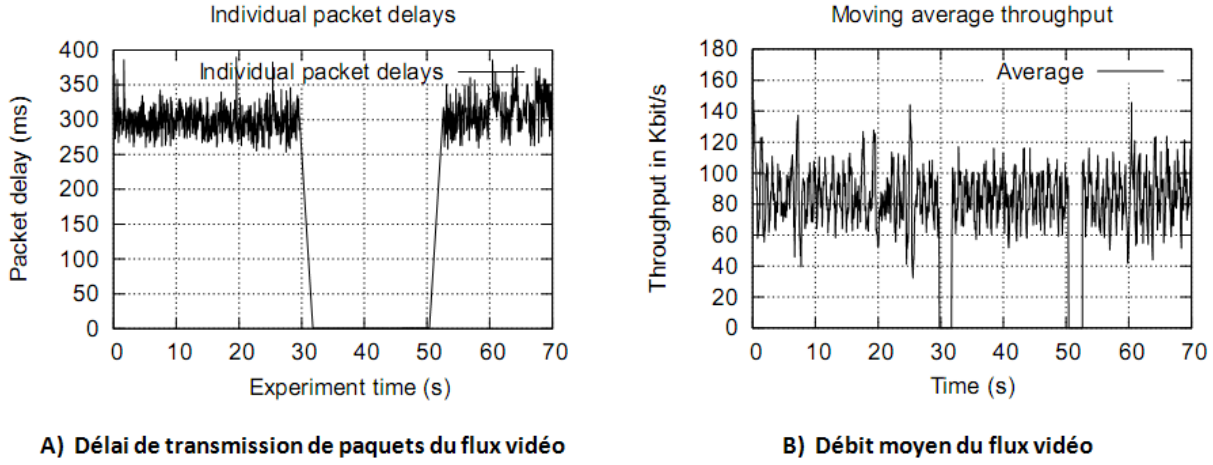


Figure 60 – Mobilité et QoS gérées par SIP

Ensuite, à t=50,4 s, un nouveau changement de réseau a lieu et la communication reprend à t=52,6 s : le MN est retourné dans son réseau d'origine et réalise donc la procédure de réenregistrement (enregistrement auprès du proxy SIP mère localisé sur l'UT 12 et désenregistrement auprès de l'ancien proxy SIP, situé sur l'UT 32), aucune libération de ressources n'étant ici nécessaire au niveau du système satellite. Finalement, il ré-initie la session avec QoS et la communication reprend bien avec un délai environ égal à 300 ms puisque la QoS a bien été reconfigurée automatiquement.

Pour ces expériences, nous avons utilisé le mode « Enabled » dans lequel les proxies SIP transmettent les messages OK avant que la réservation de QoS n'ait effectivement eu lieu. Cela permet en effet de réduire fortement le temps d'interruption, mais il est alors possible que la communication reprenne sans QoS ou avec une QoS partielle (e.g. dans un seul sens), par exemple dans le cas où la quantité de CRA allouée à un des STs concernés par la communication est déjà maximale.

Cependant, si l'on considère que le mode « Assured » est utilisé dans le cas d'un utilisateur SIP mobile qui se déplace derrière un ST qui ne permet pas d'effectuer la réservation de QoS, la communication va alors s'arrêter après le changement de réseau. L'opérateur doit donc décider de la politique qu'il va adopter pour ses clients ou bien leur

proposer différentes possibilités : il peut leur proposer en plus des services « Assured » et « Enabled », un service dans lequel la mise en place initiale de la session se fait en mode « Assured » mais où la ré-initiation de session en cas de changement de réseau se fait en mode « Enabled » pour limiter au maximum le temps d'interruption. En effet, toutes ses possibilités sont réalisables grâce aux préconditions définies dans le cadre de la RFC 3312 (voir paragraphe II.3.3).

IV.5.3. Conclusion sur les architectures couplant mobilité et QoS

Pour les mêmes raisons que celles indiquées dans le paragraphe IV.2.3.1 (échange de tous les messages FMIPv6 mais ensuite la communication ne reprend pas par le tunnel PAR-NAR et donc cela revient à utiliser Mobile IPv6), nous n'avons pas pu réaliser les expériences couplant FMIPv6 aux mécanismes de QoS basés sur le QoS Agent mobile et le QoS Server. Cependant, étant donné les résultats obtenus dans les parties précédentes concernant respectivement la QoS (voir paragraphe IV.3) et la mobilité (voir paragraphe IV.4) et en faisant le parallèle avec les résultats obtenus par l'architecture Mobile IPv6 + QoS, nous pouvons prévoir les différentes étapes d'une architecture FMIPv6 + QoS : ainsi, des phases de tunnel bidirectionnel (PAR-NAR puis HA-MN) assez longues dans lesquelles le délai de transmission des paquets serait de 600 ms interviendraient aussi ainsi que les mêmes problèmes de configuration de la QoS que Mobile IPv6 + QoS.

Concernant l'architecture couplant Mobile IPv6 aux mécanismes de QoS que nous avons développés, on a pu voir que des modifications doivent être réalisées au niveau de la pile MIPL pour lui permettre d'indiquer au QoS Agent mobile lorsqu'il doit effectuer des libérations de ressources au niveau de l'ancien QoS Server ou encore pour permettre au CN de re-réserver ses ressources en tenant compte du changement d'adresse du MN. Toutes ces modifications supplémentaires rendent cette solution compliquée à mettre en œuvre même si elle permet tout de même la gestion de la mobilité avec QoS pour toutes les applications d'un utilisateur mobile dans le cadre d'un système satellite DVB-S2/RCS.

Finalement, il apparaît clairement que l'architecture basée sur SIP que nous avons développée, couplant QoS et mobilité, est plus simple à mettre en œuvre et plus efficace pour la gestion des applications interactives (de type VoIP ou vidéoconférence) pour les mêmes raisons que celles énoncées dans les parties concernant respectivement la QoS et la mobilité (temps d'interruption généralement plus faible, aucun overhead supplémentaire, mise en place de la QoS plus dynamique, etc…). De plus, elle s'adapte plus facilement aux besoins d'un opérateur puisqu'elle permet de mettre en œuvre différents modes d'initiation ou de ré-initiation de session SIP en utilisant les préconditions définies dans [115]. Enfin, le fait que les réservations et libérations de QoS soient initiées au niveau des proxies SIP limite le nombre de messages échangés sur le réseau Wi-Fi ou plus généralement sur le réseau sans fil utilisé, ce qui permet une utilisation optimale des ressources contrairement à Mobile IPv6 + QoS.

Conclusion Générale

Bilan

La problématique de nos travaux a concerné la mise en œuvre d'architectures permettant l'intégration et l'adaptation de la mobilité et de la QoS aux réseaux satellites DVB-S2/RCS qui représentent actuellement une alternative prometteuse aux réseaux terrestres pour offrir l'accès Internet aux zones à faible densité de population ainsi que dans le cas de catastrophes naturelles ou d'événements ponctuels. Ces travaux ont considéré plus particulièrement le cas de réseaux IPv6.

Pour gérer la mobilité d'un utilisateur, différentes solutions ont alors été détaillées en parcourant les différents niveaux du modèle en couche et en précisant leurs avantages et inconvénients respectifs. Ensuite, pour répondre aux besoins des nouvelles applications, deux modèles de QoS ont été présentés, IntServ et DiffServ, pour la prise en compte de différents services au niveau IP, ainsi qu'une sélection de protocoles (COPS, NSIS, SIP) pouvant participer au processus de mise en œuvre de la QoS. Enfin, des solutions permettant de gérer simultanément la mobilité et la QoS ont été introduites et analysées.

A partir des analyses réalisées dans les deux premiers chapitres et de la présentation des caractéristiques propres à un système satellite DVB-S2/RCS ainsi que du projet SATSIX dans le cadre duquel une partie de nos travaux a été réalisée, notre première contribution a consisté à spécifier des mécanismes de QoS adaptés aux applications interactives à fortes contraintes temporelles (de type VoIP ou vidéoconférence) et aux spécificités des réseaux satellites, en conservant l'idée que ces mécanismes puissent s'adapter à un environnement mobile. Ainsi, ces mécanismes, basés sur une architecture DiffServ, comprennent :

- La configuration des files DiffServ aux niveaux des STs en fonction des informations reçues par leurs QoS Servers respectifs.
- L'utilisation d'un ARC, permettant la modification dynamique de la quantité de CRA allouée aux STs/GW, particulièrement utile dans le cas d'applications interactives à fortes contraintes temporelles.
- L'amélioration d'un proxy SIP permettant l'analyse des descripteurs de session compris dans les messages SIP et l'envoi d'information de QoS (débit, gigue max, taux de pertes max, délai max) au QoS Server et à l'ARC en tenant compte des médias inclus dans la session.
- L'utilisation de services Web pour déterminer les caractéristiques associées aux médias utilisés lors d'une session SIP.

Concernant la mobilité, nous nous sommes, une fois encore, particulièrement concentrés sur le cas des applications interactives à fortes contraintes temporelles qui doivent, à notre sens, être traitées avec une attention toute particulière étant donné leur fort développement actuel. Nous avons alors spécifié une solution basée sur le protocole SIP et adaptée au contexte satellite et l'avons comparé aux protocoles de la famille de Mobile IPv6. Cela nous a permis de proposer différentes recommandations quant à leur utilisation dans un système satellite.

Ensuite, notre dernière contribution a consisté à faire interagir nos solutions de mobilités et de QoS. Deux types d'architecture se sont alors détachés :

- Une architecture SIP basée sur un client SIP mobile et des proxies SIP améliorés pour la QoS et la mobilité qui permettent de gérer dynamiquement la ré-initiation de session ainsi que les réservations et libérations de QoS en fonction des déplacements du terminal mobile.
- Une architecture basée sur Mobile IPv6 ou FMIPv6 pour la gestion de la mobilité et sur un QoS Agent mobile communiquant avec des QoS Servers conscients de la mobilité pour la partie QoS.

Le chapitre 4 a finalement été consacré à l'évaluation des performances de nos architectures sur la plateforme d'émulation PLATINE. Après avoir décrit les différentes implémentations réalisées au cours de nos travaux et les différents outils d'analyse de trafic utilisés, nous avons pu prouver l'efficacité de nos mécanismes de QoS en constatant que le traitement différencié des flux au niveau des STs/GW permettait une utilisation plus efficace des ressources du système satellite lorsqu'il était correctement configuré. De plus, nous avons pu montrer qu'une architecture basée sur SIP se révélait plus efficace pour la gestion des applications interactives à fortes contraintes temporelles, aussi bien en termes de temps d'interruption et d'utilisation des ressources que de gestion dynamique de la QoS. Cependant, l'évaluation de l'architecture de QoS et de mobilité basée sur Mobile IPv6 ou FMIPv6 a aussi permis de montrer qu'il était possible d'offrir des mécanismes de mobilité et de QoS pour tous les types d'applications, même si ce caractére générique est aussi à l'origine de ses principaux défauts.

Un autre point important de comparaison de ces deux architectures est leur capacité à s'adapter à d'autres types d'architectures réseaux ainsi qu'à être mise en œuvre sur une infrastructure réelle. Sur ce dernier point, l'architecture SIP semble là encore plus efficace. En effet, alors que Mobile IPv6 et ses extensions sont spécifiquement conçus pour IPv6 et nécessitent un certain nombre de modifications de l'infrastructure réseau (ajout de HA, MAP, PAR, NAR, etc…), une architecture basée sur SIP est aussi bien utilisable sur IPv4 que sur IPv6 et requiert uniquement quelques modifications au niveau des proxies et clients SIP. De plus, SIP a été choisi comme protocole de contrôle de session pour des architectures NGN telles que IMS ou EuQoS donc les mécanismes de QoS et de mobilité basés sur SIP décrits dans nos travaux peuvent y être intégrés plus aisément, d'autant plus que SIP fait aussi l'objet de recherche dans le cadre de la gestion de la mobilité d'un utilisateur IMS se déplaçant d'un réseau à commutation de paquets (type Wi-Fi) à un réseau à commutation de circuits (type GSM), sous le terme « IMS Service Continuity » [168].

Perspectives

Les différents travaux réalisés au cours de ce mémoire permettent alors d'envisager un certain nombre de perspectives.

Tout d'abord, concernant la partie purement QoS, pour la configuration des files DiffServ par le QoS Server, nous nous basons uniquement sur le paramètre du débit (ou éventuellement sur deux paramètres correspondants au débit crête et au débit moyen) alors que le proxy SIP et le QoS Agent envoient des messages XML comportant aussi des informations sur la gigue, le délai et le taux de pertes. Il sera alors intéressant d'utiliser ces dernières informations.

De plus, pour nos expériences, nous avons utilisé des réseaux Wi-Fi (couplés au réseau satellite) mais nous nous sommes uniquement intéressés à la configuration de la QoS au niveau du système satellite. Il sera alors intéressant d'étudier comment notre proxy SIP amélioré (pas forcément le même que celui qui configure le ST) peut aussi initier des mécanismes de QoS au niveau du Wi-Fi ainsi que pour d'autres technologies telles que le WiMAX. Or, nous pouvons déjà prévoir ce fonctionnement car maintenant que nous avons ajouté au QoS Agent la capacité de recevoir des messages RESV et FREE, il suffit de lui ajouter la fonctionnalité de marquage de paquet (la même que pour le QoS Server) pour qu'il affecte à un flux une classe de service DiffServ qui peut ensuite être mappée directement sur une des catégories d'accès Wi-Fi. Un exemple de fonctionnement peut alors être : un utilisateur se connecte à un ST donné, il obtient l'adresse du proxy SIP en charge du domaine et s'enregistre auprès de lui (enregistrement SIP avec ajout d'une option « adresse et port du QoS Agent » ou alors enregistrement du QoS Agent directement auprès du proxy SIP de la même manière qu'avec le QoS Server) ; cet utilisateur veut ensuite initier une session de vidéoconférence par SIP, le proxy SIP intercepte les messages SIP et prévient alors le QoS Agent en plus du QoS Server. Deux solutions sont alors possibles : soit le marquage se fait uniquement au niveau du QoS Agent et le QoS Server reconfigure uniquement la taille des files DiffServ du ST, soit le marquage peut se faire au niveau des deux entités en supposant que des politiques différentes peuvent être appliquées au niveau du réseau Wi-Fi et du réseau satellite. Dans les deux cas, les flux audio et vidéo seront priorisés au niveau du réseau IEEE802.11e et au niveau du système satellite.

Enfin, concernant les services Web, on pourra étendre leurs capacités en permettant, par exemple, au proxy SIP de savoir à quel niveau de QoS (classe de service, nécessité d'une allocation CRA) il doit associer telle ou telle type d'application (audio, vidéo, text, etc…) puisque pour l'instant, cette configuration est statique. Les services Web pourront aussi permettre au proxy SIP de connaître la quantité de CRA encore disponible pour son ST correspondant, pour qu'il puisse déterminer par exemple quels codecs sont acceptables ou non, lors de la réception de l'initiation d'une session SIP.

Ensuite concernant la partie purement mobilité, il faudra étudier le cas où les deux utilisateurs sont mobiles et plus particulièrement lorsqu'ils changent de réseau au même moment pour voir comment Mobile IPv6 et ses extensions ou la mobilité SIP peuvent gérer cette situation.

Enfin, concernant le couplage de la mobilité et de la QoS, comme on l'a déjà précisé, il sera indispensable de permettre une communication de type cross-layer entre Mobile IPv6 et le QoS Agent mobile pour indiquer précisément à ce dernier quand il doit effectuer les réservations et libérations au niveau du MN et permettre au CN de mettre à jour ses réservations.

De plus, nous nous sommes focalisés sur un utilisateur mobile qui utilise la même interface avant et après son déplacement mais il sera intéressant d'évaluer le cas d'un utilisateur pouvant utiliser le « multi-homing ». Dans le cas de SIP par exemple, on pourra alors ré-initier une session depuis une 2ème interface (et utiliser le mode « assured », ainsi que les spécifications de la RFC 3312, sans être pénalisant en termes de temps d'interruption) tandis que la communication continue sur la première. On pourra ainsi obtenir une mobilité « sans couture » avec garantie de QoS.

L'utilisation de MIH pourra aussi améliorer la gestion de la mobilité et de la QoS en permettant par exemple d'échanger des informations sur le type de liens disponibles et les classes de service associées ou encore, dans le cas de SIP, de prévenir le client SIP mobile qu'un changement de réseau va intervenir et qu'il faut qu'il ré-initie la session, etc…

Ensuite, il sera intéressant de voir comment notre solution basée sur SIP pourrait s'intégrer à une architecture de type IMS. Nos mécanismes de QoS pourront ainsi être couplés à ceux d'IMS basés sur les PEPs et les PDPs. Ainsi, concernant le système satellite en lui-même, un PDP serait positionné au niveau du NCC/GW tandis qu'un PEP serait utilisé au niveau de chaque ST. On pourrait alors aussi voir comment la QoS pourrait être configurée de bout-en-bout en se basant sur le concept de nos proxies SIP améliorés.

Enfin, pour une solution plus globale, il faudra permettre aux deux solutions d'interagir en évitant toutefois des messages de signalisations redondants et ayant le même objectif (par exemple BU échangé entre MN et HA et REGISTER échangé entre client SIP et proxy SIP mère). La mobilité SIP gérerait les applications interactives tandis que Mobile IPv6 et ses extensions pourrait être en charge des autres applications, à faible contraintes temporelles.

Bibliographie

[1] H. Schulzrinne and E. Wedland. “Application-Layer Mobility using SIP”, ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 4, Number 3, July 2000, pp.47-57.

[2] J. Manner and M. Kojo, “Mobility Related Terminology”, IETF RFC 3753, June 2004.

[3] D. Kempf, “Problem Statement for Network-Based Localized Mobility Management (NETLMM)”, IETF RFC 4830, April 2007.

[4] Data Processing Open System Interconnection, Basic Reference Model, ISO IS 7498, 1984.

[5] 80211. IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999), pages C1– 1184, 12 2007.

[6] 802.11g. IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band.

[7] IEEE 802.11n-2009 Standard for Information technology— Telecommunications and information exchange between systems - Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) & Physical Layer specifications Enhancements for Higher Throughput

[8] 802.11f. IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, IEEE Std 802.11f-2003, July 2003.

[9] 802.11k. IEEE Standard for Information technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1 : Radio Resource Measurement of Wireless LANs. IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007), pages c1–222, 12 2008.

[10] 802.11r. IEEE Standard for Information technology – Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 2: fast Basic Service Set (BSS). IEEE Std 802.11r-2008 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008), pages c1–108, July 2008.

[11] D. Loher, D. Nelson, O. Volinsky and B. Sarikaya, “Evaluation of Candidate Control and Provisioning of Wireless Access Points (CAPWAP) Protocols”, IETF RFC 4565.

[12] P. Calhoun, M. Montemurro and D. Stanley, “Control and Provisioning of Wireless Access Points Protocol Specifications”, IETF RFC 5415, March 2009.

[13] CAPWAP Open Source project, http://sourceforge.net/projects/capwap.

[14] P. McCann, “Mobile IPv6 Fast Handovers for 802.11 Networks”, IETF RFC 4260, November 2005.

[15] R. Koodli, "Mobile IPv6 Fast Handovers", IETF RFC 5568, July 2009.

[16] E. Ivov Petrov, J. Montavont and T. Noël, “Thorough Empirical Analysis of the IETF FMIPv6 protocol over IEEE 802.11 networks”, IEEE Wireless Communications Magazine, Special Issue on Architectures and Protocols for Mobility Management in All-IP Mobile Networks, Volume 15, Number 2, page 65-72, April 2008.

[17] 802.16. IEEE Standard for Local and Metropolitan area Networks Part16: Air Interface for fixed broadband Wireless Access Systems, 2004.

[18] IEEE 802.16e, IEEE Standard for Local and Metropolitan Area Networks, Air Interface for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1, February 2006 (Approved: 7 December 2005).

[19] J. Postel, “Internet Protocol”, IETF RFC 791, September 1981.

[20] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, IETF RFC 2460, December 1998.

[21] R. Droms, “Dynamic Host Configuration Protocol”, IETF RFC 2131, March 1997.

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

[23] S. Thomson, T. Narten and T. Jinmei, “IPv6 Stateless Address Autoconfiguration”, IETF RFC 4862, December 1998.

[24] J. Postel and K. Harrenstien, “Time Protocol”, IETF RFC 868, May 1983.

[25] K. Sollins, “The TFTP Protocol (Revision 2)”, IETF RFC 1350, July 1992.

[26] WiMAX Forum: “A Technical Overview and Performance Evaluation,” Mobile WiMAX – Part I, August 2006.

[27] C. Perkins, “IP Mobility Support for IPv4”, IETF RFC 3344, August 2002.

[28] H. Jang, J. Jee, Y. Han, S. Park and J. Cha, “Mobile IPv6 Fast Handovers over IEEE 802.16e Networks”, IETF RFC 5270, June 2008.

[29] WiMAX Forum Network Working Group, "WiMAX Forum Network Architecture—Stage 2: Architecture Tenets, Reference Model and Reference Points—Release 1, Version 1.2," WiMAX Forum, January 2008.

[30] WiMAX Forum Network Working Group, "WiMAX Forum Network Architecture—Stage 3: Detailed Protocols and Procedures—Release 1, Version 1.2," WiMAX Forum, January 2008.

[31] D. Johnson, C. Perkins and J. Arkko, “Mobility Support in IPv6”, IETF RFC 3775, June 2004.

[32] T. Bchini, N. Tabbane, S. Tabbane, E. Chaput, A.L. Beylot, “Inter-ASN Handover Using MSCTP Protocol in IEEE 802.16e Networks”, 2009 Seventh Annual Communication Networks and Services Research Conference (CNSR’09), May 2009.

[33] S.J. Koh, M.J. Chang and M. Lee, “mSCTP for Soft Handover in Transport Layer,” IEEE Communication Letters, Vol. 8, No.3, pp.189-191, March 2004.

[34] P. Srisuresh and K. Egevang, “Traditional IP Network Address Translator (Traditional NAT)”, IETF RFC 3022, January 2001.

[35] V. Fuller and T. Li, “Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan”, IETF RFC 4632, August 2006.

[36] J. Arkko, V. Devarapalli, and F. Dupont. “Using IPsec to Protect Mobile IPv6 Signaling Beteween Mobile Nodes and Home Agents”, IETF RFC 3776, June 2004.

[37] S. Kent and K. Seo, “Security Architecture for the Internet Protocol”, RFC 4301, December 2005.

[38] C. Perkins, "Securing Mobile IPv6 Route Optimization Using a Static Shared Key", IETF RFC 4449, June 2006.

[39] C. Vogt and J. Arkko, "A Taxonomy and Analysis of Enhancements to Mobile IPv6 Route Optimization", IETF RFC 4651, February 2007.

[40] J. Arkko and C. Vogt, and W. Haddad, "Enhanced Route Optimization for Mobile IPv6", IETF RFC 4866, May 2007.

[41] R. Wakikawa, “Home Agent Reliability Protocol”, IETF Internet Draft, draft-ietf-mip6-hareliability-04.txt, July 2008.

[42] H. Soliman, C. Castelluccia, K. El Malki and L. Bellier, “Hierarchical Mobile IPv6 (HMIPv6) Mobility Management”, IETF RFC 5380, October 2008.

[43] Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture", IETF RFC 4877, April 2007.

[44] H. Soliman, C. Castelluccia, K. El Malki and L. Bellier. “Hierarchical Mobile IPv6 Mobility Management (HMIPv6)”, RFC 4140, August 2005.

[45] S. Gundavelli, K. Lung, V. Devarapalli, K. Chowdhury and B. Patil, “Proxy Mobile IPv6”, IETF RFC 5213, August 2008.

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

[47] B. Sarikaya, A. Qin, A. Huang and W. Wu, “PMIPv6 Route Optimization Protocol,” IETF draft-qin-netlmm-pmipro-00,” February 2008.

[48] M. Liebsch, L. Le and J. Abeille, “Route Optimization for Proxy Mobile IPv6,” IETF draft-abeille-netlmm-proxymip6ro-01, November 2007.

[49] T.Chiba, H. Yokota, A.Dutta, D. Chee, H. Schulzrinne, “Route Optimization Techniques for Proxy MIPv6 in IMS Network,” IEEE second International Conference on Signal Processing and Communication Systems (ICSPCS’2008), December 2008.

[50] H. Soliman, “Mobile IPv6 Support for Dual Stack Hosts and Routers”, IETF RFC 5555, June 2009.

[51] R. Moskowitz and P. Nikander, “Host Identity Protocol (HIP) Architecture”, IETF RFC 4423, May 2006.

[52] W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory vol. IT-22, number 6, pages 644-654, Nov 1976.

[53] R. Moskowitz, P. Nikander, P. Jokela and T. Henderson, “Host Identity Protocol”, IETF RFC 5201, April 2008.

[54] P. Nikander, T. Henderson, C. Vogt and J. Arkko, “End-Host Mobility and Multihoming with the Host Identity Protocol”, IETF RFC 5206, April 2008.

[55] J. Laganier and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", IETF RFC 5204, April 2008.

[56] Nikander, P. and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", IETF RFC 5205, April 2008.

[57] J. Postel, “Transmission Control Protocol”, IETF RFC 793, September 1981.

[58] J. Postel, “User Datagram Protocol”, IETF RFC 768; August 1980.

[59] D.A. Maltz and P. Bhagwat, "MSOCKS: An Architecture for Transport Layer Mobility", IEEE Conference on Computer Communications (INFOCOM’98), San Francisco, CA, pp. 1037 - 1045, March 29 - April 2, 1998.

[60] D.A. Maltz and P. Bhagwat, "TCP Splicing for Application Layer Proxy Performance" IBM Research Report RC 21139, IBM T.J. Watson Research Center, March 1998.

[61] A. Bakre and B.R. Badrinath, "I-TCP: indirect TCP for mobile hosts", IEEE International Conference on Distributed Computing Systems, Vancouver, Canada, pp. 136 - 143, May 30 - June 2, 1995.

[62] K. Brown and S. Singh, "M-UDP: UDP for mobile cellular networks" Computer Communication Review, vol. 26, no. 5, pp. 60 - 78, October 1996.

[63] A.C. Snoeren and H. Balakrishnan, "An End-to-End approach to Host Mobility," 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, pp. 155 - 166, August 2000.

[64] D. Eastlake 3rd, "Secure Domain Name System Dynamic Update", IETF RFC 2137, April 1997.

[65] P. Vixie, S. Thomson, Y. Rekheter and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", IETF RFC 2136, April 1997.

[66] S. Fu, L. Ma, M. Atiquzzaman and Y. Lee, "Architecture and Performance of SIGMA: A Seamless Handover Scheme for Data Networks," IEEE ICC, Seoul, South Korea, May 2005.

[67] R. Stewart, "Stream Control Transmission Protocol", IETF RFC 4960, September 2007.

[68] R. Stewart, Q. Xie, M. Tuexen, S. Maruyama and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", IETF RFC 5061, September 2007.

[69] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.

[70] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP : A Transport Protocol for Real-Time Applications", IETF RFC 3550, July 2003.

[71] M. Handley, V. Jacobson and C. Perkins, "SDP : Session Description Protocol", IETF RFC 4566, July 2006.

[72] R. Fielding, J. gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee, "Hypertext Transfert Protocol – HTTP/1.1", IETF RFC 2616, June 1999.

[73] J. Klensin, "Simple Mail Transfert Protocol", IETF RFC 2821, April 2001.

[74] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 8). 3GPP TS 23.228 V8.7.0.

[75] J. Rosenberg and H. Schulzrinne, "Session Initiation Protocol (SIP) : Locating SIP Servers", IETF RFC 3263, June 2002.

[76] H. Schulzrinne, "SIP Registration", IETF Internet Draft, draft-schulzrinne-sip-register-01.txt, April 2001.

[77] A. Floris and Luca Veltri, "Roaming Scenarios Based on SIP", Proceedings of the 5th IFIP/IEEE International Conference on Management of Multimedia Networks and Services: Management of Multimedia on the Internet, vol. 2496, pp. 302 – 314, October 2002.

[78] IEEE802.21 Media Independent Handover Services, http://www.ieee802.org/21/

[79] Mobility for IP: Performance, Signaling and Handoff Optimization (MIPSHOP), http://www.ietf.org/dyn/wg/charter/mipshop-charter.html

[80] Q.B. Mussabbir and W. Yao, "Optimized FMIPv6 Handover using IEEE802.21 MIH Services", First ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture (MobiArch'2006), December 2006.

[81] K.N. Choong, V.S. Kesavan, S.L. Ng, F. de Carvalho, A.L.Y. Low and C. Maciocco, "SIP-based IEEE802.21 media independent handover - a BT Intel collaboration", BT Technology Journal, Vol. 25, No. 2, pp. 219 – 230, April 2007.

[82] Y.M. Chen, M.Y. Lai, S.C. Lin, S.C. Chang and T.Y. Chung, "SCTP-based handoff based on MIH triggers information in campus networks", The 8th International Conference of Advanced Communication Technology (ICACT'06), February 2006.

[83] A. Cabellos-Aparicio and J. Domingo-Pascual "Enhanced Fast Handovers Using a Multihomed Mobile IPv6 Node", Proceedings of the 6th International Conference of Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN'06), May 29 – June 2, 2006.

[84] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst and K. Nagami, "Multiple Care-of Addresses Registration", IETF RFC 5648, October 2009.

[85] M.K. Park, J.Y Lee, B.C. Kim and D.Y. Kim, "Design of Fast Handover Mechanism for Multiple Interfaces Mobile IPv6", International Symposium on Wireless Pervasive Computing (ISWPC'08), May 2008.

[86] R. Mahy and D. Petrie, "The Session Initiation Protocol (SIP) Join Header", IETF RFC 3911, October 2004.

[87] N. Banerjee, S.K. Das and A. Acharya, "SIP-based mobility architecture for Next Generation Wireless Networks", 3d IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), March 2005.

[88] J. Zhang, H.C.B Chan and V.C.M Leung, "A SIP-Based Seamless Handoff (S-SIP) Scheme for Heterogeneous Mobile Networks", IEEE Wireless Communications and Networking Conference (WCNC'07), March 2007.

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

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

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

[92] W. C. Hardy, "QoS Measurements and Evaluation of Telecommunication Quality of Service", Wiley, 2001.

[93] ITU-T-Rec. G.1010, “End-user Multimedia QoS Categories”, Technical Report, International Telecommunication Union, 2001.

[94] R. Braden, D. Clark and S. Shenker, “Integrated Services in the Internet Architecture : an Overview”, IETF RFC 1633, June 1994.

[95] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, “Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification”, IETF RFC 2205, September 1997.

[96] S. Shenker, C Partridge and R. Guerin, “Specification of Guaranteed Quality of Service”, IETF RFC 2212, September 1997.

[97] J. Wroclawski, “Specification of the Controlled-Load Network Element Service”, IETF RFC 2211, September 1997.

[98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, “An Architecture for Differentiated Service”, IETF RFC 2475, December 1998.

[99] D. Grossman, “New Terminology and Clarifications for DiffServ”, IETF RFC 3260, April 2002.

[100] K. Nichols, V. Jacobson and L. Zhang, “A Two-bit Differentiated Services Architecture for the Internet”, IETF RFC 2638, July 1999.

[101] V. Jacobson, K. Nichols and K. Poduri, “An Expedited Forwarding PHB”, IETF RFC 2598, June 1999.

[102] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, “Assured Forwarding PHB Group”, IETF RFC 2597, June 1999.

[103] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski and E.Felstaine, “A Framework for Integrated Services Operation over DiffServ Networks”, IETF RFC 2998, November 2000.

[104] R. Yavatkar, D. Pendarakis and R. Guerin, “A Framework for Policy-Based Admission Control”, IETF RFC 2753, January 2000.

[105] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry and S. Waldbusser, “Terminology for Policy-Based Management”, IETF RFC 3198, November 2001.

[106] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan and A. Sastry, “The COPS (Common Open Policy Service) Protocol”, IETF RFC 2748, January 2000.

[107] S. Herzog, J. Boyle, R. Cohen, D. Durham, R. Rajan and A. Sastry, “COPS Usage for RSVP”, IETF RFC 2749, January 2000.

[108] K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar and A. Smith, “COPS Usage for Policy Provisioning (COPS-PR)”, IETF RFC 3084, March 2001.

[109] NSIS Work Group, http://www.ietf.org/dyn/wg/charter/nsis-charter.html

[110] R. Hancock, G. Karagiannis, J. Loughney and S. Van den Bosh, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005.

[111] H. Schulzrinne and R. Hancock, "GIST: General Internet Signalling Transport", IETF Draft, draft-ietfnsis-ntlp-20, June 2009.

[112] J. Manner, G. Karagiannis and A. McDonald, "NSLP for Quality-of-Service Signaling", IETF Draft, draft-ietf-nsis-qos-nslp-17.txt, October 2009.

[113] M. Stiemerling, H. Tschofenig, C. Aoun and E. Davies, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", IETF Draft, draft-ietf-nsis-nslp-natfw-20.txt, November 2008.

[114] T. Sanda, X. Fu, S. Jeong, J. Manner and H. Tschofenig, "Applicability Statement of NSIS Protocols in Mobile Environments", IETF Draft, draft-ietf-nsis-applicability-mobility-signaling-13.txt, July 2009.

[115] G. Camarillo, W. Marshall and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", IETF RFC 3312, October 2002.

[116] ITU-T-Rec. Y.2001, General Overview of NGN. Technical Report, International Telecommunication Union, 2001.

[117] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 8). 3GPP TS 23.228 V8.10.0, 2009.

[118] E. Mingozzi, G. Stea, M.A. Callejo-Rodríguez, J. Enríquez-Gabeiras, G. García-de-Blas, F.J. Ramón-Salquero, W. Burakowski, A. Beben, J. Sliwinski, H. Tarasiuk, O. Dugeon, M. Diaz, L. Baresse and E. Monteiro, "EuQoS: End-to-End Quality of Service over Heterogeneous Networks", Computer Communications, vol. 32, pp. 1355 – 1370, July 2009.

[119] H. Chaskar, "Requirements of a Quality of Service (QoS) Solution for Mobile IP", IETF RFC 3583, September 2003.

[120] A.K. Talukdar, B.R. Badrinath, A. Acharya, "MRSVP: A Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts", Wireless Networks, vol. 7, pp. 5 – 19, October 2004.

[121] N.F. Huang and W.E. Chen, "RSVP Extensions for Real-Time Services in Hierarchical Mobile IPv6", Mobile Networks and Applications, vol. 8, pp. 625 – 634, December 2003.

[122] S. Elleingand, S. Pierre, "FH-RSVP scheme for intra-site handover in hierarchical mobile IPv6 networks", Computer Communications, vol. 30, pp. 416 – 427, 2007.

[123] A. Terzis, J. Krawczyk, J. Wroclawski and L. Zhang, "RSVP Operation Over IP Tunnels", IETF RFC 2746, January 2000.

[124] T. Braun, C. Castelluccia, and G. Stattenberger, “An Analysis of the DiffServ Approach in Mobile Environments”, IWQiM´99, April 1999

[125] J.C. Chen, A. McAuley, A. Caro, S. Baba, Y. Ohba, P. Ramanathan, "QoS Architecture Based on Differentiated Services for Next Generation Wireless IP Networks", draft-itsumo-wireless-diffserv-00.txt, July 2000

[126] H. Chaskar and R. Koodli, “A Framework for QoS Support in Mobile IPv6”, draft-chaskar-mobileip-qos-00.txt, November 2000.

[127] Z. Kan, D. Zhang, R. Zhang and J. Ma, “QoS in Mobile IPv6”, International Conference on Info-tech and Info-net (ICII’01), 29 October – 01 November 2001.

[128] M. Kim and Y. Mun, “Cost Evaluation of Differentiated QoS Model in Mobile IPv6 Networks”, International Conference on Computational Science and Its Application (ICCSA’06), May 2006.

[129] J. Loughney, M. Nakhjiri, C. Perkins and R. Koodli, “Context Tranfert Protocol (CXTP)”, IETF RFC 4067, July 2005.

[130] C. Liu, D. Qian, Y. Liu and K. Xiao, “A Framework for End-to-End QoS Context Transfert in Mobile IPv6”, International Conference on Personal Wireless Communications (PWC’04), September 2004.

[131] W. Sun, S. Yang and D. Wang, “QoS Pre-Configure Mechanism on DiffServ Mobile IPv6 Networks”, International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM’06), September 2006.

[132] B.H. Ahn, D.Y. Kim, K.H. Cho, S.H. Cha and M. Jo, “An Efficient Resource Reservation and QoS Provisioning Mechanism Based on mSCTP for Next Generation Network”, 5th International Conference on Software Engineering Research, Management and Applications (SERA’2007), August 2007.

[133] H.T. Nguyen, T.H. Nguyen, N.L. Tran, Q.T. Tran, H. Do and T. Magedanz, “mSCTP-based Proxy in Support of Multimedia Session Continuity and QoS for IMS-based Networks”, 2nd International Conference on Communications and Electronics (ICCE’2008), June 2008.

[134] ETSI EN 300 421 v1.1.2, “Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for the 11/12 GHz satellite services”, August 1997.

[135] ETSI EN 301 790 v1.4.1, “Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems”, April 2005.

[136] ETSI EN 302 307 v1.1.2, “Digital Video Broadcasting (DVB), Second Generation Framing Structure, Channel Coding And Modulation Systems For Broadcasting, Interactive Services, News Gathering And Other Broadband Satellite Applications”, June 2006.

[137] SatLabs System Recommendations – Quality of Service Specifications, June 2008.

[138] IST SATSIX project, http://www.ist-satsix.org/

[139] IST SATIP6 project, IST-2001-34344 contract.

[140] O. Alphand, P. Berthou, T. Gayraud and S. Combes, “QoS Architecture over DVB-RCS satellite networks in a NGN framework”, IEEE Global Communications Conference 2005 (Globecom’2005), December 2005.

[141] World Wide Web Consortium, http://www.w3.org/2002/ws/

[142] ETSI EN 301 790 v1.5.1, “Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems”, January 2009.

[143] A. Mishra, M. Shin and W. Arbaugh, “An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process”, ACM SIGCOMM Computer Communication Review, vol. 33, pp. 93 – 102, April 2003.

[144] N. Montavont and T. Noël, “Analysis and Evaluation of Mobile IPv6 Handovers over Wireless LAN”, Mobile Networks and Applications, vol. 8, n°6, pp. 643 – 653, December 2003.

[145] N. Nakajima, A. Dutta, S. Das and H. Schulzrinne, “Handoff Delay Analysis and Measurement for SIP based Mobility in IPv6”, IEEE International Conference on Communications (ICC’03), vol.2, pp. 1085 – 1089, May 2003.

[146] M. Moore, “Optimistic Duplicate Address Detection (DAD) for IPv6”, IETF RFC 4429, April 2006.

[147] TR102157 v1.1.1. ETSI TC SES; Broadband Satellite Multimedia; IP Interworking over satellite: Performance, Availability and Quality of Service, July 2003.

[148] C. Baudoin, P. Berthou, T. Gayraud, F. Nivor, B. Jacquemin, D. Barvaux and J. Nicol, “PLATINE: DVB-S2/RCS enhanced testbed for next generation satellite networks,” in IP Networking over Next-Generation Satellite Systems, International Workshop, Budapest, July 2007.

[149] Margouilla c++ Runtime : http://cqsoftware.free.fr/margouilla

[150] HTB for Linux, http://luxik.cdi.cz/~devik/qos/htb/manual/theory.htm

[151] Simple DNS Plus : http://www.simpledns.com/

[152] ISC DHCP Server : https://www.isc.org/downloadables/12

[153] Linux IPv6 Router Advertisement Daemon : http://www.litech.org/radvd/

[154] D. L. Mills, “Network Time Protocol (Version 3) Specification, Implementation and Analysis”, IETF RFC 1305, March 1992.

[155] O. Alphand, “Architecture à qualité de service pour systèmes satellites DVB-S/RCS dans un contexte NGN”, s.l : Rapport LAAS N°05672, Décembre 2005.

[156] Jugi’s Traffic Generator (JTG), http://www.cs.helsinki.fi/u/jmanner/software/jtg/

[157] The Multi-Generator (MGEN), http://pf.itd.nrl.navy.mil/mgen/mgen.html#_MGEN_Log_File

[158] OreNETa (One Way Delay Real-time Network Analyser), http://cba.upc.edu/oreneta

[159] USAGI-patched Mobile IPv6 for Linux, http://umip.linux-ipv6.org/

[160] FMIPv6 project, http://www.fmipv6.org/

[161] R. Koodli, "Fast Handovers for Mobile IPv6", IETF RFC 4068, July 2005.

[162] Silogic, http://www.silogic.fr/

[163] JAIN-SIP Project, https://jain-sip.dev.java.net/

[164] National Institute of Standards and Technology – IP telephony project, http://snad.ncsl.nist.gov/proj/iptel/

[165] Apache Web Services Project – Axis, http://ws.apache.org/axis/

[166] Apache Tomcat, http://tomcat.apache.org/

[167] M. Gineste, B. Jacquemin, P. Berthou, A. El Fatni, T. Gayraud and C. Baudoin, "A Web Services Based Resource Signaling Scheme in Multimedia Satellite Systems", 4th Advanced Satellite Mobile Systems Conference (ASMS'2008), August 2008.

[168] 3rd Generation Partnership Project; Technical Specification Group Services and Architecture; IP Multimedia Subsystem (IMS) Service Continuity; Stage 2 (Release 10). 3GPP TS 23.237 V10.0.0, December 2009.

[169] B. Jacquemin, "Utilisation de l'outil TC pour la configuration de classes de service DiffServ ", http://hal.archives-ouvertes.fr/hal-00470674/fr/.

[170] Qi Wang and Mosa Ali Abu-Rgheff, "Signalling analysis of cost-efficient mobility support by integrating mobile IP and SIP in all IP wireless networks", International Journal of Communication Systems, 2006.

Publications

Conférences Internationales

- B. Jacquemin, P. Berthou, T. Gayraud and C. Baudoin, “Dynamic QoS Configuration of a DVB-RCS Satellite Terminal for SIP-based Applications”, submitted to 16th Ka and Broadband Communications Conference.
- C. Baudoin, F. Arnal, I. Buret, B. De la Cuesta, A.C. Salas, J.A. Torrijos, A. Ramos, P. Zautasvili, B. Jacquemin, M. Gineste, T. Gayraud and P. Berthou, “First end to end IPv6 DVB-RCS systems”, 4th Advanced Satellite Mobile Systems Conference (ASMS 2008), Bologne (Italie), 25-28 Août 2008, 6p.
- M. Gineste, B. Jacquemin, P. Berthou, A. El Fatni, T. Gayraud and C. Baudoin, “A web services based resource signaling scheme in multimedia satellite systems”, 4th Advanced Satellite Mobile Systems Conference (ASMS 2008), Bologne (Italie), 25-28 Août 2008, 7p.
- F. Arnal, T. Gayraud, C. Baudoin and B. Jacquemin, “IP mobility and it's impacts on satellite networking”, 4th Advanced Satellite Mobile Systems Conference (ASMS 2008), Bologne (Italie), 25-28 Août 2008, 6p.
- B. Jacquemin, P. Berthou, T. Gayraud and C. Baudoin, “QoS et mobilité dans un système DVB-S/RCS”, Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 2008), Les Arcs (France), 25-28 Mars 2008, 12p.
- C. Baudoin, M. Dervin, P. Berthou, T. Gayraud, F.Nivor, B. Jacquemin, D. Barvaux, J. Nicol, “PLATINE: DVB-S2/RCS enhanced testbed for next generation satellite networks”, International Workshop on IP Networking over Next-generation Satellite Systems (INNSS'07), Budapest (Hongrie), 5 Juillet 2007, 11p.
- I. Melhus, F. Arnal, T. Gayraud and B. Jacquemin, “SATSIX Mobility Architecture and its performance evaluation”, International Workshop on IP Networking over Next-generation Satellite Systems (INNSS'07), Budapest (Hongrie), 5 Juillet 2007, 17p.
- T. Gayraud, O. Alphand, P. Berthou, B. Jacquemin and C. Baudoin, “Mobility architectures for DVB-S/RCS satellite networks”, 12th Ka and Broadband Communications Conference, Naples (Italie), 27-29 Septembre 2006, 8p.

Conférences Nationales

- B. Jacquemin, T. Gayraud, P. Berthou and C. Baudoin, “Mobilité et QoS pour les applications SIP dans les réseaux DVB-S/RCS”, EDSYS 2008. 9ème Congrès de Doctorants, Toulouse (France), Mai 2008.
- B. Jacquemin, T. Gayraud and P. Berthou, “Mobilité et QoS pour les applications multimédias SIP dans un système satellite DVB-S/RCS”, Ecole d'Eté RESCOM 2007. Réseaux Autonomes et Internet du Futur, Calcatoggio (France), 16-23 Juin 2007, 2p.
- B. Jacquemin, T. Gayraud and P. Berthou, “Architecture de mobilité dans un système satellite DVB-S/RCS”, 8èmes Journées Doctorales en Informatique et Réseaux (JDIR'2007), Marne la Vallée (France), 17-19 Janvier 2007, pp.143-151.

Rapports de contrat et rapports techniques

- B. Jacquemin, T. Gayraud, P. Berthou and M. Gineste, “Services contunuity for mobile users in hybrid terrestrial/satellite IPv6 networks”, Rapport LAAS N°09564, Septembre 2009.
- N. Hennion, J.A. Torrijos, P. Zautasvili, G. Banszki, F. Goas, A. Yun, E. Callejo, B. De La Cuesta, .J.A. Guerra, M. Catalan de Domingo, R. Munoz, T. Gauraud, B. Jacquemin, P. Berthou and C.A. Gil Garcia, “Live trials definition (D3000-6)”, Rapport LAAS No08543 Integrated Project 26950: SATSIX, Octobre 2008, 51p.
- G. Faihurst, G. Renker, F. Goas, E. Pechereau, P. Berthou, B. Jacquemin, M. Gineste, T. Gayraud, I. Tou, C. Baudoin, F. Arnal, A. Pietrabissa, H. Cruickshank, A. Yun and E. Callejo, “Emulation testbed. Results and evaluation (D3000-5)”, Rapport LAAS No08542 Integrated Project 26950: SATSIX, Octobre 2008, 34p.
- C. Baudoin, F. Arnal, D. Barvaux, F. Goas, J. Nicol, G. Renker, G. Fairhurst, P. Berthou, B. Jacquemin, F. Nivor, T. Gayraud, A. Pietrabissa, S. Iyengar and L. Liang, “Emulation tested implementation. Validation document (D3000-2)”, Rapport LAAS No08530 Integrated Project 26950: SATSIX, Octobre 2008, 75p.
- C. Baudoin, D. Barvaux, F. Goas, J. Nicol, G. Renker, G. Fairhurst, P. Berthou, B. Jacquemin, F. Nivor, T. Gayraud, A. Pietrabissa and S. Iyengar, “Emulation tested implementation. Design document (D3000-1)”, Rapport LAAS No08528 Integrated Project 26950: SATSIX, Octobre 2008, 94p.
- F. Nivor, P. Berthou, T. Gayraud and B. Jacquemin, “About integrating DVB-S2/RCS satellite networks in NGN : a SIP based QoS Enhancing Proxy”, Rapport LAAS N°10017, Janvier 2010.
- C. Baudoin, L. Duquerroy, K. Leconte, F. Arnal, E. Callejo, A. Yun, T. Gayraud, P. Berthou, B. Jacquemin, A. Pietrabissa, G. Santoro, S. Iyengar and G. Fairhurst, “Satellite access architecture (D2000-2) ”, Rapport LAAS No07796 Integrated Project 26950: SATSIX, Janvier 2007, 156p.
- L. Fan, L. Liang, S. Iyengar, H. Cruickshank, Z. Sun, B. Cuesta, R.Castellot, L. Duquerroy, C. Baudoin, F. Arnal, P. Berthou, B. Jacquemin, T. Gayraud, S. Abdellatif, M. El Masri , G. Juanole, F. Rodriguez, M. Rossi, G. Fairhurst, A. Sathiaseelan, I. Melhus, R. MM, J.A. Guerra, M. Catalan de Domingo, A. Yun, E. Callejo and I. Moreno, “Network Architecture (D2000-1)”, Rapport LAAS No07795 Integrated Project 26950 : SATSIX, Janvier 2007, 194p.
- O. Alphand, P. Berthou, T. Gayraud and B. Jacquemin, “Architecture pour la mobilité dans un système satellite DVB-S/RCS”, Rapport LAAS N°06360, Mai 2006, 12p.