Architecture de coopération de réseaux sans fil

Davy Darche

UFR ESSTIN
École Doctorale IAEM Lorraine
DFD Automatique

Thèse

présentée pour l'obtention du

Doctorat de l'Université Henri Poincaré, Nancy 1

par

Davy Darche

Architecture de coopération de réseaux sans fil

Soutenue le 14 Juin 2006

Composition du jury

Rapporteurs :	K. Chen
	F. Krief
Examinateurs :	L. Toutain
	F. Lepage, directeur de thèse
	R. Kopp, tuteur en entreprise
	E. Gnaedinger, encadrant de thèse

Centre de Recherche en Automatique de Nancy
CRAN–UMR 7039 CNRS–UHP–INPL

A Ingrid

Remerciements

Je remercie sincèrement mes rapporteurs, Monsieur Ken Chen, Professeur à l'institut Galilée de Paris 13, et Madame Francine Krief, Professeur au LaBRI, pour leur travail de relecture et leurs remarques constructives.
Merci à Monsieur Laurent Toutain, Maître de conférences à l'ENSTB, pour avoir apporté son regard critique et examiné minutieusement ce travail.
J'adresse mes remerciements à Monsieur Francis Lepage, Professeur au CRAN, pour l'aide qu'il m'a apportée au cours de cette recherche en tant que directeur de thèse. Son intérêt et ses précieux conseils m'ont été d'un grand profit.
J'exprime toute mon amitié et mon estime à Monsieur René Kopp, responsable du service Architecture au centre de recherche de TDF Metz, pour m'avoir accueilli dans son service et m'avoir offert l'opportunité de réaliser ce travail de thèse. Je te remercie chaleureusement, René, pour la qualité de tes conseils et de ton encadrement durant ces trois dernières années.
J'offre mes sincères remerciements à Monsieur Eric Gnaedinger, Maître de conférences à l'ESSTIN, pour sa patience et sa disponibilité depuis de nombreuses années. Depuis bien avant que ne débute ce travail de thèse tu m'as toujours soutenu pour atteindre mes objectifs. Pour toute l'aide que tu m'as apportée, je te dit simplement : merci pour tout Eric.
J'adresse mes remerciements à toute l'équipe de TDF avec qui j'ai eu le plaisir de travailler. Je remercie particulièrement Monsieur Bertrand Mazières pour la pertinence de ses remarques et son esprit critique vis-à-vis de mon travail de thèse.
Merci à toute l'équipe du CRAN avec qui j'ai partagé la vie du laboratoire depuis mon DEA.
Je remercie Pavol Barger, pour sa patience et l'aide précieuse qu'il m'a apporté dans la modèlisation en réseaux de Petri.
Merci à mon vieux compère Sam pour ses encouragements et son amitié depuis de si nombreuses années.
Je remercie particulièrement toute ma famille, la famille Masson et tous mes proches pour leur soutien moral qu'ils m'auront fournis tout au long de la réalisation de ces travaux.
Enfin, merci à ma fiancée, Ingrid, pour tout son amour et à qui je dédie ce travail de thèse.

Table des matières

Préface

Cette thèse a été réalisée dans le cadre d'une convention CIFRE entre le centre de recherche de TDF, Télédiffusion de France, situé à Metz, et le CRAN, Centre de Recherche en Automatique de Nancy, laboratoire associé au CNRS. La démarche de cette thèse s'inscrit dans la résolution d'une problématique industrielle liée au métier de TDF à l'aide d'une approche théorique développée au CRAN.

« *L'information est une troisième dimension fondamentale de la matière, au-delà de la masse et de l'énergie* »

K.J. Boulding,
président de l'Académie des Sciences de New-York, 1952.

Dans cette préface nous allons tout d'abord présenter rapidement l'entreprise Télédiffusion De France (TDF) afin de définir plus précisément le contexte industriel dans lequel s'est déroulé ce travail de thèse. Ensuite nous aborderons succinctement les technologies de diffusion numérique qui sont un des cœurs de métier de TDF.

Le Groupe TDF

L'éclatement de l'Office de Radio Télévision Français (ORTF) constitua un tournant majeur dans l'histoire de la radiotélévision française. Sept organismes voient donc le jour : quatre sociétés nationales de programmes Radio France, TF1, A2, FR3 ; la Société Française de Production (SFP), chargée de la production, et deux établissements publics à caractère industriel et commercial : l'Institut National de l'Audiovisuel (INA) et TDF. Formé à partir du regroupement de la régie de diffusion et de la direction de l'action technique de l'ancien office, l'organisme chargé de la diffusion prend le nom de Télédiffusion De France. TDF se dote d'un nouveau centre de recherche dans le domaine de la radiodiffusion. Il va permettre de compléter le potentiel de recherche dont disposait l'établissement public au centre d'études et

de recherche d'Issy-les-Moulineaux (Cerim) et celui qu'il partageait avec le CNET au CCETT, à Rennes. Après avoir envisagé plusieurs sites, c'est à Metz que TDF décida, en avril 1984, d'implanter le nouveau centre, baptisé, centre d'études et de recherche de Lorraine (Cerlor). En 1992, incité par les pouvoirs publics à procéder à la délocalisation d'une partie des services parisiens, TDF propose de regrouper ses laboratoires du Cerim et du Cerlor. Baptisé, Centre d'études en Radiodiffusion et en Radiocommunications (TDF-C2R). C2R devient l'un des éléments clés de la politique de développement suivie par TDF. La convergence et l'évolution rapide des technologies audiovisuelles et télécoms montrent qu'aujourd'hui la recherche et l'ingénierie sont indissociables. La force de TDF est de disposer, dans ces domaines, des compétences et de l'expertise technique de 330 collaborateurs.
Forte de son expérience en matière de diffusion analogique, TDF contribue depuis plusieurs années à la Télévision Numérique Terrestre (TNT) une avancée technologique qui permet la diffusion du son et de l'image en qualité Digital Video Disc (DVD) d'une trentaine de services de télévision. Dès 1998, TDF a créé, en Bretagne et à Metz des plate-formes d'expérimentation de la télévision numérique terrestre en vraie grandeur.
TDF a lancé dès le 17 janvier 2005 la TNT en pré-déploiement depuis la tour Eiffel en conditions réelles de diffusion. Les activités de TDF s'étendent au-delà de la diffusion numérique ou analogique de contenu audiovisuel et se sont diversifiées dans des domaines tels que le transport intelligent ou la diffusion sur Internet. TDF participe également activement à de nombreux projets de recherche européens dans le domaine des réseaux de communication (DAIDALOS, CISMUNDUS, INSTINCT, ENTHRON...) Audiovisuel, télécoms, scientifique, réseaux et fréquences sont les cinq domaines d'excellence que TDF cultive au sein de sa direction technique pour ses clients, pour le développement de son activité et pour ses propres services en France et à l'étranger. Le but de cette thèse est de déterminer une architecture protocolaire valorisant l'intégration des divers supports de transmission liés aux métiers de TDF que sont les réseaux radios informatiques, télécoms et de diffusion audiovisuelle qui sont présentés dans la section suivante.

TDF, un acteur majeur dans le développement de DVB-T

Depuis sa conception en 1993, le projet Digital Video Broadcasting (DVB) [1] a démontré la valeur et la viabilité de la mise au point de standards ouverts dans le domaine de la diffusion numérique. Deux cent soixante compagnies contribuent

Standard DVB	*Modulation courante*	*Application*
DVB-C	64QAM	Cable
DVB-S	QPSK	Satellite
DVB-T	16QAM ou 64 QAM	Terrestre
DVB-H	16QAM ou 64 QAM	Terrestre (mobile)

TAB. 1 – Configurations courantes de DVB

actuellement à la poursuite des travaux dans divers modules, à l'amélioration des standards existants, et à la définition de nouvelles spécifications afin de répondre aux besoins d'un monde de diffusion numérique toujours en constante évolution. Digital Video Broadcasting Terrestrial (DVB-T) est le plus récent des systèmes DVB, après Digital Video Broadcasting Cable (DVB-C) destiné à la diffusion sur le câble, et Digital Video Broadcasting Satellite (DVB-S) utilisé pour la diffusion satellite. Reposant sur le Coded Othogonal Frequency Divisional Multiplexing (COFDM) [2],[3],[4] et le Quaternary Phase Shift Keying (QPSK), avec une modulation de 16QAM ou 64QAM, Quadrature Amplitude Modulation (QAM) [5],[6], DVB-T est un des moyens de diffusion numérique parmi les plus souples et les plus sophistiqués, disponibles aujourd'hui (Tab. 1). DVB-T permet au diffuseur d'augmenter la couverture des émetteurs tout en diminuant leurs puissances. DVB-T ne fut pas conçu pour la réception mobile, cependant de nombreuses expérimentations en environnement réel ont permis de mieux comprendre son fonctionnement et d'accroître ses performances en utilisant par exemple la diversité d'antenne. Il a été ainsi possible d'étendre les usages de DVB-T à des terminaux mobiles ou en réception indoor, ce qui était impossible auparavant avec d'autres moyens de diffusion numérique. Le nouveau standard Digital Video Broadcasting Handheld (DVB-H), a permis de résoudre l'épineux problème de la consommation d'énergie pour les terminaux mobiles, qui était particulièrement conséquente pour assurer une bonne réception DVB-T. DVB-H intègre un mécanisme de time slicing permettant au récepteur de se mettre en veille pendant les périodes d'inactivité et économisant ainsi environ 90% d'énergie. Un autre point clé est le développement de solutions d'IP datacasting, qui facilitera l'inter-opérabilité entre les réseaux de diffusion et les réseaux de télécommunication en se basant sur le protocole IP [7],[8]. L'usage de liens DVB pour le transport de données IP est donc un point prédominant dans ce travail de recherche. Par la suite nous emploierons le terme de « lien DVB »pour désigner une voie de diffusion unidirectionnelle utilisant le standard DVB-T ou DVB-H.

Introduction

Nous assitons aujourd'hui à trois manifestations majeures qui aboutissent à l'apparition du concept « d'Internet ambiant ».
Dans un permier temps, Internet est devenu un outil de communication incontournable tant pour des besoins professionnels, que pour des usages personnels. La convergence des services, comme la téléphonie, la diffusion vidéo, ou la vidéoconférence, vers un support de communication full IP font de celui-ci un média universel pour le transport de l'information.
D'autre part l'accroissement des capacités de traitement de l'information, la miniaturisation des équipements, et le développement des applications multimédia, intégrant la voix et/ou l'image ont entraîné l'apparition de nouveaux types de terminaux polyvalents. Ces nouveaux clients légers, comme les Personal Digital Assistant (PDA) ou les téléphones mobiles, peuvent désormais offrir des services audio, vidéo ou d'échanges de fichiers informatiques en situation de mobilité ou de nomadisme.
Enfin le développement et le déploiement des technologies sans fil comme l' Universal Mobile Telecommunications System (UMTS), le Wireless Fidelity (WiFi), ou DVB-T, fournissent une connectivité potentielle importante au réseau Internet pour un terminal multi-interfaces.
Cependant, dans un contexte de mobilité à travers des réseaux sans fil hétérogènes, le modèle TCP/IP actuel limite l'exploitation des ressources réseaux par le terminal. En effet, la continuité de service lors d'une commutation inter-technologique, la sélection d'un réseau adéquat pour un service donné, ou l'exploitation simultanée de plusieurs réseaux, sont autant de fonctionnalités qui ne sont pas assurées dans les architectures réseaux actuelles.
Les principales problématiques résident dans la définition d'agencements des ressources réseaux afin de satisfaire les contraintes relatives au service de l'application, et dans la définition de processus permettant leurs exécutions.
Dans ce mémoire de thèse, nous avons défini une architecture protocolaire offrant

des mécanismes de coopération de réseaux opérant ainsi une gestion optimisée des diverses ressources de communications disponibles pour un terminal multi-interfaces en situation de mobilité.

Le premier chapitre de cette thèse expose les phénomènes de convergence qui s'opèrent à différents niveaux et les problèmatiques liées à l'hétérogénéité dans un contexte de réseaux coopérants. Cette première partie aborde également le principe de coopération de réseaux et liste les quatres fonctionnalités que peut offrir une architecture coopérante.

Dans le chapitre suivant, nous présentons l'architecture UniDirectional Link Routing (UDLR), qui est un des exemples les plus connus de coopération de réseaux. Nous proposons également deux autres solutions fonctionnant à différents niveaux de la pile protocolaire. Une analyse de ces différentes approches nous permet de préciser les caractéristiques d'une solution de coopération de réseaux d'un point de vue plus général.

La chapitre trois, détaille les protocoles qui apparaissent comme les meilleurs candidats pour constituer une pile protocolaire offrant des moyens de coopération de réseaux. A la fin de ce chapitre nous précisons les fonctionnalités d'une architecture protocolaire coopérante reposant sur les protocoles séléctionnés.

Dans le quatrième chapitre, nous décrivons une classification simple des applications à travers des contraintes caractérisant le type de service fourni. Nous déterminons également une liste de critères en partie mesurés et en partie spécifiés et indépendants des technologies sous-jacentes. A partir des contraintes applicatives et des critères caractérisant les réseaux d'accès de chaque interface, nous proposons une nouvelle couche de gestion des moyens de communications mettant en œuvre les principes de coopération de réseaux.

Le dernier chapitre montre une description SDLde la signalisation de bout en bout relative aux modules de coopération de réseaux ainsi qu'une représentation en réseaux de Petri du fonctionnement global du processus de coopération. Finalement, deux scénarii de coopérations de réseaux permettent d'évaluer notre modèle à l'aide d'une expérimentation et de simulations.

Chapitre 1

Contexte et problématique

1.1 Une ère de l'information numérique

Depuis quelques années, notre société subit de profondes mutations, conséquences directes de l'évolution des technologies de l'information. L'émergence de cette nouvelle société, dite « de l'information », a été possible grâce à deux principaux phénomènes : la numérisation de l'information et la transmission numérique .
Si chacun de ces phénomènes, pris indépendamment, apporte quelques avantages notables, la conjonction des deux a engendré une véritable révolution dans de nombreux domaines d'activité, comme l'industrie (entreprise en réseaux) ou les marchés financiers (transactions électroniques).
Le réseau Internet est la base sur laquelle repose toutes ces innovations, et la capacité et la rapidité d'accès à Internet peut s'avérer être un facteur déterminant dans certaines situations.
Pour l'avenir, deux grandes tendances semblent se dessiner sur lesquelles s'accordent les différentes visions [9] : une augmentation significative des débits et l'omniprésence du réseau qui devient pervasif.
Avec l'accroissement des débits, il devient possible d'offrir des services gourmands en terme de ressources réseaux comme la Video on Demand (VoD) ou la visio-conférence.
Les différents réseaux de transmission offrant une connectivité à Internet sont de plus en plus nombreux, et si un type de réseau particulier n'offre pas une couverture totale sur un territoire donné, celle-ci pourrait être améliorée par la complémentarité ou la *coopération* de ces réseaux.
De plus la façon de consommer l'information tend de plus en plus à devenir nomade ou mobile et les services sont maintenant souvent destinés à des terminaux portables.

Ces trois facteurs induisent de nombreuses problématiques pour une exploitation optimisée des ressources réseaux disponibles pour satisfaire un service de communication donné. C'est dans cette problématique de coopération de réseaux, dans un contexte de mobilité continue, que s'inscrit ce travail de thèse.

1.2 La convergence des traitements de l'information

La convergence, le changement numérique de la communication et de l'information, produit un nouveau genre d'interchangeabilité et d'interconnectivité parmi différents types de supports. Les principales conséquences de la convergence sont :

- l'intégration de différentes formes de communication : textes, sons, vidéos, images, à travers une seule application,
- un degré croissant de chevauchement dans les fonctions qui peuvent être exécutées par différents réseaux de communications,
- une croissance de l'inter-activité et de l'inter-opérabilité de différents réseaux et instruments de l'information.

Deux phénomènes majeurs ont rendu possible l'application du concept de convergence numérique. Premièrement, la progression importante, ces dernières années, des capacités de traitement et de stockage de l'information. Deuxièmement, l'établissement de standards ouverts, auxquels participent les principaux industriels fournisseurs d'équipements (CISCO, HP, IBM...), et qui permettent l'interopérabilité des processus communiquants et les échanges d'informations [10]. Si le concept de convergence des réseaux téléphonique, informatique, et audiovisuel vers un support unique n'est pas récent, les premières études remontent à une dizaine d'année [11][12], leurs exploitations commerciales commencent seulement à apparaître aujourd'hui. Avec les avancées dans le domaine de l'Internet et des télécommunications, l'accroissement constant des réseaux hauts débits, et la baisse des coûts des solutions réseaux, les grandes comme les petites entreprises ont besoin de revoir leurs moyens d'offrir des services innovants ou plus compétitifs. Excepté pour les réseaux de communications de données informatiques, qui peuvent transporter le contenu de différentes sortes d'applications, comme la messagerie électronique, le web, et le transfert de fichiers, sur un réseau local ou sur Internet, les entreprises possèdent différents moyens de communications pour fournir leurs divers services. Cependant avec l'émergence de nouvelles technologies, comme la voix sur IP ou Voice over IP (VoIP), il est maintenant possible de rattacher les services de téléphonie sur les infrastructures de réseaux

existantes, en tant que nouveaux services de communications.
Dans cette nouvelle approche, nous considérons l'infrastructure IP de façon plus large qu'un simple support de communication de données électroniques (web, mail, transfert de fichier...). Cela offre des opportunités plus vastes comme le transport de vidéo, de voix, et de services interactifs pouvant se rattacher à des infrastructures IP. Dans un contexte de coopération de réseaux, nous avons donc choisi le protocole IP comme support universel pour l'acheminement des informations, de quelque nature quelles soient (audio, vidéo, texte...). Nous allons rappeler brièvement l'historique et les principes fondamentaux du protocole internet. Par la suite nous ne considérerons que la version six du protocole, qui sera détaillée dans la section 3.2 page 42.

1.2.1 Les origines du protocole IP

Les origines du protocole IP remontent à 1957 avec la création de l'Advanced Research Project Agency (ARPA) et la proposition en 1966 de Robert Taylor de créer un réseau de communication capable de résister à des attaques nucléaires [13]. Le concept fondamental, sur lequel repose le protocole IP, consiste à découper les informations à transmettre en paquets au niveau de l'émetteur, de transmettre les paquets à travers un réseau constitué de plusieurs nœuds, et de reconstruire le message original au niveau du récepteur. Le protocole IP peut être qualifié de robuste, puisque les paquets peuvent potentiellement emprunter différents chemins dans le réseau pour atteindre leur destination. Le protocole TCP [14] gère le contrôle de flux et la retransmission des paquets en cas d'erreur, assurant ainsi d'une part la stabilité du réseau, et d'autre part la fiabilité de la communication.
Le protocole IP associé au protocole TCP offre donc un service de transport de données fiable et robuste à travers un réseau maillé. Deux fonctionnalités importantes sont intégrées dans le protocole IP, l'adressage et la fragmentation. La fragmentation permet de découper et de ré-assembler les messages transmis. L'adressage des paquets IP, qui comprend l'adresse de la source du paquet et l'adresse de destination, est utilisé par les routeurs du réseau afin d'aiguiller correctement le paquet vers sa destination.
Dans le protocole IP, la notion de circuit de communication est absente, et les différents paquets IP d'un même message peuvent transiter par des chemins différents en fonction de la charge des liens ou des défaillances des routeurs. Un circuit de communication est recréé de façon logique au niveau de la couche TCP.
Ces différentes caractéristiques en font un protocole relativement simple, mais particulièrement robuste, ce qui explique en partie le succès de ce protocole sur lequel

repose l'Internet tel que nous le connaissons aujourd'hui.
La principale difficulté induite par le routage des paquets IP est le synchronisme car le temps de traversée d'un datagramme est lié, à un moment donné, à la charge du réseau. Par conséquent, le choix du protocole IP comme support de communication unique pour des services tels que la téléphonie ou la diffusion vidéo ne semblait pas évident il y a encore quelques années.
Cependant l'introduction de qualité de service ou Quality of Service (QoS), la conception de protocoles de signalisation complémentaires du protocole IP (Resource ReSerVation Protocol (RSVP) [15]), l'amélioration des algorithmes de routage en fonction des types de flux de données (Differentiated Services (DiffServ) [16]), et l'accroissement de la bande passante dans les cœurs de réseaux avec la fibre optique, ont permis de palier les lacunes d'IP et positionne celui-ci comme le meilleur candidat en tant que support de communication universel pour tous types de services.
La nouvelle version du protocole IP, IPv6, devrait encore optimiser les communications IP en intégrant de façon native des fonctionnalités de qualité de service, de sécurité ou encore de mobilité (cf. § 3.2 p. 42).

1.2.2 La convergence des réseaux

La convergence des réseaux de voix, de vidéo et de données vers un unique support IP entraîne une baisse des coûts d'exploitation, de maintenance et d'administration des infrastructures réseaux pour les entreprises. La convergence vers le support IP fournit une base solide pour le développement de nouvelles applications basées sur les technologies IP (contrôle de bande passante, authentification, chiffrement...).
Cette notion de convergence apparaît à différentes échelles des réseaux de communication. Aujourd'hui les cœurs de réseaux des fournisseurs d'accès fonctionnent en tout IP et permettent d'offrir aux particuliers des services de VoIP, de Television over IP (TVoIP) et d'accès Internet haut débit à travers une seule connexion.
Avec l'apparition des réseaux de télécommunication de troisième génération, les opérateurs de téléphonie mobile proposent des offres similaires vers des terminaux portables. Bien qu'avec l'UMTS les communications vocales empruntent toujours un réseau commuté, certains opérateurs outre-atlantique propose déjà des solutions de VoIP sur des téléphones WiFi.
Une conséquence directe et facilement remarquable est la polyvalence des nouveaux terminaux mobiles ou fixes. En effet un terminal classique, (ordinateur, téléphone mobile...) est aujourd'hui capable de recevoir, d'émettre et de capturer de la vidéo, des images, des sons, de la musique, et des données à travers une seule interface de

quelque type que ce soit.
Le protocole IP permet ainsi d'offrir une multitude de services multimédias sur un support unique. Dans le modèle TCP/IP, la couche IP fournit une abstraction des couches protocolaires sous-jacentes. Par conséquent, un même service multimédia reposant sur IP, peut être fourni au terminal par des réseaux sous-jacents différents (WiFi, ethernet, Universal Mobile Telecommunications System (UMTS), DVB-T).

1.2.3 La convergence des terminaux

Avec la numérisation de l'information et le développement des capacités multimédia des ordinateurs, on voit apparaître aujourd'hui des terminaux pouvant faire office de téléviseur, de téléphone, et de client Internet. Couplé avec une convergence entre les services fixes et mobiles (téléphonie, diffusion TV), ce phénomène a engendré l'apparition de nouveaux terminaux mobiles intégrant diverses fonctionnalités multimédias. Un ordinateur portable peut être utilisé comme lecteur de DVD, ou récepteur de télévision numérique, et il est possible de naviguer sur Internet ou de communiquer par courrier électronique avec son téléphone portable. Ordinateurs portables, téléphones, PDA, tous ces terminaux convergent progressivement vers un terminal unique, capable d'assurer un minimum de traitement informatique, de fournir des services de téléphonie ou de visiophonie et de pouvoir recevoir des diffusions TV.
Plusieurs approches existent aujourd'hui, l'une consistant à utiliser le même réseau pour fournir plusieurs services, et l'autre proposant d'utiliser des terminaux multimodaux, possédant plusieurs interfaces adaptées à chaque service. A l'instar du triple play, disponible sur les réseaux ADSL, certains opérateurs UMTS proposent aujourd'hui des services de téléphonie, de diffusion TV et d'accès Internet vers des terminaux mobiles. Inversement, certains constructeurs intègrent plusieurs interfaces complémentaires, comme GPRS/UMTS et WiFi, ou GPRS/UMTS et DVB-H. Cette dernière approche comporte certes des contraintes d'intégration de ces différentes technologies, mais apportent deux avantages majeurs sur le plan des services :

- Le choix du réseau le mieux adapté au service souhaité (par exemple UMTS pour les données et DVB-H pour la réception TV),
- la redondance potentielle de connectivité pour l'accès à certains services (accès à Internet ou communications vocales par UMTS ou WiFi).

Des projets comme l'Unlicensed Mobile Access (UMA) [17] vise à fournir des accès à des services mobiles Global System for Mobile (GSM) et General Packet Radio Service (GPRS) à travers des réseaux radio tel que le WiFi ou le bluetooth [18]. D'autres technologies, comme la radio logicielle ou Software Defined Radio (SDR)

permettent à un composant matériel, responsable d'un traitement de signal, de se reconfigurer afin de s'adapter à un nouveau type de réseaux. Une même interface matérielle pourrait ainsi se connecter à différents réseaux comme UMTS, DVB-T, bluetooth...
Il apparaît aujourd'hui clairement que l'usage des différents réseaux de communication comme l'UMTS, DVB-T et le WiFi sont de moins en moins spécifiques aux services pour lesquels ils ont été conçus à l'origine.

1.3 Vers une coopération des réseaux de communication

Le concept de coopération de réseau englobe plusieurs notions qu'il est important de préciser. Les trois grands types de réseaux, que sont les réseaux de téléphonie, les réseaux informatiques (Internet), et les réseaux de diffusion, possèdent tous des caractéristiques propres aux types de services qu'ils transportent.
Les réseaux de téléphonie ont été conçus pour répondre aux besoins d'applications possédant de fortes contraintes temporelles, comme le transport de la voix. Ces réseaux intègrent également des moyens de facturation très efficaces. En contrepartie, ils nécessitent une forte signalisation et une gestion relativement rigide des ressources réseaux en termes de bande passante.
La souplesse des réseaux informatiques reposant sur IP, fait de ceux-ci un moyen de communication idéal pour le transport de données. L'absence de circuit de communication et le concept de transport en mode paquet associés à des architectures hiérarchiques et des protocoles de routage évolués, optimisent considérablement les ressources en bande passante nécessaires dans les cœurs de réseaux.
Les réseaux de diffusion numérique, comme DVB-S ou DVB-T, peuvent fournir un service identique à plusieurs milliers d'usagers avec un parfait synchronisme. Cependant, le service élémentaire fourni par ces réseaux de diffusion ne propose aucune inter-activité.
La demande de nouveaux services multimédia personnalisés, interactifs, et de haute qualité nécessiterait une adaptation importante des divers réseaux actuellement déployés. Une alternative consiste à faire coopérer ces différents réseaux dans une architecture hybride, en conservant les points forts de chaque réseau. Cette coopération peut être réalisée à différents niveaux : service, applicatif, ou réseaux de transmission.

1.3.1 Une coopération au niveau service

Un embryon de coopération de réseaux est apparu avec la participation des auditeurs de radio ou des téléspectateurs à des émissions diffusées en direct. Ceux-ci utilisent le réseau téléphonique pour transmettre leurs avis, questions ou réponses à l'animateur, qui sont rediffusés en quasi-simultanéité sur le lien de radio-diffusion. Cette coopération, rudimentaire, est assurée par le présentateur de l'émission mais présente déjà quelques problèmes techniques. En effet, il est fréquent d'entendre un animateur d'émissions radio-diffusées, ou télé-diffusées, demander à l'auditeur ou au téléspectateur de réduire le volume de son poste de réception afin d'éviter l'effet Larsen dû au bouclage du son.
Cette coopération entre un réseau de diffusion et un réseau téléphonique se retrouve également dans les systèmes de vote par appel téléphonique ou par SMS dans certaines émissions de variétés. Un autre exemple de coopération entre un réseau téléphonique et le réseau Internet consiste à utiliser une connexion téléphonique pour la facturation, et le réseau Internet pour l'accès aux données.
Ces divers exemples de coopérations, certes limitées, entre plusieurs types de réseaux montre bien que l'idée est présente depuis longtemps, et souligne une caractéristique importante de la coopération de réseaux, qui est d'utiliser les avantages de chaque sorte de réseaux, pour fournir un nouveau type de services (inter-activité de la voix et diffusion TV, ou facturation téléphonique et transport de données IP).

1.3.2 Une coopération au niveau applicatif

Une intégration plus profonde de la coopération de réseaux place celle-ci au niveau de la couche application. On retrouve ce type de coopération dans les systèmes de diffusion satellite payants, dans lesquels les récepteurs possèdent une connexion téléphonique en plus de l'interface de réception DVB-S.
Le projet européen CISMUNDUS, montre la réalisation d'une application de vidéo à la demande, reposant sur l'utilisation d'un réseau de téléphonie mobile, GPRS ou UMTS, pour la requête et la facturation, et un réseau DVB-T pour la diffusion de vidéos vers le terminal client [19].
Dans les deux cas de figure précédents, chaque réseau constitue un canal de communication indépendant entre l'application serveur et l'application client. Le lien téléphonique est utilisé pour la signalisation et/ou la facturation, et le lien de diffusion pour le transport de la vidéo vers le terminal client. Les usages des différents réseaux sont intégrés au sein même de l'application.

1.3.3 Une coopération au niveau des réseaux de transmission

L'intégration de la coopération de réseaux peut également être implémentée au sein de la communication elle-même en séparant la voie de transmission ascendante de la voie de transmission descendante.
Ces réseaux coopérants comprennent généralement un lien de diffusion haut débit pour acheminer les données vers l'utilisateur, et un réseau IP, fourni par une connexion Réseau Téléphonique Commuté (RTC) comme voie de retour. Ces architectures réseaux sont fortement asymétriques et orientent naturellement son usage vers le téléchargement de données. Dans le protocole UDLR, la coopération de réseaux est implémentée en modifiant les modules systèmes des interfaces unidirectionnelles au niveau de la couche liaison de données et de la couche réseau.
L'axe de recherche de cette thèse m'a conduit à développer et à évaluer d'autres solutions de coopération de réseaux qui permettent d'opérer directement au niveau de la couche réseau et même de la couche transport. Une première solution mise en œuvre au niveau IP est décrite dans la section 2.2.1 . Une seconde architecture s'appuyant sur le protocole Stream Control Transmission Protocol (SCTP) est détaillée dans la section 2.2.2.
Le principe de cette coopération de réseaux, au sein de la communication elle-même, est identique aux exemples des niveaux applicatif et services cités précédemment. Les atouts de chaque type de réseaux, large bande passante et robustesse de la transmission du lien de diffusion pour l'envoi des paquets de données, couplés à la bidirectionnalité d'un réseau téléphonique bas débit pour le renvoi des accusés de réception, sont associés afin de fournir un nouveau service : un accès haut débit, asymétrique, et bidirectionnel.

1.4 Les problématiques de la coopération de réseaux

Le modèle TCP/IP actuel, initialement prévu pour des hôtes fixes possèdant une seule interface, est reconnu comme un moyen de communication robuste et efficace. Dans un environnement offrant des connectivités multiples, via des liens de communication hétérogènes pour un hôte en situation de mobilité, ce modèle présente de nombreuses insuffisances et peut limiter les performances des communications.

1.4.1 Des réseaux hétérogènes

L'UMTS, le WIFI et DVB-T sont les technologies radio les plus représentatives des trois grands types de réseaux. Ces trois réseaux possèdent des paramètres mé-

Paramètres	*UMTS*	*WIFI*	*DVB*
Délai	240ms	2-3ms	50ms
Gigue	20ms	1-2ms	10ms
Bande passante (Ul/Dl)	60kbit/60kbit	5-40Mbit/5-40Mbit	24Mbit
Taux d'erreur	0%	0.01%	$< 10^{-8}$

TAB. 1.1 – Paramètres réseaux, mesures empiriques

trologiques, délai, gigue, bande passante et taux d'erreur, qui différent fortement les uns des autres (Tab. 1.1).

Afin de caractériser ces réseaux, plusieurs campagnes de mesures on été effectuées avec différents opérateurs réseaux (Orange, SFR, Sonera, Radiolinja pour le GPRS ou l'UMTS, et TDF et Digita pour DVB-T) ou différentes technologies (802.11b, 802.11a, 802.11g pour le WiFi). Les résultats rassemblés dans le tableau 1.1 sont les valeurs moyennes les plus représentatives des différents attributs de chaque famille de réseaux. En considérant l'ensemble de ces technologies comme une seule entité, celles-ci peuvent être perçues comme une ressource réseau unique et fortement hétérogène, offrant une connectivité multiple à un terminal multi-interfaces.

Dans ces conditions, plusieurs communications peuvent être établies en empruntant chacune des interfaces différentes, et donc utiliser des réseaux possédant des paramètres physiques distincts. Cette hétérogénéité peut se révéler problématique pour les piles protocolaire actuellement utilisées avec le protocole IP.

Les protocoles de couche transport aujourd'hui les plus répandus, TCP et UDP, peuvent être adaptés à des caractéristiques réseaux particulières (augmentation des fenêtres de congestion TCP pour les réseaux ayant un délai important, TCP Delayed-ACK pour les réseaux asymétriques, réduction de la taille des paquets UDP pour des réseaux possédant un taux d'erreur de transmission important...).

Un changement dans ces caractéristiques nécessite une intervention de l'utilisateur afin d'adapter les paramètres de la couche transport. Dans un contexte de coopération de réseaux, où les divers réseaux ne forment plus qu'une seule ressource hétérogène, grâce à des mécanismes de combinaison ou d'agrégation (cf. § 3.5, p. 62), les performances des protocoles de transport peuvent être fortement dégradées.

De plus, la combinaison ou l'agrégation de réseaux nécessite certaines fois des architectures spécifiques et particulièrement la présence d'un routeur d'interconnexion entre les trois types de réseaux. L'ensemble des communications transitant par ce noeud, celui-ci représente un point de défaillance important de l'architecture, et peut rendre le routage des paquets sous-optimal.

1.4.2 Des interfaces multiples

Des terminaux possédant plusieurs interfaces de communication sont de plus en plus répandus. Cette intégration de plusieurs interfaces dans une entité unique est directement liée au phénomène de convergence des moyens de communication, comme la téléphonie, la diffusion TV, ou Internet. La connectivité potentielle offerte par les différents réseaux disponibles apporte quatre fonctionnalités importantes :

- la redondance,
- l'aggrégation,
- la sélection,
- la combinaison.

La redondance des liens de communication est actuellement sous-exploitée. Pour illustrer cet exemple, prenons un terminal possédant deux connexions IP chacune sur une interface différente. Le terminal initie une communication TCP/IP, par exemple un téléchargement de fichier, sur la première connexion. La perte de cette connexion entraîne un arrêt de la communication. La connexion TCP/IP ouverte par l'application est incapable de basculer sur la seconde interface. Le téléchargement est donc interrompu après l'expiration d'un timer. Des mécanismes de reprise sont couramment utilisés afin de poursuivre le service à partir de son point d'interruption. Cependant ces mécanismes sont implémentés au niveau applicatif, ce qui entraîne une durée importante avant la reprise du service, et nécessite l'initialisation d'une nouvelle communication sur la seconde interface.
Plusieurs technologies permettent de répartir un flux IP sur plusieurs interfaces physiques (load balancing, ieee802.3ad). Il est ainsi possible de cumuler la bande passante disponible de plusieurs interfaces réseaux, et d'exploiter l'ensemble comme une seule ressource de connectivité. Cependant ces mécanismes opèrent au niveau de la couche liaison de données ou nécessitent des équipements spécifiques, ce qui limite leur utilisation aux réseaux locaux ou à la connexion de serveurs. Le modèle TCP/IP actuel limite donc l'exploitation des ressources réseaux disponibles pour un terminal multi-interfaces.
La présence de plusieurs interfaces de communication devrait offrir la possibilité de choisir le réseau d'accès en fonction de critères relatifs au service souhaité. De même la combinaison de différents réseaux pourrait former une nouvelle ressource exploitable par des services, qui ne pourrait être fourni par aucun réseau pris indépendamment. Plusieurs exemples de combinaison de réseaux, ou réseaux hybrides, seront détaillés par la suite (UDLR, Hybrid Network Interconnection System (HNIS), cf. § 2, p. 13).
Si la robustesse du modèle TCP/IP n'est plus à démontrer, la rigidité de celui-ci,

due à son ancienneté, peut apparaître aujourd'hui comme une limitation en termes de fonctionnalité et de performances pour des terminaux multi-accès. Le modèle TCP/IP ne permet pas de fournir, de façon native, les quatre principales fonctionnalités de la coopération de réseaux que sont la redondance, l'agrégation, la sélection et la combinaison (cf. § 3.5). La mise en place d'architectures réseaux dédiées (HNIS, UDLR) peut assurer une coopération de réseaux simple, réservée à des usages spécifiques.

1.4.3 La mobilité des réseaux

La notion de mobilité IP, peut être définie pour un terminal comme sa capacité à changer de réseau d'accès IP sans interruption des communications en cours et à conserver son accessibilité vis-à-vis d'hôtes distants quelle que soit sa localisation. Pour des terminaux multi-accès, deux types de mobilité IP peuvent être envisagé :

- un changement de réseau IP afin de préserver les communications en cours,
- un changement de réseau IP afin d'améliorer les performances des services fournis au niveau des couches supérieures.

Les protocoles utilisés aujourd'hui dans Internet n'offre aucun service de mobilité. Par contre, la prochaine version de protocole Internet, IPv6, offre des mécanismes de mobilité définis dans la RFC 3775 [20]. IPv6 permet donc à un terminal de rester joignable avec la même adresse IPv6, appelée adresse mère, quelle que soit l'adresse temporaire fournie par le réseau IPv6 d'accueil du terminal (cf. § 3.2.1, p. 45).
Cependant, ces fonctionnalités se limitent à assurer la connectivité IPv6 et le protocole IPv6 est incapable de fournir des critères de sélection pour déterminer le réseau le plus adéquat aux services utilisés au niveau des couches supérieures.
Les grandes évolutions des technologies de l'information et la convergence qui s'opère aujourd'hui aussi bien au niveau des traitements de l'information que des réseaux et des terminaux introduisent de nouvelles problématiques qui ne peuvent être résolues par le paradigme actuel. Dans un contexte de terminal multi-interfaces en situation de mobilité, un processus de coopération peut apparaître comme une solution en tirant profit des diverses ressources réseaux disponibles afin d'assurer le service définit par la couche applicative.
Dans le chapitre suivant, nous allons étudier une architecture de coopération de réseaux relativement courante, UDLR, consistant à combiner un réseau unidirectionnel avec un réseau tiers en guise de voie de retour. Deux solutions alternatives, que j'ai développée, sont également présentées montrant que cette coopération de réseaux peut être réalisée à différents niveaux de la pile protocolaire.

Chapitre 2

Différentes architectures de coopération de réseaux

Ce chapitre présente trois solutions de coopération de réseaux comprenant chacune une approche différente et permettant de fournir un service de transmission de données sur un lien DVB-T avec une voie de retour alternative. Dans un premier temps nous présentons la solution existante UDLR qui peut être qualifiée de standard. Ensuite nous analysons deux nouvelles solutions HNIS et SCTP Variable Ack Rate (SCTPVAR) que j'ai développées dans le cadre de cette thèse. Enfin, une synthèse comparative de ces trois propositions précise l'approche retenue pour la défintion d'une nouvelle architecture protocolaire.

2.1 Une solution standard

2.1.1 Le protocole UDLR

Présentation

Le protocole UDLR est le résultat de travaux de recherche menés par l'équipe du projet Rodéo/Planète de l'INRIA, un groupe de travail (UDLR) fut créé à l'IETF en 2001 [21].
Le protocole UDLR est un mécanisme standard qui ouvre la bi-directionnalité du canal satellitaire en utilisant, par exemple, une voie de retour terrestre (modem téléphonique) tout en supportant l'ensemble des applications du standard Internet (IP, Transmission Control Protocol (TCP), User Datagram Protocol (UDP)...). La plupart des protocoles de routage et de communication d'Internet ont été créés et optimisés pour des liens symétriques et bidirectionnels. De nouveaux médias de com-

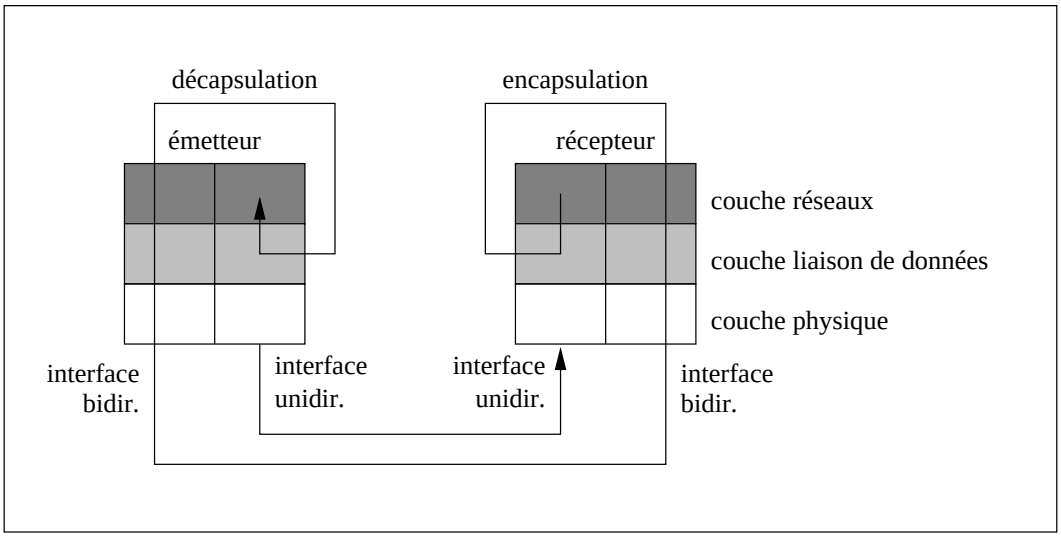

FIG. 2.1 – Mécanismes d'encapsulation UDLR

munication, comme les satellites de diffusion ou les émetteurs de télévision numérique terrestre ont été proposés pour fournir des accès Internet haut débit. Cependant, si ces médias possèdent une bande passante importante, ils sont dépourvus de voie de retour. Le protocole UDLR permet de coupler un lien de diffusion unidirectionnel avec un second réseau, utilisé comme voie de retour, et rend transparent, au niveau IP, l'usage de médias différents pour l'envoi et la réception de données.
UDLR nécessite une voie de retour bidirectionnelle et utilise des mécanismes de tunnel et d'encapsulation sur la voie de retour. Le protocole UDLR propose d'émuler une zone de broadcast. Ceci permet par la suite de déployer rapidement de nombreux protocoles de niveaux supérieurs sans apporter de modifications.

L'encapsulation UDLR

Le tunnel utilisé sur la voie de retour opère au niveau de la couche liaison de données. Ainsi au niveau de la couche réseau, le lien unidirectionnel apparaît comme un lien bidirectionnel.
Lorsque des données doivent être émises sur l'interface unidirectionnelle, celles-ci sont encapsulées dans une trame de niveau liaison de données, avec l'adresse matérielle de l'interface unidirectionnelle. L'interface ne pouvant pas émettre de données, celles-ci sont traitées par les mécanismes de tunnel. Les paquets sont alors encapsulés dans une trame IP ayant l'adresse IP de l'interface bidirectionnelle comme adresse source,

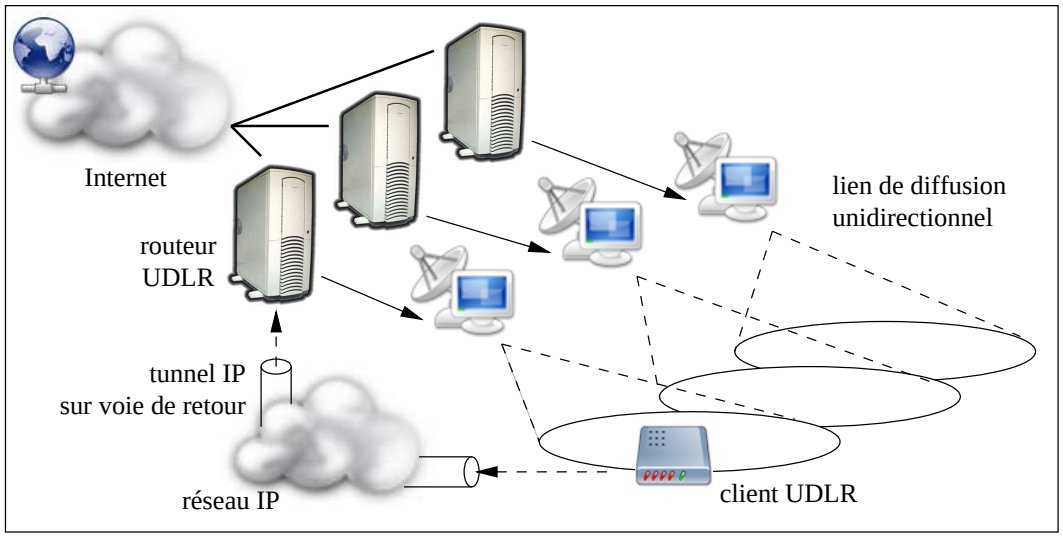

FIG. 2.2 – Architecture UDLR

et l'adresse d'un routeur UDLR comme adresse de destination (Fig. 2.1).
Si l'adresse MAC de destination est celle d'une interface du routeur UDLR, le paquet tunnel est envoyé au routeur UDLR. Sinon (par exemple l'adresse MAC est de type broadcast ou multicast) la destination du paquet tunnel est un routeur UDLR par défaut. Les paquets sont alors décapsulés en arrivant au routeur UDLR, et sont ensuite retransmis vers leurs destinations.

Le routage UDLR

Chaque routeur UDLR maintient une liste des interfaces unidirectionnelles des autres routeurs UDLR présents dans le domaine de broadcast. Cette liste est configurée manuellement au niveau du routeur UDLR, impliquant que le nombre de routeurs UDLR soit relativement restreint. Les routeurs UDLR peuvent utiliser trois modes différents pour transmettre des données sur leur lien unidirectionnel selon la destination de la trame liaison de données.

- Si l'adresse MAC est celle d'une interface de réception connectée au lien unidirectionnel, la trame est émise sur le lien unidirectionnel,
- si l'adresse MAC est celle d'une interface d'émission unidirectionnelle, la trame est traitée par les mécanismes de tunnel,
- si l'adresse MAC est de type broadcast ou multicast, elle est retransmise en copie à tous les autres routeurs UDLR.

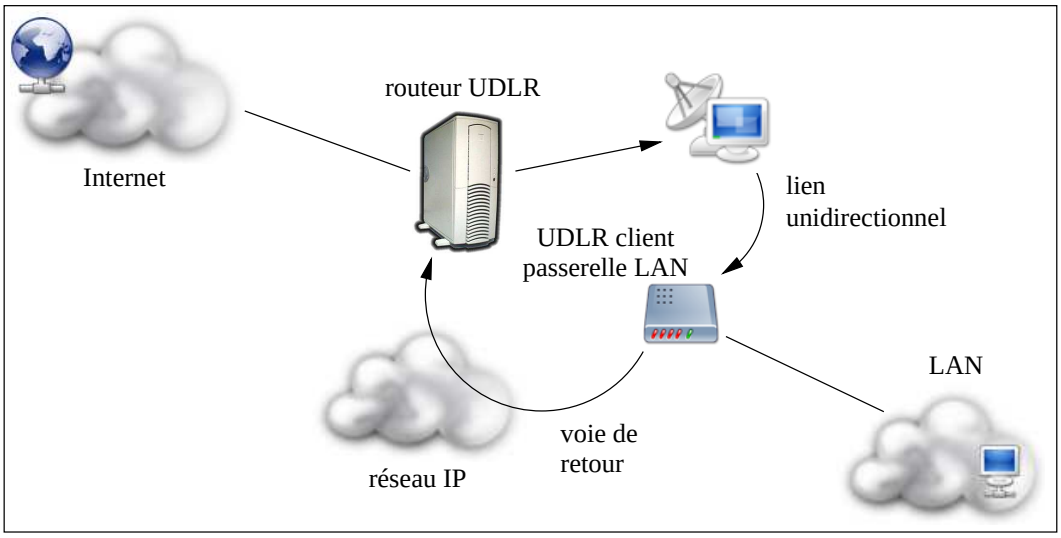

FIG. 2.3 – Architecture UDLR courante

Les routeurs UDLR réceptionnent les paquets encapsulés sur les extrémités des tunnels. Ces datagrammes reçus par l'interface bidirectionnelle passent ensuite par le processus de décapsulation. La décapsulation apporte des informations du niveau liaison de données du paquet originel, puisque celles-ci ne sont pas altérées lors de la traversée du tunnel. La commutation des paquets au niveau du routeur UDLR dépend de l'adresse MAC de destination :

- Si l'adresse MAC de destination est celle de l'interface d'émission unidirectionnelle, le paquet est délivré localement comme s'il provenait de l'interface elle-même,
- si l'adresse MAC de destination est celle d'un client UDLR, le paquet est transmis sur le lien unidirectionnel approprié,
- si l'adresse de destination est de type multicast ou broadcast, le routeur détermine si il est sélectionné comme routeur par défaut pour retransmettre le paquet. Si le routeur est désigné par défaut, il délivre le paquet localement, sur le lien unidirectionnel et aux autres routeurs. Sinon, le paquet est simplement délivré localement.

Le protocole DTCP

Les routeurs et les récepteurs UDLR doivent connaître l'ensemble des routeurs UDLR du domaine de broadcast afin de pouvoir gérer les mécanismes des tunnels IP.

La configuration sur les routeurs UDLR est manuelle, puisque le nombre de routeurs UDLR doit être relativement réduit. En revanche, la configuration des clients est automatique, puisque leur nombre peut être considérable. Pour cela UDLR utilise le protocole Dynamic Tunnel Configuration Protocol (DTCP) afin que les récepteurs puissent découvrir dynamiquement les routeurs UDLR présents, et maintenir une liste des points de terminaison des tunnels actifs. Le protocole DTCP envoie régulièrement des messages d'annonce (DTCP HELLO) sur le lien unidirectionnel comprenant, par exemple, la liste des routeurs UDLR, des interfaces unidirectionnelles, le type de tunnel...
Le protocole DTCP fonctionne au dessus d'UDP, et utilise une adresse multicast prédéfinie pour la diffusion des messages HELLO.
Les tunnels utilisés dans UDLR reposent sur le protocole Generic Routing Encapsulation (GRE) et l'adresse multicast 224.0.0.36, ainsi que le port UDP 652, ont été reservés auprès de l'Internet Assigned Numbers Authority (IANA) pour la diffusion des messages d'annonce DTCP. Le protocole UDLR offre donc une solution de coopération efficace et robuste couvrant la totalité des types de trafic IP (unicast, multicast et broadcast) puisque celui-ci opére la coopération de réseaux au niveau de la couche liaison de données. Cependant les modifications que nécessite UDLR, au niveau du système d'exploitation, entraîne une certaine rigidité, limitant généralement son usage client en tant que routeur d'accès pour de petits réseaux locaux plutôt que sur des terminaux utilisateur proprement dit.

2.2 Deux nouvelles propositions

2.2.1 L'architecture HNIS : Hybrid Network Interconnection System

Le système HNIS est une solution de coopération de réseaux que nous avons développée au sein de TDF pour un usage spécifique de coopération entre un réseau de diffusion DVB-T et un réseau téléphonique GPRS [22]. Dans une architecture de réseaux coopérants, ou architecture hybride, un terminal utilise des voies de transmission différentes pour envoyer et recevoir des données. DVB-T, qui est un réseau de diffusion robuste, unidirectionnel et possédant une large bande passante, est utilisé pour transmettre des données à destination du terminal, alors que le réseau GPRS, qui est un réseau de téléphonie mobile, bidirectionnel, avec une faible bande passante et un taux d'erreurs de transmission conséquent, est utilisé en tant que voie de retour. Nous avons conçu ce système afin de fournir un accès Internet haut débit, sans fil,

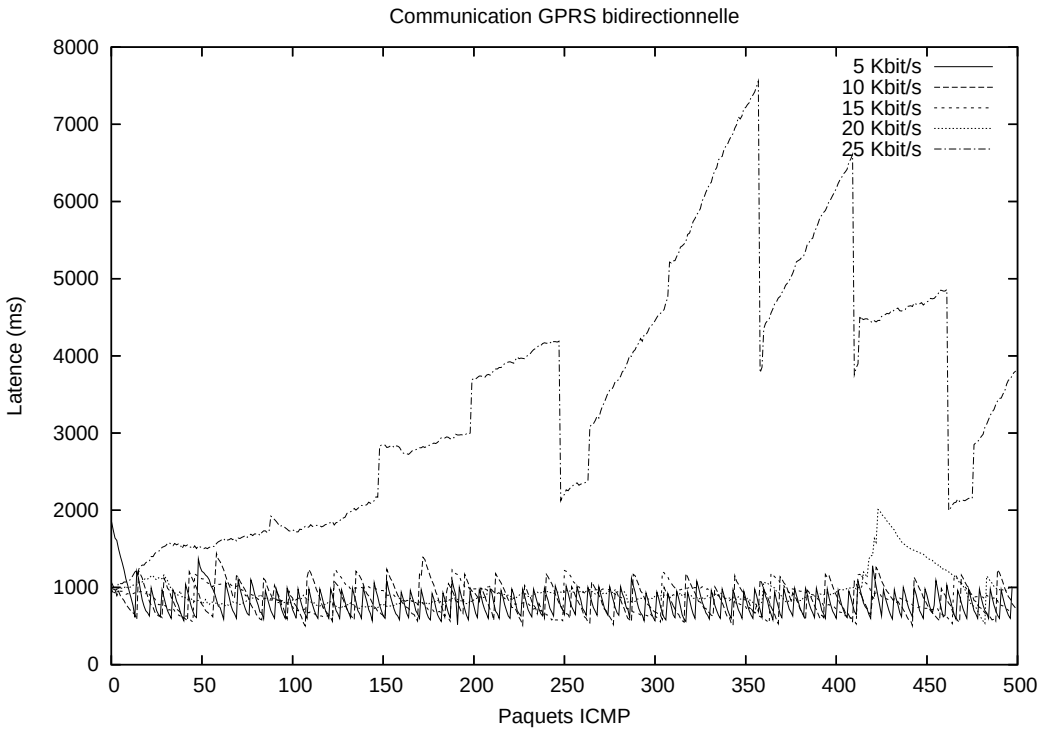

FIG. 2.4 – Latence du réseau GPRS pour une communication bidirectionnelle

pour des téléchargements de données unicast vers des terminaux. L'interconnexion des trois réseaux, Internet, DVB-T et GPRS est effectuée au niveau de la couche IP, par un routeur spécifique appelé HNIS. Cette architecture est un prototype expérimental, développé afin de répondre à une problématique spécifique à l'usage de GPRS en tant que voie de retour. Des solutions plus « courantes », comme UDLR, se sont révélées inadaptées à cette problématique à cause du caractère statique des règles de routage.

La voie de retour GPRS

Nous avons réalisé les expérimentations suivantes sur différents réseaux GPRS de production. Plusieurs abonnements GPRS français (Orange et SFR) et finnois (Sonera et Radiolinja) ont été utilisés. Chaque abonnement proposait une connexion GPRS avec trois timeslots pour la voie descendante, et deux timeslots pour la voie montante. Les résultats des différents abonnements étant très similaires entre eux, seuls ceux de l'opérateur Radiolinja sont présentés ci-dessous. Dans cette étude, ne pouvant agir sur les paramètres de configuration du réseau GPRS, celui-ci a été

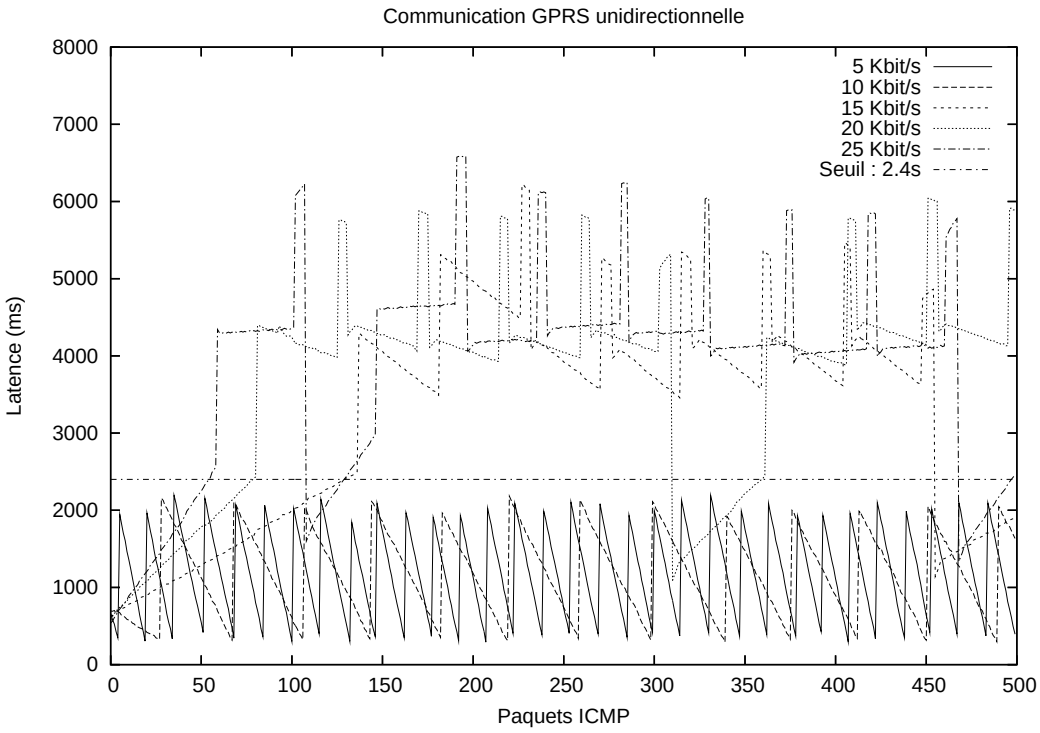

FIG. 2.5 – Latence du réseau GPRS pour une communication unidirectionnelle

considéré comme une boîte noire. Les tests expérimentaux consistaient à envoyer des paquets IP de petite taille (ICMP, 84 octets) du terminal vers un serveur, avec différents débits, et à recevoir les réponses à ces paquets de deux façons différentes. Les paquets sont envoyés de façon à générer des débits constants et on mesure le Round Trip Time (RTT) relatif à chaque paquet.

Dans un cas, les paquets sont envoyés par GPRS et reçus par GPRS. Ce mode de transmission est appelé *bidirectionnel*, puisque la voie ascendante et la voie descendante de GPRS sont utilisées. Dans un autre cas, les paquets sont envoyés par GPRS et reçus par LAN. Ce mode de transmission est appelé *unidirectionnel* puisque nous utilisons exclusivement la voie montante de GPRS pour l'envoi de données. La latence du réseau Local Area Network (LAN) étant négligeable ($\leq$ 1ms) devant celle de GPRS ($\simeq$ 500ms) celle-ci ne sera pas prise en considération. Il est donc possible de comparer le comportement de GPRS dans un usage spécifique, correspondant à une utilisation de GPRS en tant que voie de retour (transmission *unidirectionnelle*) par rapport à un usage classique (transmission *bidirectionnelle*).

Dans un mode *bidirectionnel*, la latence du réseau GPRS reste proche de 800ms

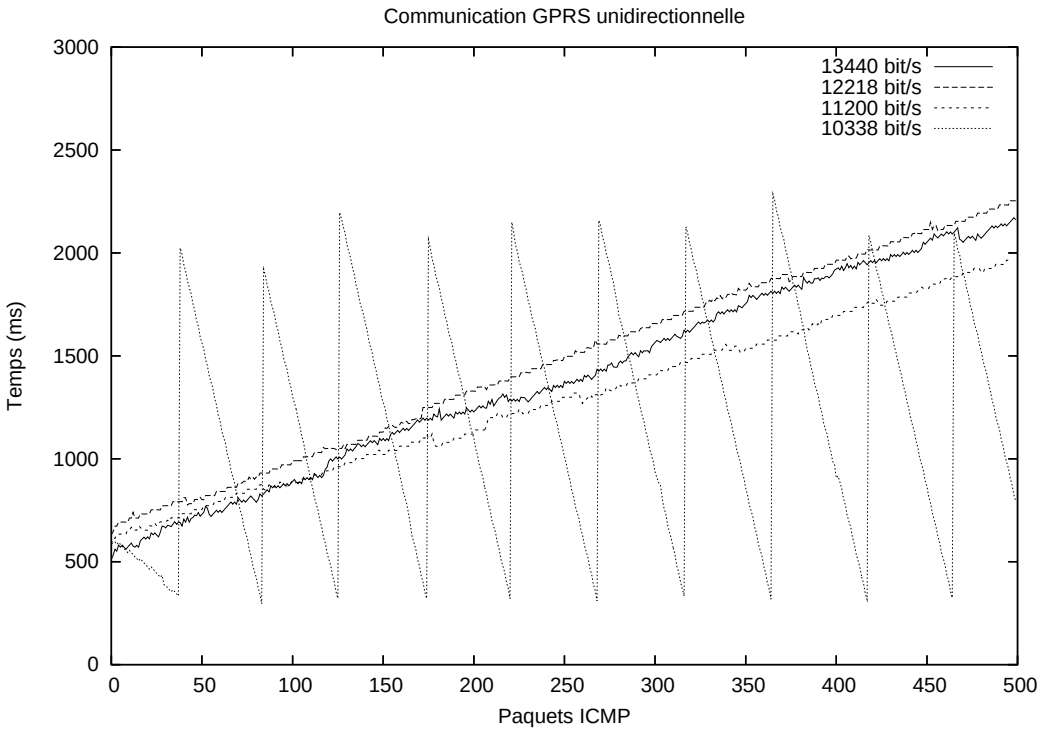

FIG. 2.6 – Valeur du débit critique

malgré de nombreuses oscillations (Fig. 2.4) pour des débits inférieurs ou égaux à 20Kbit/s. A 25 Kbit/s, la limite de bande passante est atteinte, et la latence du réseau augmente fortement dû au remplissage des mémoires tampons du réseau GPRS. Les petites oscillations peuvent être imputées au multiplexage temporel de la transmission sur GPRS. Dans un mode *unidirectionnel*, le comportement du réseau (latence en fonction de la charge) diffère totalement. En effet, la latence augmente brutalement pour des débits beaucoup plus faibles. Une analyse plus détaillée montre un saut de latence de 2,4s à plus de 4s pour des débits de 15Kbit/s, 20Kbit/s, et 25Kbit/s comme le montre la figure 2.5 . La latence augmente de façon quasi-linéaire jusqu'à atteindre un seuil critique de 2,4s. GPRS possède donc deux comportements différents lors d'une communication unidirectionnelle :

- un régime stationnaire oscillant pour des débits inférieurs à 10Kbit/s,
- un régime linéaire croissant, conduisant à une brusque augmentation de latence de 2,4s à 4s, pour des débits supérieurs à 10Kbit/s.

Une étude plus précise a permis de spécifier le débit critique γ au delà duquel apparaît le saut de latence. (11200 Bit/s pour le réseau Radiolinja) (Fig. 2.6). L'hypothèse

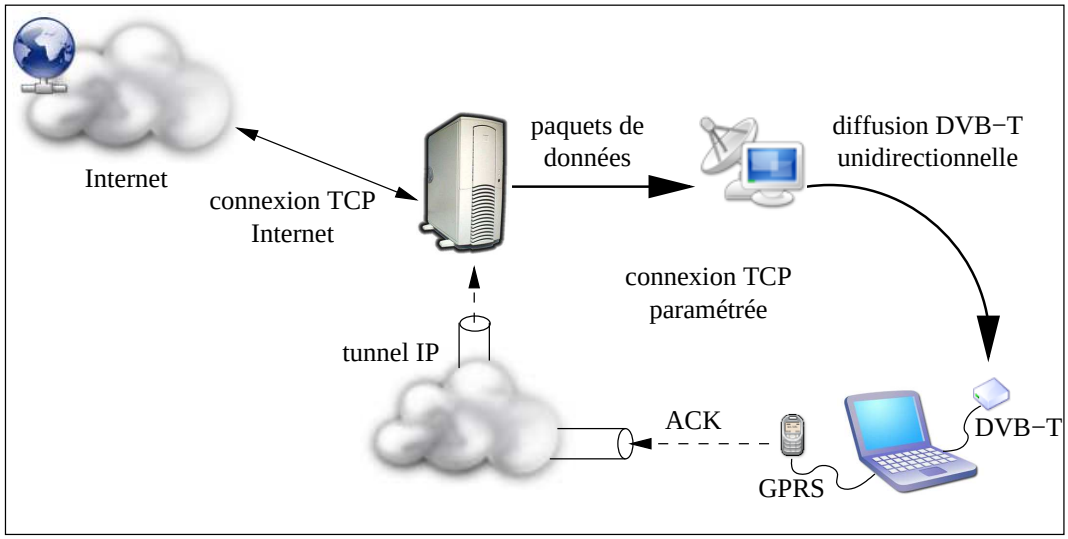

FIG. 2.7 – Connexion TCP séparée

peut donc être faite que le réseau GPRS utilise différentes classes de trafic sur sa voie montante, et que celles-ci sont directement liées à la présence de trafic sur la voie descendante GPRS. Dans cette architecture hybride DVB-T/GPRS, il est important de prévenir ce saut de latence, puisque celui-ci pourrait conduire à des *timeouts* au niveau de la couche transport TCP. Ne pouvant accéder aux paramètres de configuration du réseau GPRS, une solution est proposée en adaptant les couches réseau et transport.

Analyse de l'architecture

Cette architecture de réseaux hybride présente la problématique courante, concernant la différence de bande passante des réseaux asymétriques, à laquelle s'ajoutent d'autres asymétries dues à l'hétérogénéité des réseaux. Le coefficient d'asymétrie de cette architecture, pour le protocole TCP/IP, est particulièrement élevé : 27. Le coefficient d'asymétrie est défini comme étant le rapport entre la bande passante de la voie descendante sur la bande passante de la voie montante, divisé par le rapport de la taille des paquets transmis sur ces voies [23]. Avec des débits de 20Mbit/s pour DVB-T et de 20Kbit/s pour la voie montante GPRS, ainsi que des tailles de paquets de 1500 octets transmis sur DVB-T et de 40 octets sur GPRS, le coefficient

d'asymétrie vaut :

$$K = \frac{\left(\frac{20Mbit/s}{20Kbit/s}\right)}{\left(\frac{1500}{40}\right)} = 26,6 \tag{2.1}$$

Une asymétrie supérieure à 1 entraîne inévitablement une congestion sur la voie de retour. Divers mécanismes existent afin de réduire le phénomène de congestion (Ack filtering, Delayed Ack...) [23]. Seules les solutions ne nécessitant que des modifications coté terminal ont été retenues. La couche transport TCP du terminal a donc été paramétrée de façon à utiliser la fonctionnalité DelayedACK [24], et en fixant la valeur de retard à 500ms. De cette façon le terminal génère un accusé de réception tous les deux paquets ou à l'expiration d'un délai d'attente de 500ms. En divisant ainsi par deux le nombre d'Ack générés, le coefficient d'asymétrie est lui aussi divisé par deux. Le débit maximal atteignable est alors de :

$$Débit_{max_1} = \frac{20Mbit/s}{\left(\frac{27}{2}\right)} \simeq 1,4Mbit/s \tag{2.2}$$

Sur le plan temporel, l'asymétrie de délai entre GPRS (700ms) et DVB-T (50ms) n'entraîne aucune conséquence puisque TCP ne considère que le délai aller-retour. C'est donc bien la valeur importante du RTT ($\simeq$750ms), et non pas la différence entre les délais GPRS et DVB-T qui affecte les performances du protocole TCP. Le débit maximal que peut atteindre le protocole TCP, sans saturation de la bande passante, est limité par la taille de sa fenêtre de congestion et par le RTT.

$$Débit_{max_2} = \frac{TCPWindowSize}{RTT} \tag{2.3}$$

Le routeur HNIS intègre un proxy TCP, permettant de relayer les connexions du terminal vers Internet. Il est ainsi possible de conserver un paramétrage TCP spécifique à cette architecture, quels que soient les paramètres TCP de l'hôte Internet distant. La couche transport TCP du routeur HNIS a donc été paramétrée avec une fenêtre de congestion égale à 128Ko (cette valeur est supportée par toutes les implémentations de TCP, quelque soit le système d'exploitation). Le débit maximal est alors de 1,3Mbit/s, donc de même ordre que $Débit_{max_1}$, donné par l'étude des débits.

La robustesse du réseau DVB-T, par rapport aux erreurs de transmission, fait de lui un candidat idéal pour la transmission des paquets de données à destination du terminal. Les accusés de réception étant cumulatifs, le taux d'erreur important de la voie de retour GPRS n'influe pas sur les performances de TCP. L'utilisation d'un proxy-relai et le paramétrage spécifique de la connexion TCP au niveau du routeur

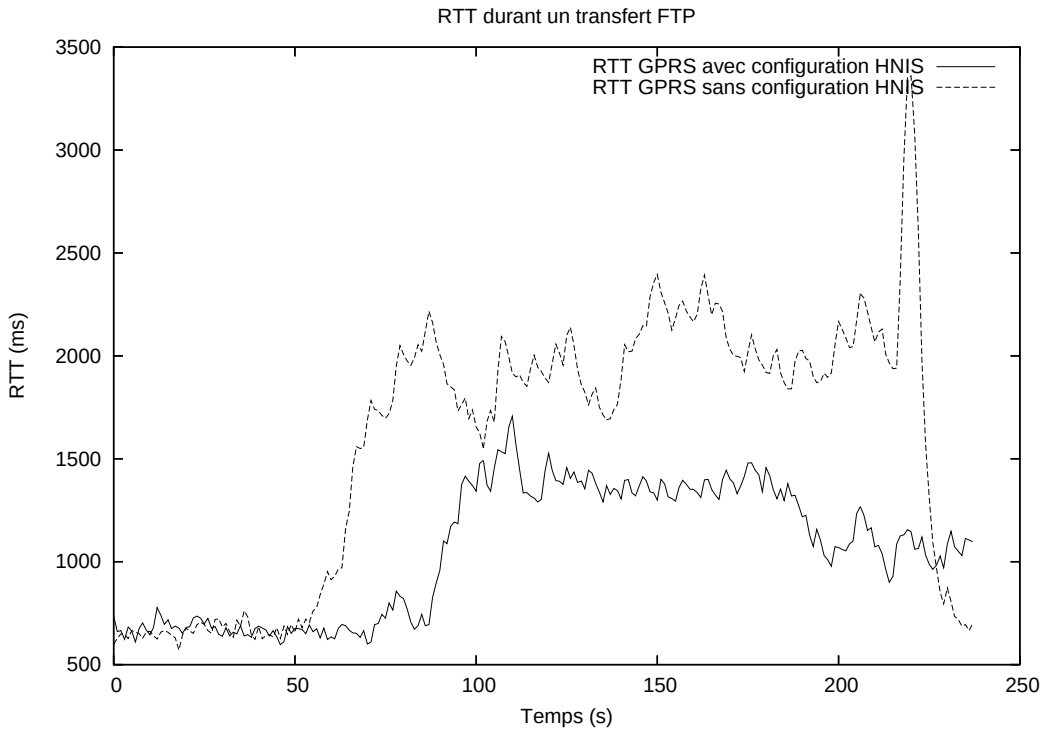

FIG. 2.8 – RTT GPRS durant un transfert FTP

hybride améliorent considérablement les performances de transfert de données vers le terminal dans cette architecture asymétrique et hétérogène. Cependant, une problématique subsiste lorsque le terminal envoie des données sur le lien GPRS pendant que celui-ci véhicule des accusés de réception relatifs à un transfert sur DVB-T. En effet ce trafic additionnel entraîne un dépassement du débit critique, engendrant ainsi un saut de latence qui conduit à un *timeout* pour le transfert de données sur DVB-T. Afin de prévenir ce saut de latence, qui apparaît lors d'un usage exclusif de la voie montante GPRS, la communication sur GPRS bascule en mode *bidirectionnel*.
A cet effet, le routeur HNIS intègre des règles de routage dites « hybrides ». Ces règles de routage spécifient que l'ensemble des accusés de réception relatifs à une communication TCP initiée par le terminal, est routé sur la voie descendante GPRS et non sur DVB-T. Ces acquittements génèrent ainsi un faible trafic sur la voie descendante, qui suffit à faire commuter le comportement de GPRS d'un mode unidirectionnel à un mode bidirectionnel. Avec ces règles de routage, le trafic sur la voie descendante n'est généré que lorsque cela s'avère nécessaire, et non de façon systématique.
Par conséquent, la consommation de bande passante GPRS, souvent onéreuse, est

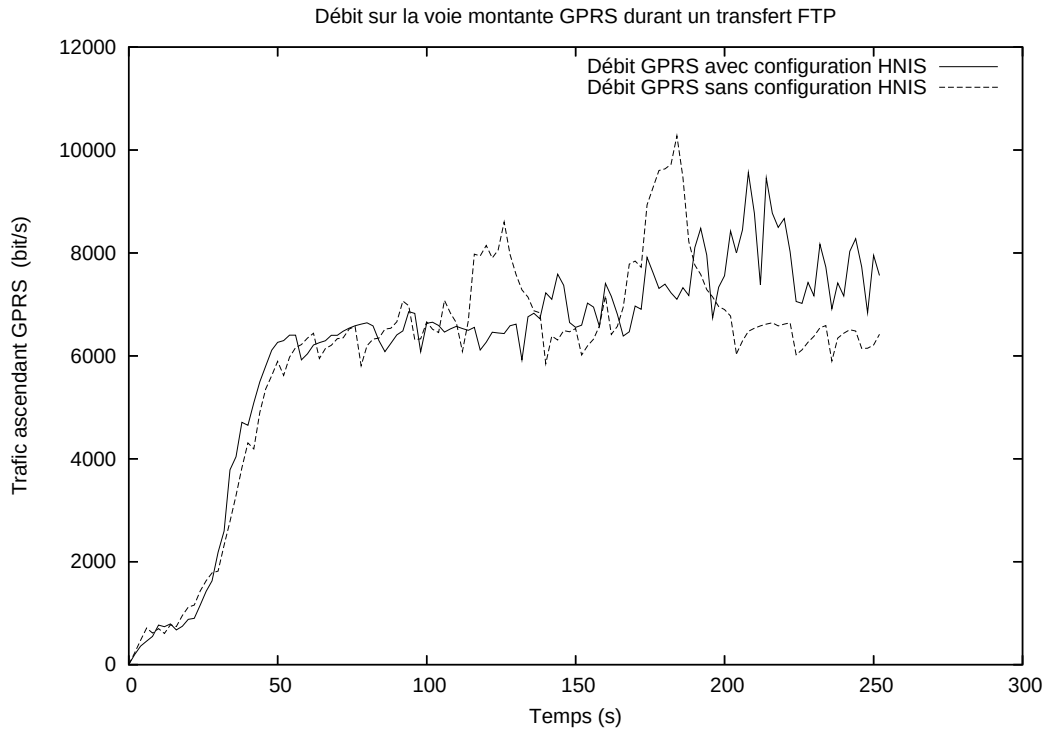

FIG. 2.9 – Débit GPRS durant un transfert FTP

limitée, réduisant ainsi le coût du service proposé.
Le figure 2.10 montre que le gain de performance obtenu avec HNIS en terme de débit sur le lien DVB-T est de l'ordre de 60%. En parallèle le RTT est limité à une valeur moyenne inférieure de 40% à celle obtenue sans configuration HNIS (Fig. 2.8) et ce pour une consommation identique sur le lien GPRS (Fig. 2.9).
Contrairement à la solution UDLR, la solution de coopération de réseaux implémentée avec le système HNIS fonctionne au niveau IP et non au niveau de la couche liaison de données. La solution HNIS offre moins de transparence vis-à-vis des couches supérieures (le routage multicast nécessite une adaptation des règles de routage afin de transférer les flux sur DVB-T) et requiert des adaptations au niveau du terminal (désactivation de la protection contre l'IP spoofing). Cependant cette solution permet de résoudre une problématique spécifique à l'usage de GPRS en tant que voie de retour. Cette perte de transparence vis à vis des couches supérieures est compensée par l'apport de nouvelles fonctionnalités spécifiques, tel que le routage hybride, améliorant les performances globales du système.
Les architectures précédentes HNIS et UDLR, nécessitent l'intégration d'entités ré-

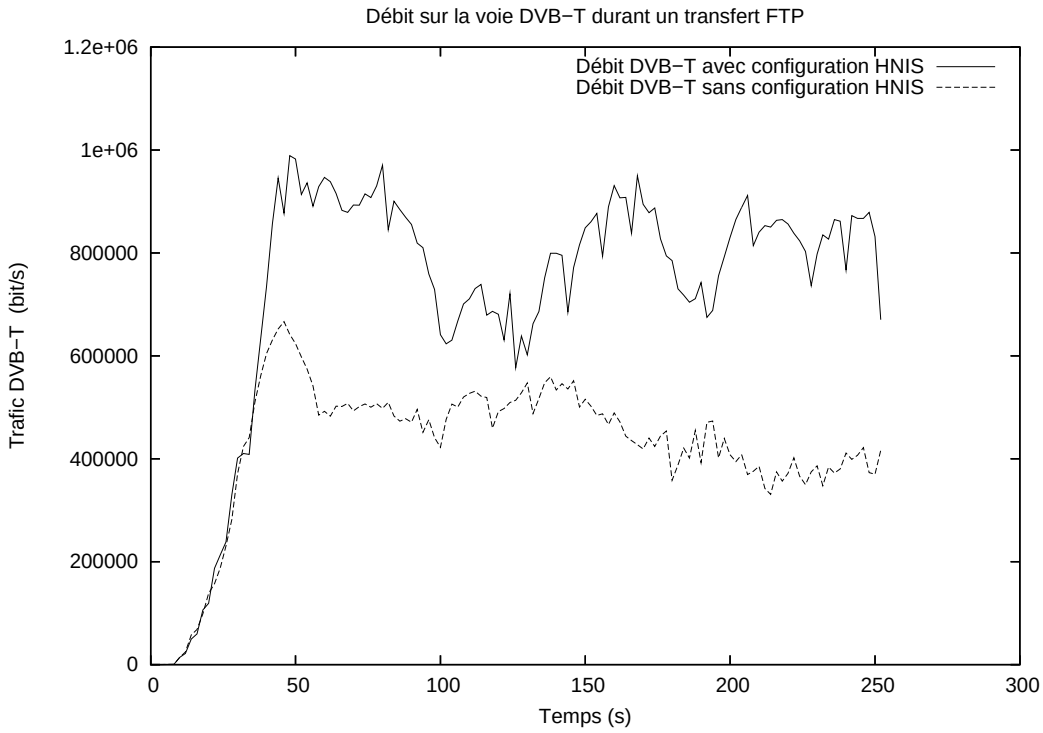

FIG. 2.10 – Débit DVB durant un transfer FTP

seaux spécifiques, les routeurs d'interconnexion, afin d'opérer la coopération de réseaux. L'étude des performances du protocole TCP dans l'architecture HNIS montre la viabilité de ce type d'architecture. Dans la section suivante, nous proposons une solution basée sur le protocole SCTP, n'exigeant pas de routeur d'interconnexion, et conservant des performances similaires.

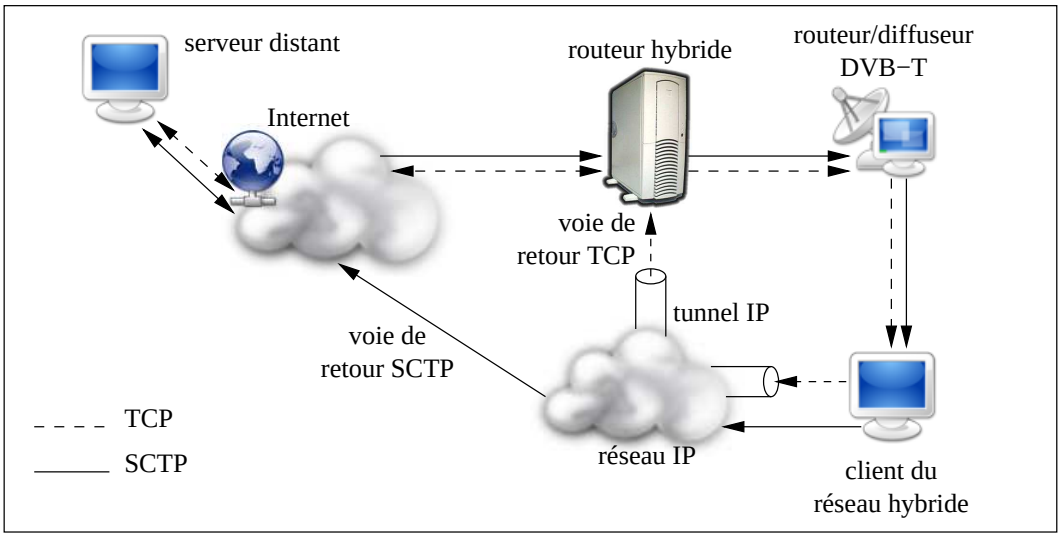

FIG. 2.11 – Comparaison entre TCP et SCTP dans une architecture hybride

2.2.2 Usage du protocole SCTP dans une architecture hybride

Description de l'architecture

Le contexte de coopération de réseaux est identique à celui décrit précédemment. Dans cette approche le protocole SCTP a été substitué au protocole TCP en tant que couche transport. Dans les architectures classiques de réseaux hybrides, les différents réseaux sont interconnectés à l'aide d'un routeur hybride. Le routeur hybride assure la commutation de paquets au niveau de la couche IP ou liaison de données afin de rendre transparent l'usage de différents réseaux pour l'envoi et la réception de données, pour les couches supérieures. Généralement un terminal utilise un tunnel pour envoyer un flux de données ascendant au routeur hybride, qui le relaie sur Internet. Tous les paquets IP relatifs à une communication passent ainsi par le routeur hybride, qui représente donc un point de défaillance significatif dans l'architecture. Ce modèle d'architecture réseaux est couramment utilisé avec la pile protocolaire TCP/IP. La configuration du routage peut être soit statique, soit gérée par un processus du niveau applicatif.
A l'origine, SCTP fût conçu pour fournir un protocole de transport générique pour les applications orientées messages, comme le transport d'information de signalisation. Sa conception inclut un mécanisme de prévention de congestion approprié, le rendant compatible (TCP Friendly) avec TCP Newreno, la version la plus répandue de TCP. Lors de la mise au point de SCTP, les failles du protocole TCP ont été

corrigées, protégeant ainsi SCTP des attaques classiques utilisées contre TCP (par exemple le SYN flooding). En utilisant le protocole SCTP comme couche transport, le plan de routage peut être optimisé et la robustesse de l'architecture améliorée. En effet, une propriété essentielle de SCTP est sa capacité à gérer des nœuds « multi-accès »(multi-homing), nœuds qui peuvent être atteints via différentes adresses IP. Un client multi-accès informe le serveur de ses différentes adresses IP avec le paramètre « ADDRESS »du message d'initialisation INIT. En contrepartie, le client n'a besoin de connaître qu'une seule adresse IP du serveur pour initialiser la connexion, puisque celui-ci informe le client de ses différentes adresses avec le message INIT-ACK. L'ensemble des connexions potentielles entre les adresses IP du serveur et du client est appelé : *association*. A l'initialisation de l'association, une adresse IP de la liste est sélectionnée comme adresse primaire et les messages de données sont transmis par défaut sur le chemin de cette adresse primaire. Dans cette architecture de coopération de réseaux, une association SCTP comprend donc les adresses suivantes $\{@S; @T_g, @T_d\}$ où l'adresse @S est l'adresse d'un hôte distant sur Internet, et @T$_g$ et @T$_d$ respectivement les adresses IP des interfaces GPRS et DVB-T du terminal. En initiant la communication, du terminal vers le serveur, sur la voie GPRS, puis en spécifiant l'adresse IP DVB-T comme adresse primaire, les messages de données à destination du terminal sont envoyés par DVB-T, et les accusés de réception sont renvoyés par GPRS. La communication ne passe donc plus par un routeur hybride et le routage triangulaire sur la voie montante est supprimé (terminal → routeur → serveur) (Fig. 2.11). Une extension de ce mécanisme pourrait être utilisée pour réaliser un semi-handover (handover uniquement sur la voie descendante et non sur la voie montante) entre différentes cellules DVB-T au niveau de la couche transport [25].

Problématique

La problématique de l'étude que nous avons réalisée est plus générale que celle décrite précédemment, spécifique à un usage unidirectionnel de GPRS, et concerne principalement le phénomène de saturation de la voie de retour, qui est le problème majeur dans les architectures réseaux asymétriques. Le protocole SCTP est utilisé en tant que couche transport afin de définir une nouvelle solution à cette problématique par le biais de ses nouvelles fonctionnalités. Lors d'une saturation de la voie de retour, les paquets sont mis dans une mémoire tampon au niveau des routeurs entraînant ainsi une augmentation de délai. Dans cette architecture, le serveur envoie les paquets de données par DVB-T et reçoit les accusés de réception par GPRS. Le RTT global

du système est donc égal à la somme du délai de la voie ascendante GPRS et du délai de la voie descendante DVB-T : RTT_{global}=DELAY_{gprs}+DELAY_{DVB-T}. Par conséquent, une saturation de la voie ascendante GPRS entraîne une augmentation du RTT_{global}. Cette augmentation du RTT_{global} affecte la boucle de régulation du protocole, dégradant ses performances et pouvant conduire à des timeouts.

$$Débit_{max} = \frac{TCPWindowSize}{RTT} \tag{2.4}$$

Le coefficient d'asymétrie K de bande passante pour cette architecture est différente pour le protocole TCP et SCTP. Dans ces simulations, la taille des paquets de données est de 1500 octets et la taille des accusés de réception est de 40 octets pour le protocole TCP et 48 octets pour le protocole SCTP.

$$K_{TCP} = \frac{\frac{20Mbit/s}{20Kbit/s}}{\frac{1500}{40}} = 26,6 \tag{2.5}$$

$$K_{SCTP} = \frac{\frac{20Mbit/s}{20Kbit/s}}{\frac{1500}{48}} = 32 \tag{2.6}$$

Le protocole SCTP semble donc a priori moins adapté que TCP à des architectures fortement asymétriques. Cependant les nouvelles fonctionnalités de SCTP, et une légère modification du protocole coté terminal peuvent compenser les contre-performances dues à un coefficient d'asymétrie plus important. Le mécanisme de contrôle de congestion de SCTP est dérivé de TCP NewReno, et a été adapté pour le multihoming. SCTP comporte de nouvelles caratéristiques par rapport à TCP comme :

- communication pas messages,
- multihoming,
- multistreaming,
- utilisation de COOKIES pour l'initialisation de la connexion,
- message de contrôle HEARTBEAT

Le protocole SCTP est décrit de façon plus détaillée dans le chapitre 3.4 page 51.

SCTP Variable Ack Rate : SCTPVAR

SCTP peut être utilisé comme protocole de transport pour les applications nécessitant la vérification ou la détection d'erreur de sessions. Un hôte d'extrémité peut envoyer des messages HEARTBEAT (HEARTBEAT CHUNK) pour tester la viabi-

lité d'une adresse particulière présente dans l'association. Un message d'acquittement HEARTBEAT ACK est renvoyé à l'hôte en réponse au HEARTBEAT CHUNK. Une particularité des messages HEARTBEAT est que les accusés de réception HEARTBEAT ACK sont toujours renvoyés à l'adresse source des HEARTBEAT CHUNK, qui correspond à l'adresse IP GPRS du terminal mobile dans cette architecture. Dans ce cas, les HEARTBEAT CHUNK sont donc envoyés par la voie ascendante GPRS et les HEARTBEAT ACK par la voie descendante GPRS, ce qui permet de mesurer le RTT_{GPRS}. Comme TCP, SCTP utilise un mécanisme d'acquittement cumulatif, ou un accusé, appellé STRECTH ACK, peut acquitter plus d'un paquet. Ainsi le nombre d'ACK pour un nombre de segments donné peut être réduit. Nous proposons d'utiliser le même mécanisme dans une variante de SCTP que nous nommons SCTPVAR. Avec le protocole SCTPVAR nous proposons de contrôler le nombre d'ACKs générés sur la voie ascendante GPRS en fonction du RTT_{GPRS}. Le lien DVB-T étant exclusivement utilisé pour la transmission de paquets de données à destination du terminal, l'hypothèse est faite que la voie descendante GPRS n'est jamais saturée. Ainsi une augmentation du RTT_{GPRS} traduit une saturation de la voie ascendante. Dans la version standard de SCTP, les messages HEARTBEAT CHUNK sont envoyés uniquement par le serveur toutes les trente secondes. Dans la version SCTP modifiée, le client envoie également des messages HEARTBEAT, mais à une fréquence supérieure, toutes les trois secondes, afin de détecter plus rapidement les variations de RTT sur GPRS. Un message HEARTBEAT ayant une taille de 56 octets, la consommation de bande passante moyenne induite par la mesure du RTT_{GPRS} est donc de 150bit/s, ce qui reste acceptable par rapport à la bande passante disponible sur le lien montant GPRS (20Kbit/s). A l'instar de TCP Vegas, où le mécanisme de contrôle de congestion est basé sur la mesure du RTT, l'algorithme de gestion des acquittements mémorise la valeur minimale RTT_{min} du RTT_{GPRS}. Deux paramètres α et β sont définis, afin de représenter le remplissage des mémoires tampon sur le lien GPRS. α et β sont modifiables par l'utilisateur, et sont fixés de manière empirique, pour cette architecture, à α=1.01 (1%) et β=1.02 (2%). SCTPVAR utilise l'algorithme suivant pour adapter le débit des accusés de réception :

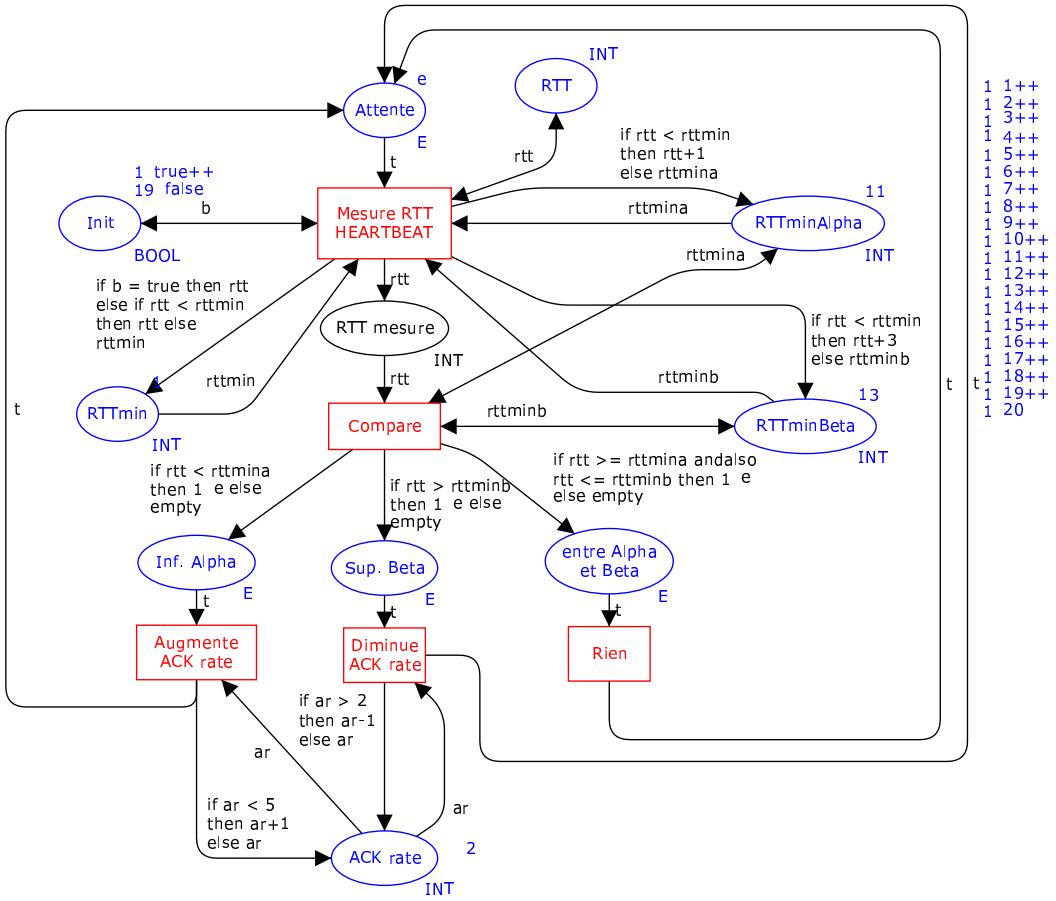

FIG. 2.12 – Déscription en réseau de Petri de l'algorithme gestion des ACKs

loop
  mesure du RTT
  if RTT $<$ α * RTT$_{min}$ **then**
    augmenter le débit des ACKs
  end if
  if RTT $>$ β * RTT$_{min}$ **then**
    diminuer le débit des ACKs
  end if
end loop

La figure 2.12 montre une repésentation en réseaux de Petri de cet algorithme. Des simulations à l'aide de l'outils CPNTOOLS ont permis d'assurer l'absence d'états bloquants dans cet algorithme. Rapport de simulations CPNTOOLS :

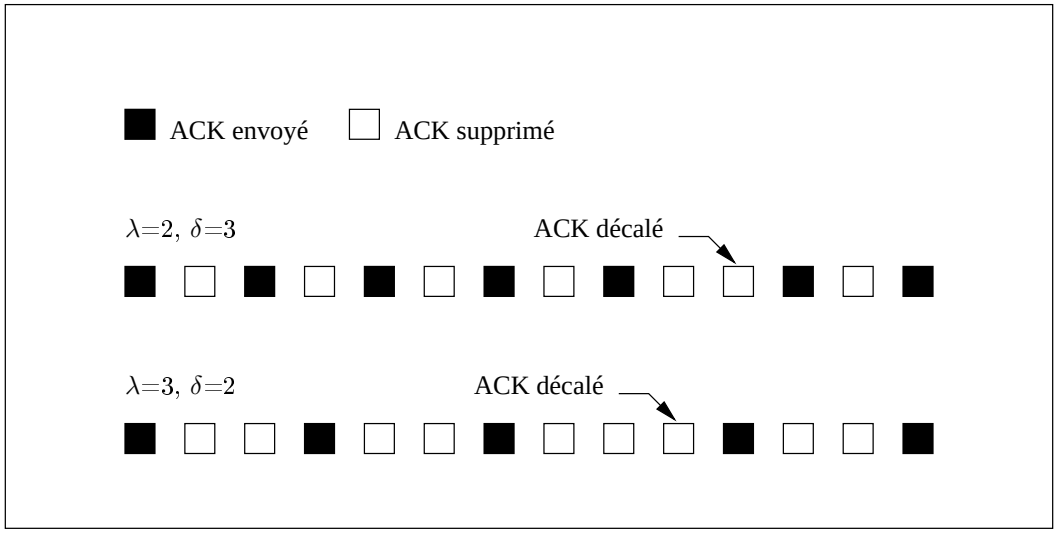

FIG. 2.13 – Régulation des accusés de réception

```
Liveness Properties
-------------------------------------------------------------------------
  Dead Markings:  None
  Dead Transitions Instances: None

  Live Transitions Instances: All
```

Cet algorithme permet d'utiliser l'ensemble de la bande passante disponible sur le lien GPRS montant tout en prévenant la mise en mémoire tampon des paquets. Afin de réduire le trafic des accusés de réception sur la voie de retour, il est possible soit d'ajouter un retard variable avant l'envoi des accusés de réception, soit d'augmenter le ratio des acquittements (nombre de segments SCTP validés par un ACK). L'utilisation d'un retard nécessite une modification du ratio des accusés de réception (augmenter la valeur du retard serait inutile si le ratio était toujours égal à sa valeur par défaut : 2) et dépend également de la précision temporelle du système d'exploitation pour une implémentation réelle. Par conséquent, il a été choisi d'ajuster la valeur du ratio des acquittements plutôt que la valeur du retard. Soit λ la valeur du ratio des ACKs, qui est égal à 2 dans la version standard de SCTP, une augmentation de 1 correspond à une réduction de deux tiers du trafic des accusés de réception sur la voie de retour.
Cette variation brutale du trafic sur la voie de retour, et par conséquent du RTT_{global}

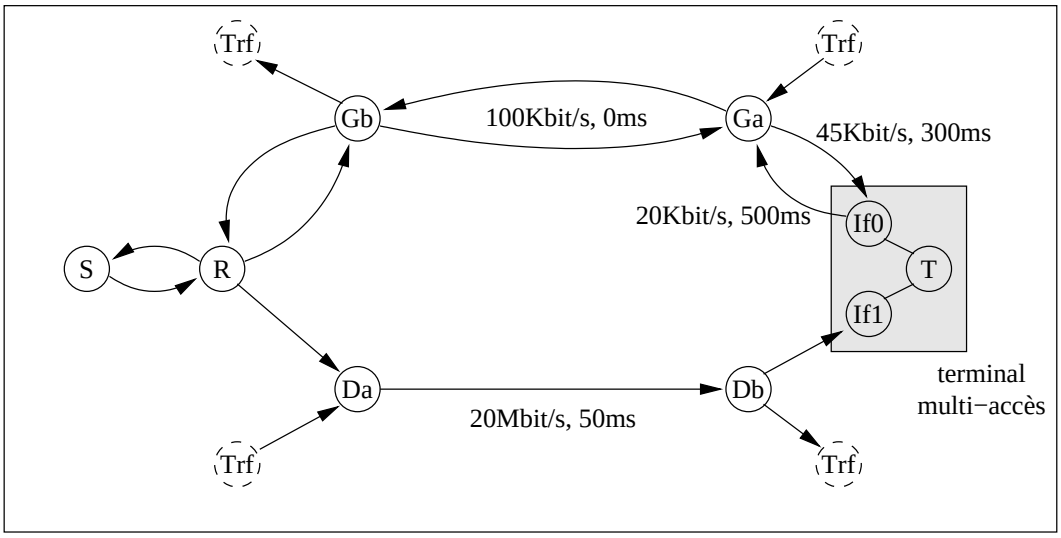

FIG. 2.14 – Modèle de simulation NS2

peut engendrer l'apparition d'un trafic bursté sur le lien DVB-T, qui dégrade les performances du système. Pour obtenir une meilleure granularité dans la variation du ratio, chaque unité est divisée en six sixièmes. Soit δ le nombre de sixièmes, celui-ci spécifie le nombre d'ACKs qui sont envoyés à un ratio λ+1. La valeur du Variable ACK Rate est donc représentée par la notation suivante VAR=$\{\lambda,\delta\}$. Par exemple, si VAR=$\{2,1\}$, SCTPVAR enverra par défaut 1 ACK pour 2 segments reçus, et 1 ACK sur 6 sera décalé pour obtenir un ratio de 3. Un autre exemple, si VAR=$\{3,2\}$, SCTPVAR enverra 1 ACK sur 3, et 2 ACKs sur 6, ou 1 ACK sur 3 sera décalé pour obtenir un ratio de 4 (Fig. 2.13).

Ce mode de régulation offre une meilleure granularité et permet des variations plus souples du trafic des acquittements sur la voie de retour. Contrairement, à la solution de « ACK filtering », généralement implémentée au niveau du routeur d'accès, le processus SCTPVAR estime directement si un accusé de réception peut ou non être transmis. La suppression des acquittements ne s'opère que sur les ACKs « standards », les accusés comportant des champs spéciaux, comme le *flag* « DupACK », sont transmis immédiatement. Si le récepteur détecte la perte d'un segment SCTP, l'algorithme est initialisé, et le ratio d'accusé de réception est rétabli à sa valeur par défaut : 2 [24]. De cette façon, SCTPVAR peut opérer de façon efficace les phases de Fast Recovery et Fast Retransmit. A l'aide des messages HEARBEAT, SCTPVAR peut surveiller de façon active l'état de congestion de la voie de retour, et adapter dynamiquement le flux d'accusés de réception. C'est sa supériorité sur le protocole

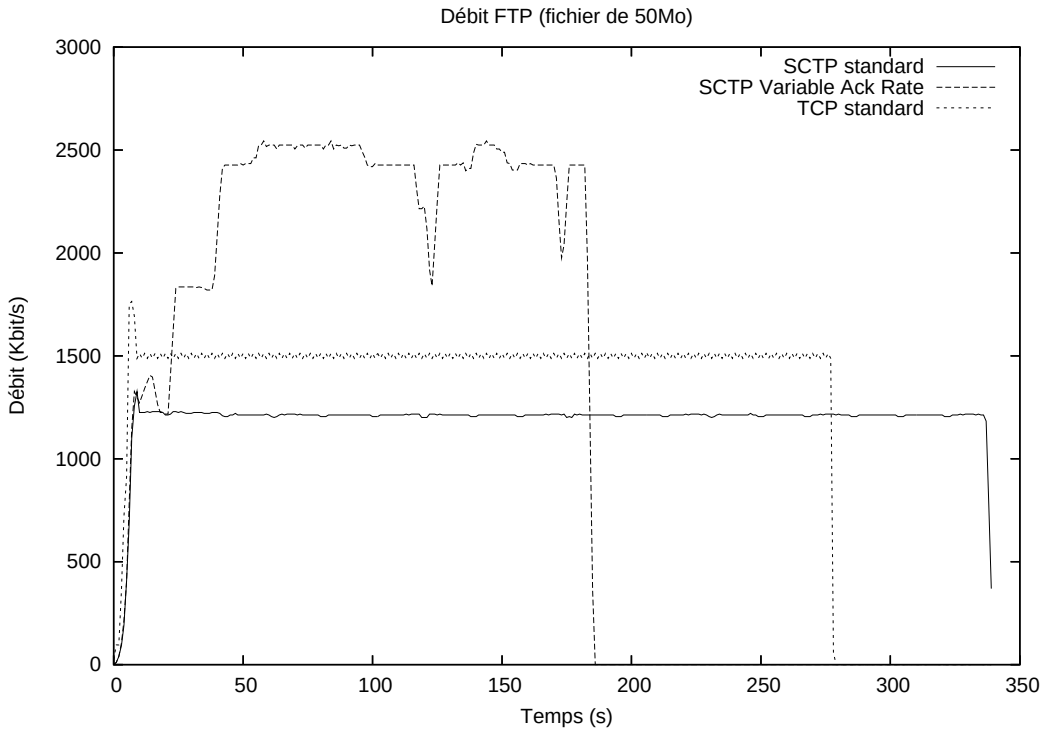

FIG. 2.15 – Débit FTP (fichier de 50Mo)

TCP qui ne peut que mesurer le RTT_{global} : il lui est donc impossible de différencier une congestion sur la voie descendante, d'une congestion sur la voie de retour, et par conséquent, le flux des acquittements ne peut pas être géré dynamiquement.

Evaluation de SCTPVAR

Une version de SCTPVAR a été développée avec Network Simulator v2.28, qui inclut une implémentation de SCTP programmée par le Protocol Engineering Laboratory. Les simulations consistent en des transferts de fichiers de différentes tailles entre un serveur et un client, interconnectés avec divers liens représentant une architecture hybride (Fig. 2.14). Les paquets ont une taille de 1500 octets, incluant les en-têtes TCP/IP et SCTP/IP. Le terminal `T` possède une interface bidirectionnelle `If0` correspondant à un accès GPRS ayant respectivement 20Kbit/s et 45Kbit/s de bande passante pour les voies ascendante et descendante. La seconde interface `If1`, est unidirectionnelle et est équivalente à un récepteur DVB-T, connecté à un lien de diffusion avec 20Mbit/s de bande passante. Les nœuds `Ga` et `Gb` constituent les routeurs d'entrée et de sortie du réseau GPRS et le lien entre les nœuds `Da` et `Db` simule

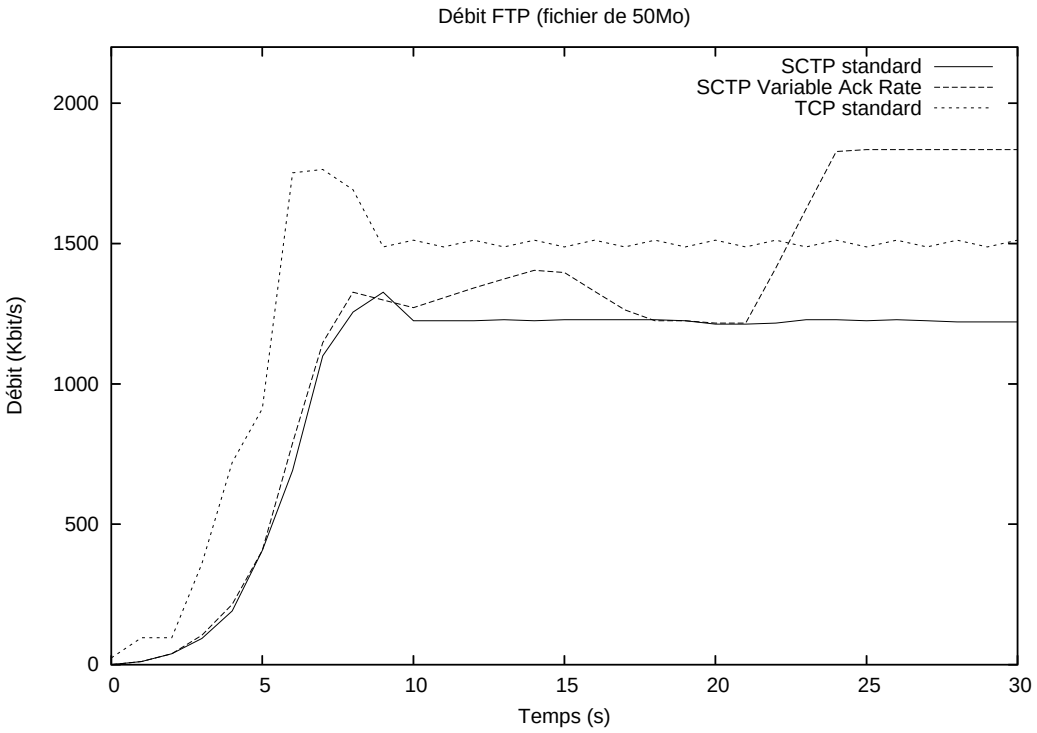

FIG. 2.16 – Débit FTP (fichier de 50Mo)

un réseau DVB-T. Tous les autres liens possèdent une bande passante de 1Gbit/s et un délai de 0ms, et permettent de connecter les nœuds `Trf` qui servent à l'injection et à la réception de trafics concurrents (phénomène de congestion).
La figure 2.15 montre le débit atteint lors d'un téléchargement FTP d'un fichier de 50Mo avec les protocoles TCP, SCTP et SCTPVAR selon le modèle décrit précédemment. Le protocole TCP atteint un débit supérieur de 20% au protocole SCTP standard, cette différence est directement liée au rapport d'asymètrie des deux protocoles. Les accusés de réception du protocole SCTP ont une taille de 48 octets, soit 20% de plus que ceux de TCP (40 octets). Cette différence se retrouve dans le coefficient d'asymétrie et directement sur les performances des protocoles.

En régulant le trafic des acquittements sur la voie de retour, le protocole SCTP-VAR limite la congestion sur GPRS et maintient le RTT_{global} à une valeur proche de 600ms, alors que la latence est supérieure à 1s avec le protocole TCP (Fig. 2.18). La latence relativement faible du protocole SCTPVAR, comparée à celle de TCP, accroît considérablement les performances du protocole lors d'un téléchargement FTP, le débit pouvant atteindre 2,5Mbit/s (Fig. 2.15).

Taille des des fichiers (Ko)	*TCP DelAck (100ms)*	*TCP_5 5 paquets/ACK*	*SCTP*	*SCTPVAR*
10	1.88s	3.43s	2.63s	2.63s
100	4.37s	8.16s	5.35s	5.21s
1000	9.54s	15.15s	11.4s	11.13s
10000	58.85s	42.87s	71.15s	50.56s
100000	552s	317s	671s	351s

TAB. 2.1 – Durée de transfert pour différentes tailles de fichiers

Le protocole SCTP nécessite quatre échanges pendant la phase d'initialisation, contre seulement trois pour le protocole TCP, ce qui ajoute une durée supplémentaire à chaque initialisation d'une association. La figure 2.16 montre le retard, par rapport à TCP, des protocoles SCTP et SCTPVAR dû à l'initialisation au début de la connexion. Les protocoles SCTP et SCTPVAR ont tous deux des performances similaires. Ce temps additionnel peut s'avérer pénalisant lors de transfert de nombreux petits fichiers, correspondant généralement à un trafic de pages HTML. Des études ont comparés les protocoles SCTP et TCP pour le transfert de pages web [26].
Le support multi-streaming du protocole SCTP permet en quelque sorte d'agréger des transferts de plusieurs fichiers à travers une seule association. Un principe de fonctionnement similaire est utilisé par le protocole HTTP1.1 pour transférer l'ensemble des fichiers constituant une page HTML à travers une seule connexion TCP [27]. Cependant une différence importante réside dans le fait que le protocole TCP considère les données qu'il transmet comme un flux d'octets alors que le protocole SCTP utilise la notion de messages séparés sur différents *streams*.
Les streams SCTP ayant des contrôles de congestion indépendants les uns des autres, la perte d'un paquet ne perturbe que le stream relatif au paquet et non l'ensemble de l'association. La perte d'un paquet TCP entraîne par contre une perturbation dans le transfert du flux d'octets, et par conséquent de l'ensemble des fichiers transférés. SCTP apporte donc une solution au problème du blocage de ligne (head of line blocking) et peut améliorer les performances des transferts de pages HTML par rapport à TCP [28] . Le comportement du protocole SCTPVAR étant identique à celui du protocole SCTP pour de petits fichiers (Tab. 2.2.2), ses performances lors de transferts HTTP seront également identiques. Dans une autre simulation, les durées de téléchargement pour les protocoles SCTP, SCTPVAR et TCP_5 ont été comparées à celles d'une version courante de TCP : TCP DelAck (100ms)(Fig. 2.19). Le protocole TCP_5 est une version modifiée de TCP dans laquelle le récepteur ne renvoie

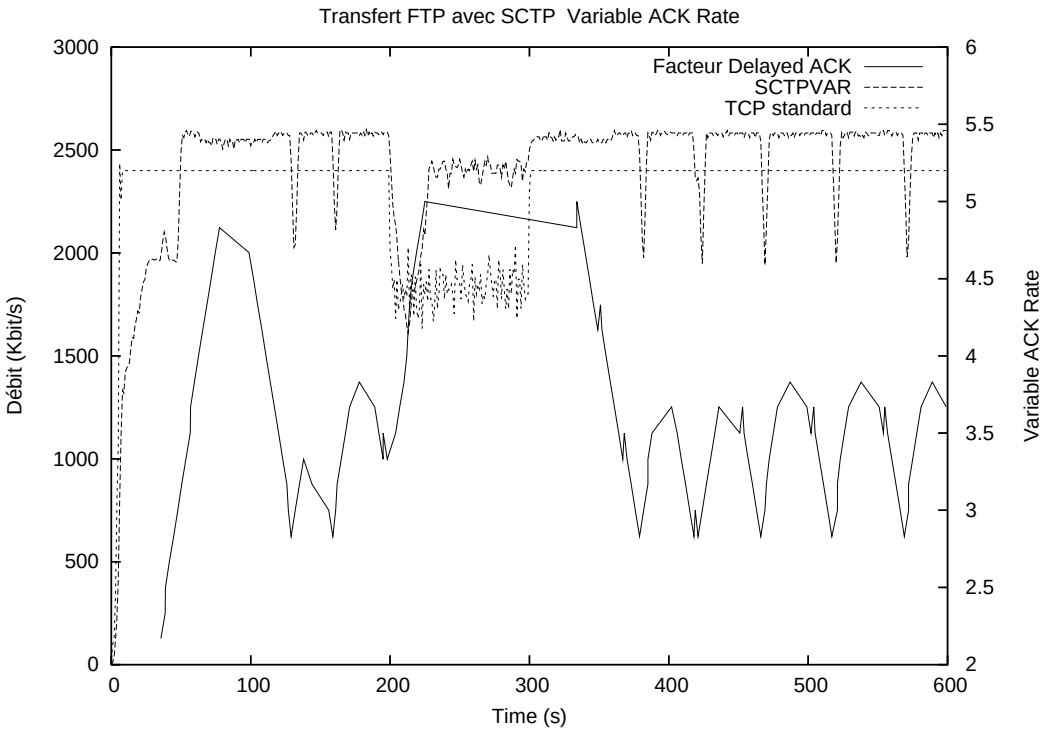

FIG. 2.17 – Transfert FTP avec SCTP Variable ACK Rate

qu'un accusé de réception pour cinq paquets reçus. La comparaison est exprimée en pourcentage, et l'axe des abscisses représente le protocole TCP DelAck (100ms).
Les temps de transfert FTP pour de petits fichiers, inférieurs à 3Mo, avec les protocoles SCTP et SCTPVAR sont très similaires et sont environ 20% supérieurs au protocole TCP de référence. Ces différences entre le protocole TCP et les protocoles SCTP et SCTPVAR ont été expliquées précédement. Les performances des protocoles basés sur SCTP peuvent être fortement améliorées dans le cas de transferts de nombreux petits fichiers. Les durées de téléchargement FTP avec le protocole TCP_5 sont extrêmement longues, 80% supérieures au protocole TCP DelAck, pour de petits fichiers. L'introduction de ce protocole dans la simulation montre bien, que si un ratio d'Ack élevé (valeur égale à 5), configuré de manière statique, améliore fortement les performances de TCP pour de grands fichiers, il entraîne des contre-performances importantes pour de petits fichiers, qui ne sont pas acceptables.
Les mécanismes de contrôle de congestion des protocoles SCTP et TCP NewReno (version la plus répandue de TCP) sont identiques. L'augmentation de la fenêtre de congestion est séparée en deux phases distinctes, la phase slow start et la phase

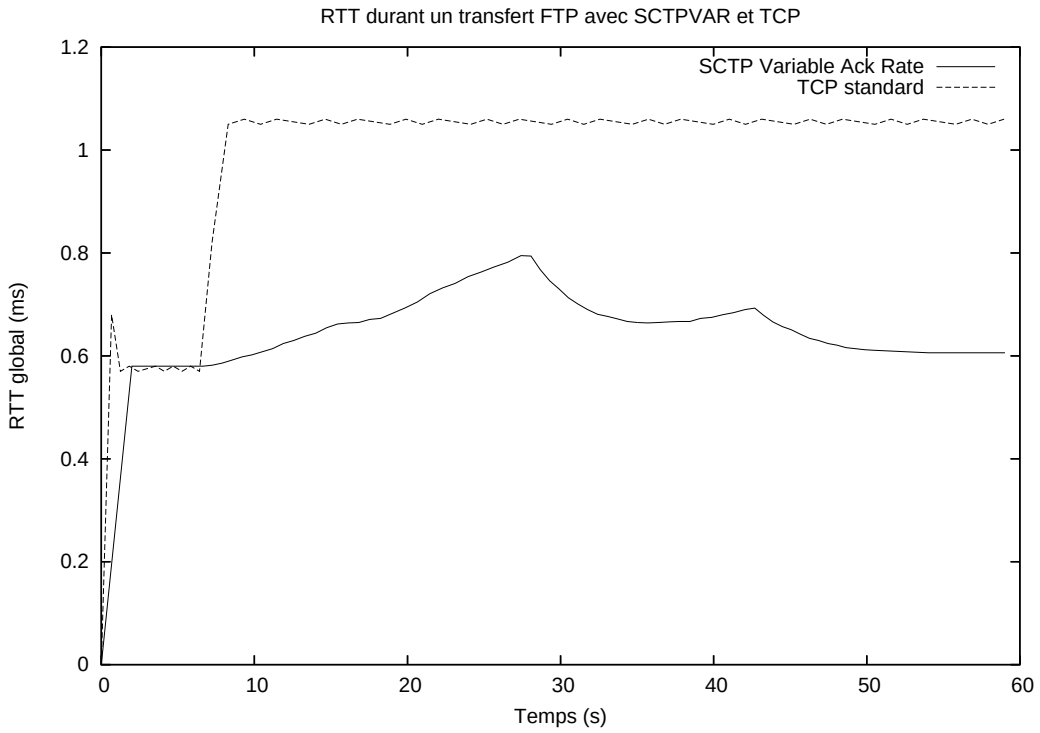

FIG. 2.18 – RTT durant un transfert FTP avec SCTPVAR et TCP

de congestion avoidance. Durant la phase de slow start, la fenêtre de congestion augmente exponentiellement en fonction du nombre d'acquittements (la fenêtre de congestion augmente de deux segments pour chaque Ack reçu), jusqu'à atteindre un seuil : le SSThreshold (Slow Start Treshold). Ensuite, celle-ci augmente linéairement en fonction du RTT (la fenêtre de congestion est augmentée d'un segment à chaque RTT). Le protocole TCP_5 ne renvoie qu'un accusé de réception pour cinq paquets reçus, et ce quelles que soient les conditions de la connexion. Ceci améliore la phase de congestion avoidance (pas de saturation de la voie de retour, et donc pas d'augmentation du RTT), mais dégrade fortement les performances durant la phase de slow start (dépend du nombre d'Ack renvoyés par le récepteur).

Les protocoles SCTP et TCP standard saturent la voie de retour durant la phase de congestion avoidance, ce qui limite leurs performances. La gestion dynamique des accusés de réception utilisée dans le protocole SCTPVAR permet de combiner une phase de slow start efficace, identique à celle de SCTP, et une phase de congestion avoidance offrant des performances proches de celle de TCP_5. Les performances du protocole SCTPVAR restent, dans certaines conditions, inférieures à celles du pro-

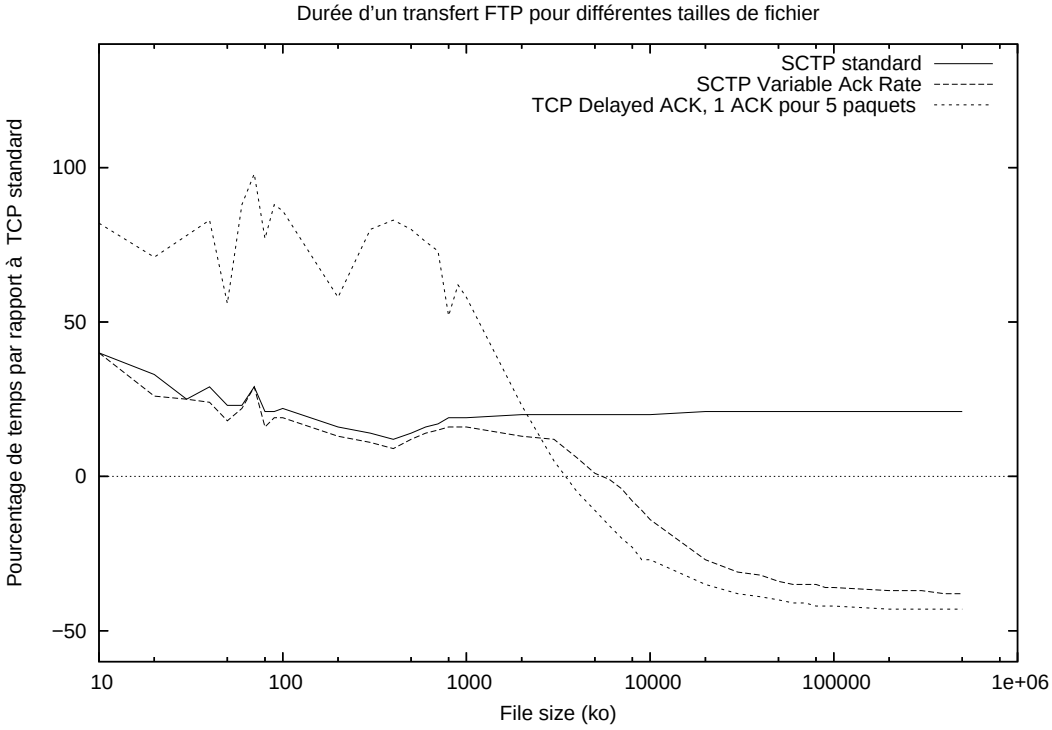

FIG. 2.19 – Durée d'un transfert FTP pour différentes tailles de fichier

tocole TCP (transfert d'un seul petit fichier). Cependant celles-ci sont toujours au moins supérieures au protocole SCTP, et parfois nettement supérieures au protocole TCP (transfert FTP 40% plus rapide pour des fichiers $>$ 10Mo).

La figure 2.19 montre les résultats d'une simulation avec une voie GPRS ascendante de 30Kbit/s afin d'introduire un trafic de congestion et de mettre en évidence l'aspect dynamique du protocole SCTPVAR pour le contrôle du trafic sur la voie de retour. Les débits offerts par les protocoles TCP et SCTPVAR sont du même ordre, et SCTPVAR a augmenté son ratio d'Ack (environ 3.5) afin de prévenir une congestion sur la voie de retour. Entre 100s et 200s, un trafic additionnel est généré sur la voie de retour, réduisant la bande passante disponible à 15Kbit/s. Par conséquent le débit du protocole TCP est réduit de 25% contre seulement 5% pour le protocole SCTPVAR, qui a augmenté son ratio d'accusé de réception à 5 afin de limiter la congestion sur la voie de retour et de limiter ainsi l'augmentation du RTT. Après l'arrêt du trafic concurent (200s), le protocole SCTPVAR rétablit son ratio d'ACK à 3.5.

En conclusion, SCTP peut représenter une alternative à TCP en tant que couche

transport pour de nouvelles architectures réseaux. Dans ce contexte d'architecture hybride, entre un réseau de télécommunication et un réseau de diffusion, SCTP offre de nouvelles opportunités pour améliorer les performances et optimiser le processus :

- optimisation du processus de routage,
- absence de routeur hybride d'interconnexion,
- introduction d'une nouvelle gestion des accusés de réception,
- adaptation uniquement du coté client.

La congestion de la voie de retour, qui est une des principales problématiques des réseaux hybrides asymétriques, peut être mesurée de façon active par le client avec les messages HEARTBEAT. En utilisant une version modifiée de SCTP, appelée SCTPVAR, et en utilisant une boucle de régulation basée sur le délai de la voie de retour, les augmentations importantes du RTT, dues aux phénomènes de congestion, peuvent être évitées. De cette façon, le débit FTP pour des fichiers importants peut être fortement amélioré (40%) tout en gardant des performances similaires à SCTP pour de petits fichiers.

2.3 Synthèse des solutions actuelles

Les trois exemples précédents (UDLR, HNIS, SCTPVAR) illustrent différentes solutions pour opérer une coopération entre différents réseaux de communication. Le protocole UDLR établit la coopération de réseaux en modifiant les couches liaison de données et réseau du routeur client UDLR et du routeur d'interconnexion. Cette approche permet de rendre totalement transparente la caractéristique hybride de l'architecture du point de vue des couches supérieures (réseaux, transport et applicative).

La solution HNIS se place au niveau de la couche réseaux IP, ce qui nécessite, par rapport à la solution UDLR, l'instauration de règles de routage particulières (par exemple le routage du trafic multicast est spécifié de façon statique). Cependant, ce procédé procure une certaine souplesse dans la gestion des communications permettant ainsi d'améliorer l'efficacité de l'architecture dans certaines configurations (instauration de règle de routage *hybride* pour l'usage de GPRS en tant que voie de retour).

L'approche SCTPVAR réduit fortement le domaine d'application aux communications unicasts exploitant le protocole SCTP. En revanche, les optimisations potentielles sont, elles, beaucoup plus nombreuses : optimisation du plan de routage, gestion dynamique de la congestion sur la voie de retour, disparition du routeur

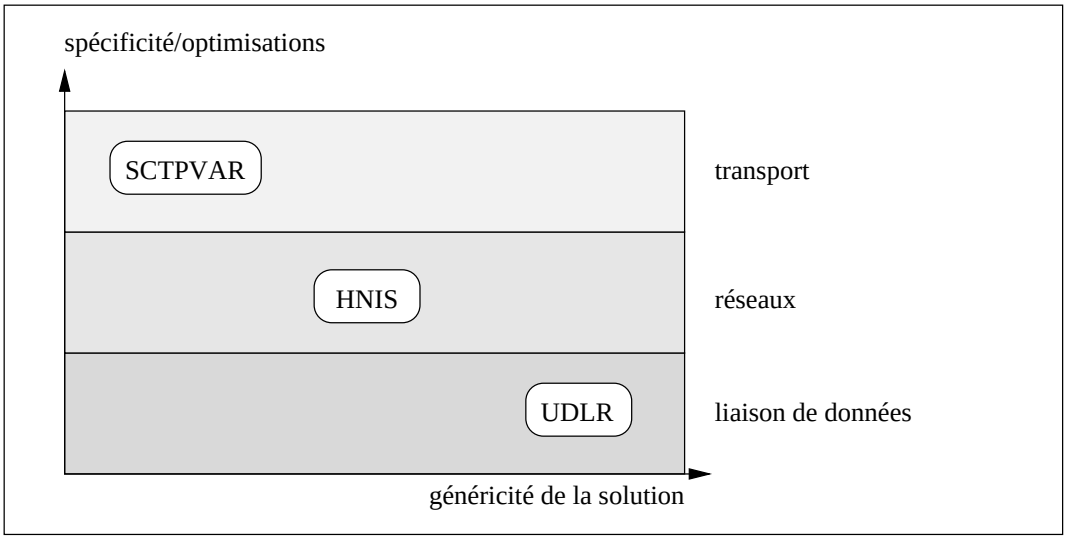

FIG. 2.20 – Comparaison des solutions de coopération de réseaux

d'interconnexion. A travers ces trois démarches on observe que les infrastructures réseaux nécessaires à la coopération (pilote UDLR, routeur d'interconnexion) peuvent être compensées par des mécanismes protocolaires (routage hybride, multi-homing SCTP) au fur et à mesure que la coopération s'effectue à un niveau plus élevé de la pile protocolaire. En contrepartie, ce phénomène a pour effet de restreindre l'usage de l'architecture coopérante à certaines formes de communication (principalement des communications unicast).

L'usage des applications multicast à destination de client mobile est encore peu répandu, ou réservé à des cadres d'utilisations spécifiques. De plus on se place dans un contexte de coopération de réseaux prévue pour fournir une connectivité réseaux, au sens large, à un terminal en situation de mobilité. Ainsi seules les problématiques des communications unicast, correspondant à la quasi-totalité du trafic Internet aujourd'hui, seront abordées par la suite.

Chapitre 3

Spécification d'une pile protocolaire

3.1 Description de l'architecture

Les terminaux utilisateurs mobiles possèdent déjà pour la plupart plusieurs interfaces afin d'accéder à diverses sources d'information, à travers des réseaux radio, multiples et hétérogènes. L'utilisation de chaque catégorie de réseaux reste encore relativement spécifique aux types d'informations véhiculées, par exemple l'UMTS pour la voie ou les données à moyen débit, le WiFi pour les données informatiques, et le DVB pour les contenus audiovisuels. Les fonctionnalités des terminaux se diversifient, et ceux-ci convergent vers un terminal unique pouvant faire office de téléphone, de récepteur télévisuel, et de terminal informatique.
Cependant, si le nombre de fonctionnalités augmente, les usages des ressources réseaux sous-jacentes sont toujours fortement cloisonnés et orientés pour une application particulière. Si l'intégration de plusieurs interfaces dans des terminaux mobiles semble réussie (autonomie, compatibilité électromagnétique...) l'exploitation des ressources réseaux est loin d'être optimale. Si le modèle TCP/IP actuel peut offrir un service sur une interface donnée, de nombreux problèmes apparaissent en situation de mobilité où les conditions de connectivité peuvent évoluer, puisque celui-ci est incapable de fournir des fonctions telles que la continuité de service, la redondance, ou encore l'agrégation de ressources, et ce de façon dynamique.
De nombreuses solutions ont été proposées pour résoudre chacune des problématiques, cependant les limites des protocoles TCP et IPv4 rendent ces solutions souvent trop complexes, sous-optimisées, ou nécessitant des infrastructures réseau particu-

lières. Afin d'intégrer les concepts de mobilité et de coopération de réseaux de façon harmonieuse dans les communications entre les terminaux, il est nécessaire de revoir le modèle TCP/IP actuel dans son ensemble, en incluant des technologies protocolaires récentes. Étant donnée la complexité d'une gestion performante d'interfaces réseaux multiples et hétérogènes, il serait approprié de séparer la couche *session*, telle que définie dans le modèle OSI [29], de la couche *application* du modèle TCP/IP (Fig. 3.1). La couche *session* du modèle OSI, établit, gère et clos les connexions entre les applications. Elle initie, coordonne et termine les conversations, les échanges et les dialogues entre les applications d'extrémité. La couche session permet aussi d'insérer des points de reprise dans le flot de données de manière à pouvoir reprendre le dialogue après une panne. Toutes ces fonctionnalités sont actuellement prises en charge par l'application dans le modèle TCP/IP. L'introduction d'une couche session indépendante, adaptée à la coopération de réseaux hétérogènes, nécessite une adaptation des couches adjacentes. Les couches réseau et transport, sur lesquelles repose la couche session, doivent fournir les moyens permettant la coopération de réseaux. La couche application, doit pouvoir spécifier ses besoins à la couche session afin de tirer profit du contexte d'architecture coopérante. « L'intelligence », permettant d'opérer la coopération de réseaux à partir des moyens disponibles (couches inférieures) et en fonction des besoins (couche supérieure) réside dans la couche session. Différents protocoles sont susceptibles de fournir les moyens nécessaires à la coopération de réseaux. Les protocoles Mobile IPv6 (MIPv6), Host Identity Protocol (HIP) et SCTP apparaissent comme de bons candidats pour définir une nouvelle architecture protocolaire. Une description détaillée des protocoles de couche réseaux IPv6 (avec l'extension de mobilité), de couche réseaux/transport (couche 3,5) HIP, et de couche transport SCTP met en évidence la nécessité de remplacer les protocoles actuels TCP/IP afin de fournir les mécanismes nécessaires à la coopération de réseaux.

3.2 Les atouts du protocole IPv6

La question d'une nouvelle version du protocole IP s'est posée au début des années 90 avec la pénurie des adresses IPv4 disponibles. Des solutions à court terme, comme le routage Classless Internet Domain Routing (CIDR) ou la translation d'adresse, ont été mises en place. Malheureusement ces solutions ont entraîné un autre problème, un risque de saturation de la mémoire disponible pour maintenir les tables de routage dans les routeurs principaux d'Internet. Des groupes de travail de l'IETF sont alors

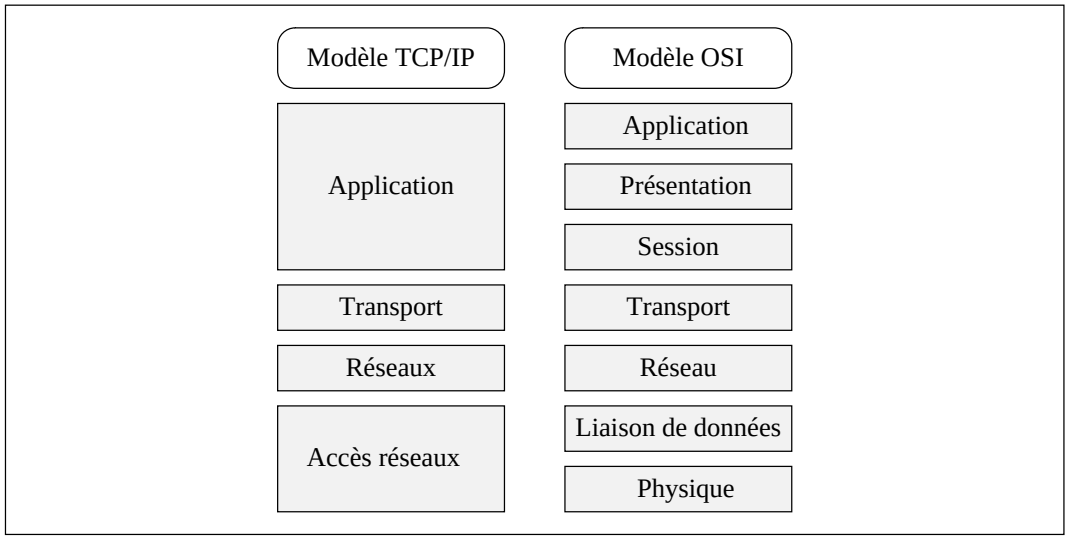

FIG. 3.1 – Modèles OSI et TCP/IP

chargés de définir et de proposer une nouvelle version du protocole IP. Les premières implémentations du protocole sont développées suite à la publication de la RFC 1752 [30] en janvier 1995. Le protocole IPv6 est aujourd'hui spécifié par la RFC 2460 [31]. L'annexe 3 page 151 fournit un rappel succinct du protocole IPv6. La section ci-dessous ne présente que les deux fonctionnalités d'autoconfiguration et de mobilité qui contribuent à la résolution de la problématique.

3.2.1 De nouvelles fonctionnalités

Le format des paquets et l'adressage du protocole IPv6 ont été définis en prenant en considération les expériences du protocole IPv4. Les erreurs du protocole IPv4 ont été corrigées et le format des paquets a été optimisé pour leurs traitements dans les routeurs intermédiaires. Cette nouvelle version du protocole IP résoud de nombreuses problématiques inhérentes au protocole IPv4 et offre la possibilité d'intégrer de nouvelles fonctionnalités apportant des avancées majeures par rapport à IPv4. La mobilité IPv6, reposant principalement sur les mécanismes de configuration automatique, apporte une solution innovante pour la connectivité des terminaux Internet qui tendent de plus en plus à devenir nomades ou mobiles.

La configuration automatique

Le protocole de découverte des voisins permet à un équipement de s'intégrer dans un environnement local et de gérer ainsi les dialogues avec les équipements connectés au lien local (hôtes, routeurs). Le protocole réalise les différentes fonctions suivantes :

- la résolution d'adresse, proche de Address Resolution Protocol (ARP) d'IPv4, mais reposant uniquement sur des messages Internet Control Message Protocol (ICMP)v6,
- la détection d'inaccessibilité des voisins,
- l'indication de redirection, utilisée lorsqu'un routeur connaît une route meilleure pour aller à une destination.

Avec IPv6, la configuration des interfaces réseaux est automatisée, ce qui signifie qu'une machine obtient toutes les informations nécessaires à sa connexion à un réseau local IP sans intervention humaine. Le processus d'autoconfiguration d'adresse d'IPv6 comprend la création d'une adresse lien-local, la vérification de son unicité, et la détermination de l'adresse unicast globale. IPv6 spécifie deux méthodes d'autoconfiguration pour l'adresse unicast globale :

- l'autoconfiguration sans état, utilisée quand une gestion administrative des adresses n'est pas nécessaire.
- l'autoconfiguration avec état, utilisée pour un contrôle strict des règles d'attribution des adresses (Dynamic Host Configuration Protocol (DHCP)v6).

L'autoconfiguration utilise plusieurs fonctionnalités du protocole de découverte des voisins :

- découverte des routeurs,
- découverte des préfixes réseaux grâce aux annonces faites périodiquement par les routeurs,
- détection d'adresses dupliquées,
- découverte de paramètres, uniquement pour une autoconfiguration avec état.

Dans un premier temps, l'hôte crée son adresse *lien local* en ajoutant son identifiant de 64 bits au préfixe réseau FE80 : :/64. L'hôte utilise ensuite l'algorithme de détection d'adresse dupliquée pour vérifier l'unicité de l'adresse. Si celle-ci est unique, la machine est en mesure de communiquer avec les autres machines du lien, sinon l'autoconfiguration s'arrête et une intervention humaine est nécessaire. Par la suite, le terminal cherche à acquérir un message d'annonce du routeur afin de déterminer la méthode d'autoconfiguration (avec ou sans état). Dans le cas d'une autoconfiguration sans état, la station concatène le préfixe réseau contenu dans le message d'annonce du routeur avec son identifiant de 64 bits pour construire son adresse unicast globale.

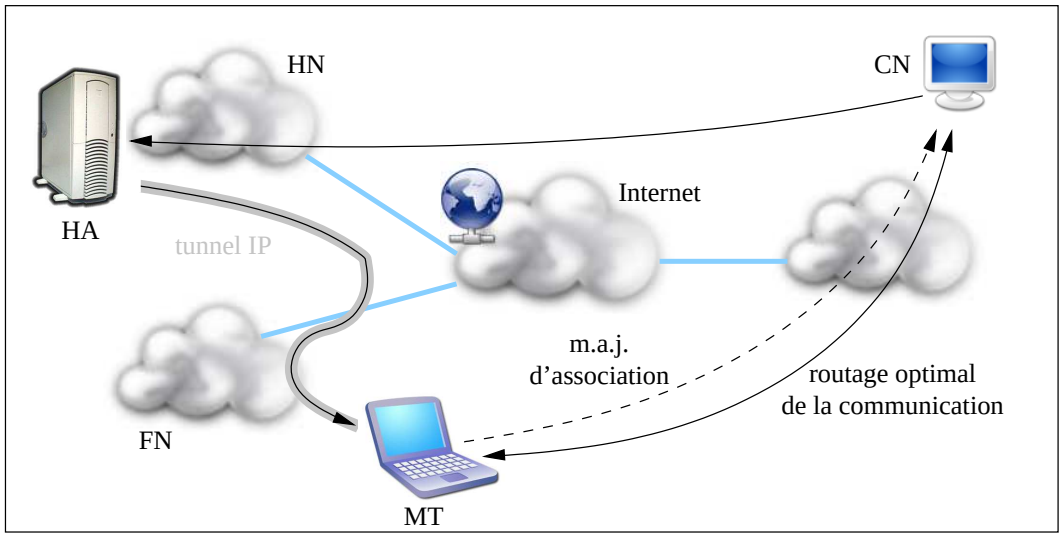

FIG. 3.2 – Initialisation d'une communication IPv6 avec un terminal mobile dans un réseau étranger

Avec le protocole IPv6, un hôte du réseau est donc capable de détecter un changement dans la configuration du réseau local, et de se reconfigurer automatiquement pour communiquer avec n'importe quel hôte distant sur Internet. Cependant le changement de l'adresse IPv6 du terminal, afin de s'adapter à son nouvel environnement réseau, ne lui permet pas de conserver les communications en cours. De même, si le terminal peut toujours avoir accès à Internet à partir d'un réseau étranger, celui-ci n'est plus joignable avec son adresse IPv6 d'origine. Les mécanismes de mobilité IPv6 ont été développés afin qu'un terminal puisse être en mesure de changer de réseau local IPv6 sans perdre les communications en cours, et reste joignable quelque soit son réseau d'accès.

La mobilité IPv6

La véritable mobilité IP propose et définit des concepts qui permettent à leurs utilisateurs de pouvoir rester connectés à l'Internet. Cela implique qu'ils puissent automatiquement obtenir une adresse quand ils se trouvent sur un réseau IP, mais aussi que les autres utilisateurs puissent les joindre à cette nouvelle adresse. Le concept de mobilité peut être étendu pour prendre en compte de manière automatique le déplacement des utilisateurs. Cette solution permet, par exemple, sous réserve de la disponibilité d'une infrastructure sous-jacente, à un utilisateur quelconque de pouvoir rester connecté à l'Internet pendant la durée de son trajet en train, en avion...

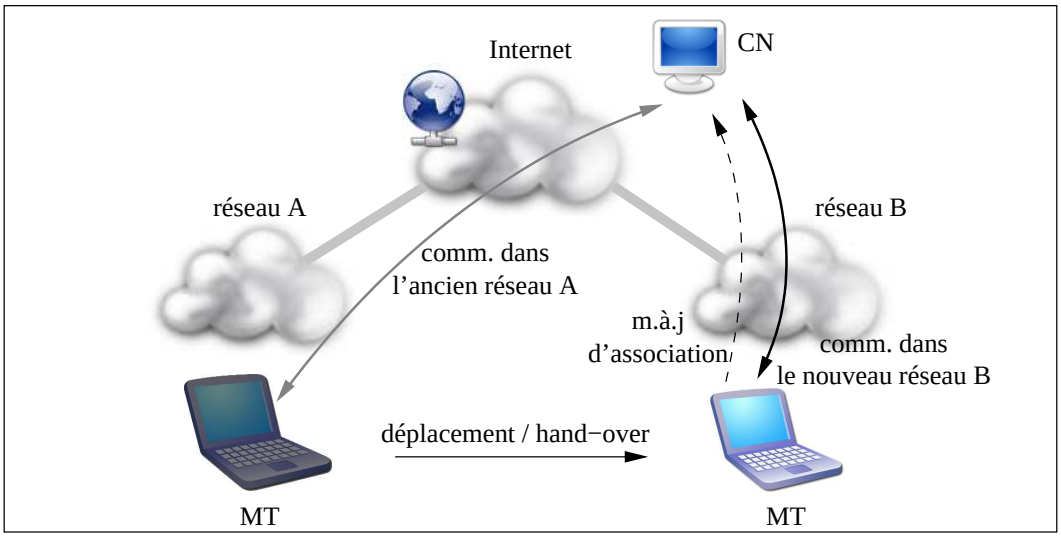

FIG. 3.3 – Hand-over durant une communication

Un mobile (Mobile Terminal (MT)) est toujours adressable par son adresse principale (adresse mère ou Home Address (HoA)) qu'il soit rattaché à son réseau d'origine (Home Network (HN)) ou à un réseau étranger (Foreign Network (FN)).
Lorsqu'un mobile est attaché à un sous-réseau étranger, il est aussi joignable par une ou plusieurs de ses adresses temporaires, en plus de son adresse mère. Les paquets adressés à une adresse temporaire seront routés à la position actuelle du mobile. La liaison entre l'adresse mère du mobile et une adresse temporaire est appelée association. Lorsqu'un mobile se déplace, il enregistre son adresse temporaire, comme adresse temporaire primaire auprès d'un routeur de son réseau d'origine, appelé agent mère (Home Agent (HA)). Par la suite, l'agent mère du mobile agira comme proxy pour intercepter tous les paquets IPv6 destinés à l'adresse mère du mobile et les transmettra, à travers un tunnel, à l'adresse temporaire primaire du mobile. Le terminal mobile reste ainsi toujours joignable.
Les nœuds correspondant (Correspondant Node (CN)) avec un mobile maintiennent une table des associations afin de router directement les paquets à destination de l'adresse temporaire primaire du mobile, si celui-ci est dans un réseau étranger, sans passer par l'agent mére.
La figure 3.2 illustre l'initialisation d'une communication vers un terminal mobile (MT). Le nœud correspondant (CN) ne connaissant pas la nouvelle adresse IP du terminal, celui-ci envoie les paquets vers son adresse mère. L'agent mère (HA), connaissant l'adresse temporaire primaire du terminal, intercepte les paquets et les retrans-

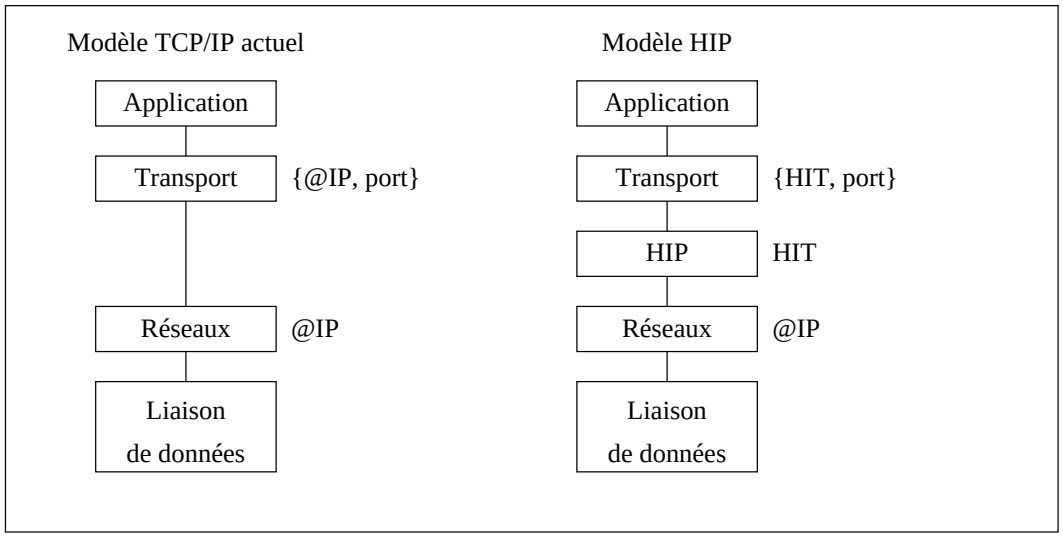

FIG. 3.4 – Comparaison des modèles TCP/IP et HIP

met vers MT à travers un tunnel IP. Le terminal renvoie un message de mise à jour d'association vers CN. Le CN met alors sa table d'association à jour, et communique alors directement avec MT sans passer par HA. Le routage est ainsi optimisé, le routage triangulaire (CN → HA → MT) est supprimé.
La figure 3.3 présente un hand-over IPv6 (changement de réseau IPv6) durant une communication. Dans un premier temps le terminal détecte le changement de réseau et lance la procédure d'autoconfiguration. Une fois configuré, il transmet un message de mise à jour d'association au nœud correspondant afin de lui préciser sa nouvelle adresse IPv6. CN envoie alors ses nouveaux paquets à la nouvelle adresse.
Dans tous les cas de figure, le terminal transmet son adresse mére aux CN. Le CN prend alors l'adresse mère comme adresse de référence, et établit toutes les communications avec cette adresse. De cette façon les sessions TCP et UDP, qui se définissent par le quadruplet {adresse source ; adresse destination ; port source ; port destination} restent inchangées quelle que soit l'adresse temporaire primaire de MT. La continuité de session, et par conséquent la continuité de la communication est donc assurée quelque soit le réseau d'accès du mobile.

3.3 HIP : un protocole de couche 3,5

Le protocole HIP[33][34], se situe entre les couches réseau (niveau 3) et transport (niveau 4) dans le modèle OSI, ou TCP/IP. Ce protocole est donc qualifié de couche

3,5. HIP propose un nouvel espace de nommage, en introduisant un identifiant d'hôte ou Host Identifier (HI), en plus des noms Domain Name System (DNS) et des adresses IP, couramment utilisés pour identifier des hôtes d'Internet [33], [34]. Les adresses IP actuelles ont deux fonctions :

- la localisation topologique de l'entité réseaux,
- l'identification de l'entité réseaux.

HIP propose de découpler ces deux fonctions en laissant l'adresse IP jouer le rôle de localisateur topologique, en permettant de gérer l'adressage et le routage des données, c'est-à-dire leur acheminement via le réseau, comme défini dans le modèle OSI. L'identification de l'hôte est quant à elle assurée par le HI représenté par le Host Identifier Tag (HIT) (Fig. 3.4).

3.3.1 Les identifiants d'hôtes

Les fonctionnalités duales de l'adressage IP actuel limite la flexibilité de l'architecture d'Internet, comme la renumérotation. De plus les couches de niveau transport actuelles (TCP et UDP) sont directement liées aux adresses IP, induisant ainsi de nombreux problèmes dans un contexte de mobilité ou de multi-accès. Le HI est généralement de nature cryptographique et est représenté par la clé publique d'une paire de clés asymétriques. La nature de la clé cryptographique pouvant varier selon les hôtes, ceux-ci n'utilisent pas directement le HI comme identifiant, mais une balise de 128 bits appelée HIT. Le HIT est une représentation de l'identifiant de l'hôte calculée à partir du HI. La longueur fixe du HI facilite son intégration dans les mécanismes d'encapsulation protocolaire des paquets IP. Il existe plusieurs versions de HI qui ne seront pas détaillées ici. HIP peut également utiliser un identifiant local, le Local Scope Identifier (LSI). Cet identifiant est codé sur 32 bits et afin de conserver son unicité, celui-ci est limité à un usage local. L'avantage du LSI est sa longueur, identique à celle d'une adresse IPv4. Le LSI peut donc être facilement intégré dans des APIs protocolaires existantes.

3.3.2 Interface avec la couche transport

Avec TCP/IP, les communications entre les applications des hôtes distants, pour un protocole donné, sont identifiées par le quadruplet : {adresse IP source ; adresse IP destination ; port source ; port destination}. Ces paramètres définissent une session unique et permettent un multiplexage des communications. Cependant une modification de l'un de ces paramètres, en cours de communication, entraîne inexorablement l'interruption de la communication. Des protocoles comme Mobile IP ou Mobile IPv6,

utilise des mécanismes de tunnel ou de substitution d'adresses afin de conserver les paramètres de session en situation de mobilité. Le protocole HIP offre un degré d'abstraction supplémentaire en remplaçant les adresses IP du quadruplet par les HIT : {HIT source ; HIT destination ; port source ; port destination}. Le protocole HIP assure l'encapsulation des paquets HIP dans les paquets IP.

3.3.3 Initialisation d'une communication avec HIP

L'hôte initialisant les échanges HIP est appelé « initiateur » et son correspondant « répondeur ». L'initialisation d'une communication HIP se traduit par l'échange de quatre paquets. Dans un premier temps, l'initiateur envoie un paquet contenant son HIT au répondeur. Le répondeur renvoie alors un paquet contenant un test cryptographique que l'initiateur doit résoudre, une signature, et les paramètres de l'algorithme de Diffie Hellman. L'initiateur échange alors un troisième paquet avec la solution du test et les paramètres de l'algorithme de Diffie Hellman, ce paquet est signé. Enfin, le répondeur finalise l'initialisation en envoyant un paquet signé. Ces échanges protègent les communications de nombreuses attaques par déni de service tout en identifiant fortement les deux hôtes distants. Une fois la phase d'initialisation terminée, les paquets transportant des données peuvent être cryptés en utilisant les clés cryptographiques définies lors des quatre premiers échanges de paquets Encapsulating Security Payload (ESP).

3.3.4 Format d'un en-tête HIP

Le protocole IPv6 considère l'en-tête HIP comme une extension IPv6 indiquée par le champ *prochaine entête*. Le champ contrôle précise certaines informations relatives au paquet et aux capacités de l'hôte (la version de HIT utilisée peut être spécifiée dans ce champ). En plus d'un champ de somme de contrôle et des HIT source et destination, les paquets HIP peuvent transporter certains paramètres qui ne seront pas tous détaillés ici. Le paramètre LOCATOR contenu dans le paquet de type UPDATE, qui est nécessaire aux fonctions de mobilité et de multi-homing sera détaillé par la suite.

3.3.5 Mobilité et multi-accès avec HIP

HIP utilise des mécanismes identiques pour la gestion de la mobilité et du multi-homing [35]. Lorsque l'hôte change d'adresse IP ou ajoute une adresse IP supplémentaire, il envoie un paquet de type UPDATE à son correspondant contenant le

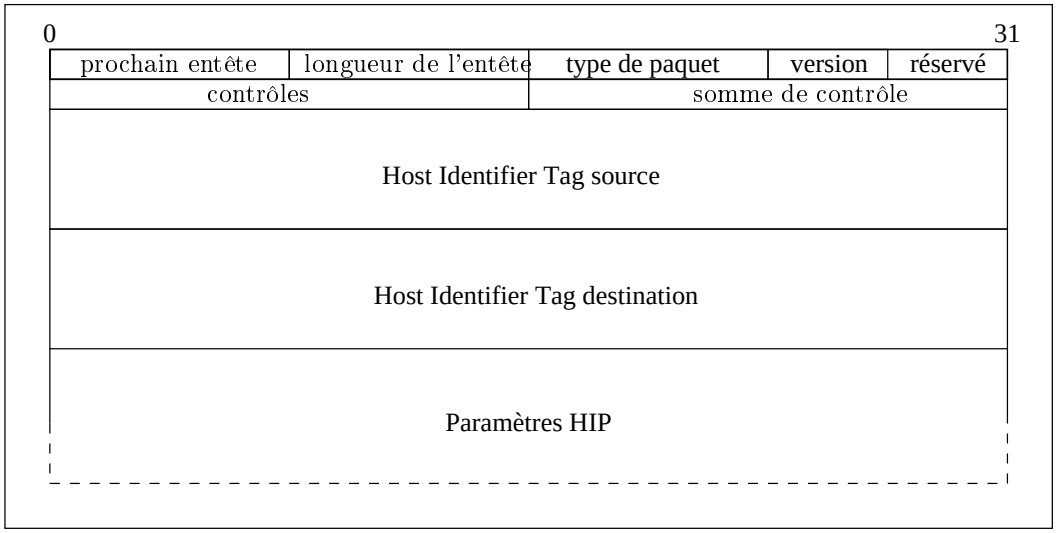

FIG. 3.5 – Format d'un en-tête HIP

paramètre LOCATOR qui spécifie l'adresse modifiée dans le cas de la mobilité ou l'ensemble des adresses disponibles dans un cas de multi-accès. Dans une situation de multi-homing, une des adresses est qualifiée de locateur (ie. adresse IP) *préféré* et sera utilisée comme adresse par défaut. Lorsqu'un hôte reçoit un message UPDATE contenant une ou plusieurs nouvelles adresses, celui-ci vérifie que les adresses IP sont bien joignables afin d'éviter un détournement éventuel du trafic vers un hôte inconnu. Durant cette période, les nouvelles adresses sont qualifiées de UNVERIFIED. Il est possible d'envoyer une quantité limitée de données, correspondant à un nombre de crédit (spécifié par l'utilisateur), vers une adresse UNVERIFIED. Si une adresse UNVERIFIED ne peut être vérifiée avant l'épuisement des crédits, celle-ci tombe dans un état INACTIVE. Le champ *type de trafic* précise si le locateur doit être utilisé pour un trafic de signalisation, de données ou les deux. Le format du locateur (adresse IPv6, IPv4-in-IPv6...) est spécifié par le champ *type de locateur*. Le bit p indique si le locateur est qualifié de *préféré* pour le type de trafic associé.

3.3.6 Le mécanisme de *Rendezvous*

Les mécanismes de mobilité décrits précédemment supposent que les hôtes sont joignables à l'origine (ie présent dans leur réseau d'origine) de la communication. Cependant le principe de mobilité exige que les hôtes soient joignables quelle que soit leur localisation topologique. L'architecture HIP introduit un nouvel élément

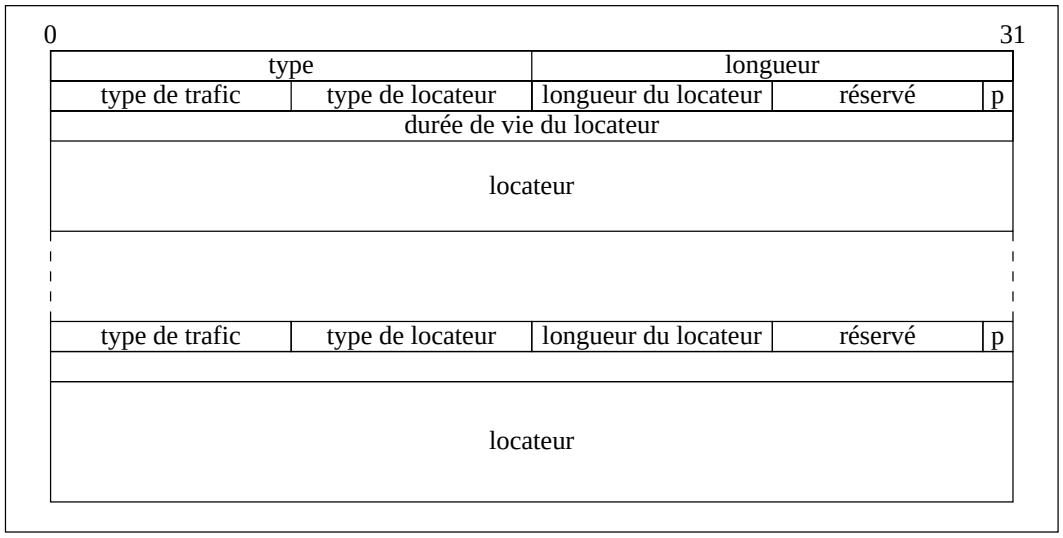

FIG. 3.6 – Format du paramètre LOCATOR d'un paquet UPDATE

statique, un serveur de *Rendezvous* dans lequel un hôte HIP enregistre ses identifiants HIT et ses adresses IP. Les DNS sont alors mis à jour afin de pointer sur le serveur de *Rendezvous*, Rendez Vous Server (RVS). En situation de mobilité, un hôte souhaitant initier une communication, reçoit donc l'adresse d'un RVS suite à une requête DNS. L'initiateur envoie alors un paquet d'initialisation au RVS, qui relaie ce paquet vers l'hôte de destination. Le répondeur renvoie alors le deuxième paquet de la phase d'initialisation, décrite ci-dessus, directement à l'initiateur de la communication. La phase d'initialisation se poursuit ensuite comme décrite précédemment.

3.4 Un protocole de couche transport mieux adapté : SCTP

3.4.1 Les origines du protocole SCTP

A la fin des années 90, le groupe SIGTRAN de l'IETF cherchait à définir de nouveaux protocoles pour traduire et adapter la signalisation Signaling System 7 (SS7) dans les réseaux IP. Le Système de Signalisation 7, SS7 est une architecture permettant d'effectuer la signalisation pour les établissements d'appels, la facturation, le routage et les fonctions d'échanges d'informations dans les réseaux téléphoniques publics commutés, Public Switching Telephone Network (PSTN). SS7 définit les fonctions à exécuter par le système de signalisation, et définit le protocole permettant

leur réalisation. Un nouveau protocole était bien sûr nécessaire, mais il n'y avait pas d'accord unanime sur celui à utiliser. Ce protocole fut initialement appelé Common Transport Protocol (CTP). Il fut d'abord essayé d'utiliser les protocoles existants comme UDP et TCP puisque ceux-ci sont présents dans tous les systèmes d'exploitation. Cependant ces protocoles ont bien vite montré leurs limites par rapport aux pré-requis du protocole CTP :

- transport de protocole de type Switch Circuit Network (SCN),
- support d'extension (évolutivité),
- contrôle de flux,
- livraison ordonnée des données,
- détection d'erreur,
- multiplexage de plusieurs communications applicatives à travers une seule communication de couche transport,
- segmentation et réassemblage des messages,
- prévention de congestion.

Face aux limitations de TCP, diverses solutions furent proposées pour implémenter CTP sur le protocole UDP :

- Reliable UDP [36],
- UDP for TCAP [37],
- Simple SCCP Tunneling Protocol [38],
- Connectionless SCCP over IP Adaptation Layer [39],
- Reliable Transport Extensions on UDP [40].

C'est en 1998 qu'une proposition de Randall R. Stewart et Qiaobing Xie, Multi-Network Datagram Transmission Protocol (MDTP) [41] retient l'attention de l'IETF. MDTP était un protocole de couche transport qui avait été conçu en prenant en considération les faiblesses de TCP. Cependant MDTP ne devint jamais un RFC. Après différentes appellations, Simple Control Transport Protocol ou Signaling Common Transport Protocol, c'est à sa neuvième version qu'il fut nommé Stream Control Transport Protocol et que sa RFC parut en octobre 2000 [42]. Un peu plus tôt en janvier 2000, le groupe de travail SIGTRAN spécifia que le protocole SCTP reposerait directement sur le protocole IP, ce qui n'avait jamais été précisé jusqu'à présent. Cette annonce souleva une certaine polémique puisque cela impliquait que SCTP pourrait être directement intégré au coeur des systèmes d'exploitation et concurencer ainsi le protocole TCP. Depuis février 2001, les discussions concernant SCTP sont faites au sein du groupe de travail TSVWG (Transport Area Working Group) de l'IETF et non plus au sein du groupe SIGTRAN. Conu à l'origine pour répondre à

des besoins de transport de signalisation de la téléphonie dans les réseaux IP, SCTP apparaît aujourd'hui comme un remplaçant très prometteur du protocole TCP en tant que couche transport dans le modèle des réseaux Internet actuels.

3.4.2 Présentation

SCTP est un protocole orienté connexion fonctionnant au dessus d'un protocole non orienté connexion comme IP. SCTP offre un service de transfert de données fiable, garanti par une gestion des erreurs à l'aide d'accusés de réception, et une non duplication de datagrammes. Les détections de corruption, de perte et de duplication des données sont effectuées par des mécanismes de checksum et de numéro de séquences. Une retransmission sélective des datagrammes est appliquée lors de la perte ou de la corruption de données.
Le design de SCTP inclut des propriétés comme la prévention de congestion et la résistance aux attaques par inondation ou masquage. La différence majeure entre SCTP et TCP est le concept de multi-homing et de flux multiples à travers une seule connexion. La notion même de connexion est abandonnée au profit de la notion d'association. Une association englobe les différentes adresses IP sources et destinations ainsi que les différents flux, intervenant dans la communication entre deux entités réseaux. Alors que TCP considère les données transférées comme un flux d'octets, SCTP les considère comme un flux de messages.
SCTP se situe au niveau de la couche transport à l'instar de TCP et UDP dans le modèle « TCP/IP ».

3.4.3 Le format des paquets SCTP

Le Protocol Data Unit (PDU) de SCTP est communément appelé paquet SCTP. SCTP fonctionne au dessus de la couche IP et par conséquent, il en constitue le payload. Un paquet SCTP est composé d'un en-tête commun et de chunks. Plusieurs chunks peuvent être multiplexés en un seul paquet. Un chunk peut contenir des informations de contrôle ou des données utilisateur.

L'en-tête commun

L'en-tête commun est constituée de douze octets (Fig. 3.7). Pour l'identification d'une association, SCTP utilise le même concept de port que TCP et UDP. Pour la détection d'erreur de transmission, chaque paquet SCTP est protégé par une somme de contrôle appelée balise de vérification (verification tag). Une balise de vérifica-

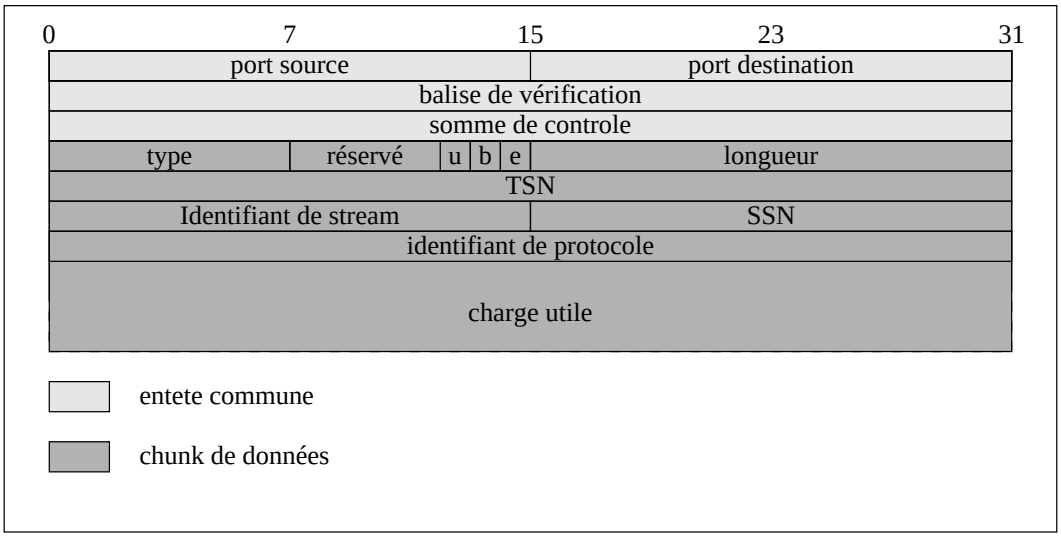

FIG. 3.7 – Format d'un paquet de données SCTP

tion est spécifique à une association et est échangée entre les terminaux source et destination lors de l'initialisation de l'association.

Les Chunks

La première partie de l'en-tête de chaque chunk représente un champ type qui est utilisé pour différencier les chunks de contrôle des chunks transportant des données. Le champ type est suivi d'un champ spécifique réservé et d'un champ longueur. Le champ longueur est nécessaire puisque les chunks peuvent être de longueurs variables. Le *payload* est positionné juste après l'entête commun. Le standard de l'IETF décrit dans le document RFC2960, définit différents types de chunk (Tab. 3.1)

Les paquets de données

En plus de l'en-tête commun, les paquets de données possèdent un champ Transport Sequence Number (TSN), utilisé pour le réassemblage des messages, un identifiant de stream, un numéro de séquence de stream Stream Sequence Number (SSN), un identifiant de protocole de la couche supérieure, (cet identifiant permet de spécifier le type d'application), et enfin les données.

Identifiant	*Type de chunk*
0	Payload Data (DATA)
1	Initiation (INIT)
2	Initiation Acknowledgement (INIT ACK)
3	Selective Acknowledgement (SACK)
4	Heartbeat Request (HEARTBEAT)
5	Heartbeat Acknowledgement (HEARTBEAT ACK)
6	Abort (ABORT)
7	Shutdown (SHUTDOWN)
8	Shutdown Acknowledgement (SHUTDOWN ACK)
9	Operation Error (ERROR)
10	State Cookie (COOKIE ECHO)
11	Cookie Acknowledgement (COOKIE ACK)
12	Reserved for Explicit Congestion Notification Echo (ECNE)
13	Reserved for Congestion Window Reduced (CWR)
14	Shutdown Complete (SHUTDOWN COMPLETE)
15 to 62	reserved by IETF
63	IETF-defined Chunk Extensions
64 to 126	reserved by IETF
127	IETF-defined Chunk Extensions
128 to 190	reserved by IETF
191	IETF-defined Chunk Extensions
192 to 254	reserved by IETF
255	IETF-defined Chunk Extensions

TAB. 3.1 – Les identifiants de chunks

Les acquitements sélectifs SACK

Ce chunk est envoyé au correspondant pour accuser réception de chunks de données, DATA CHUNK, et pour l'informer de manques éventuels dans les séquences de DATA CHUNK reçues, identifiées par leurs TSNs. Le SACK doit contenir le TSN ACK cumulatif, et le paramètre de la fenêtre de réception, Advertised Receiver Window Credit, ARWND.
Par définition la valeur du TSN ACK cumulatif est le dernier TSN reçu avant une coupure dans la réception des TSNs. Le TSN suivant n'a pas encore été reçu à l'instant où le récepteur a envoyé le TSN ACK. Ce paramètre accuse réception de tous les DATA CHUNK ayant un TSN inférieur ou égal à la valeur du TSN ACK et qui sont en séquence. Le TSN SACK peut également contenir un ou plusieurs blocs d'acquittement manquant ou Gap Ack Block (GAB). Chaque GAB accuse réception d'un ou plusieurs TSN suivant le TSN perdu. Tous les TSN acquittés par les GAB

sont par conséquent supérieurs au TSN SACK cumulatif.

Les requêtes HEARTBEAT

Un hôte de réseau peut envoyer ce CHUNK à son correspondant pour évaluer l'accessibilité d'une de ses adresses. Les requêtes HEARTBEAT sont acquittées par des HEARTBEAT ACK. Les HEARTBEAT chunks sont envoyés régulièrement (par défaut toutes les 30s). Il est ainsi possible d'obtenir régulièrement des informations sur des connexions entre deux hôtes possédant plusieurs adresses IP. Les accusés de réception des messages HEARTBEART sont toujours renvoyés à l'adresse IP source du datagramme. Les autres types d'acquittements peuvent être renvoyés à n'importe quelle adresse de l'association.

Les cookies

Ces types de chunks ne sont utilisés que dans la phase d'initialisation. La phase d'initialisation de SCTP comprend quatre échanges de messages SCTP (INIT, INIT-ACK, COOCKIE-ECHO, COOCKIE-ACK). SCTP a été conçu en tenant compte des expériences du protocole TCP. Pour rendre le protocole SCTP plus résistant aux attaques et à l'insertion de données étrangères dans une association, chaque hôte de la communication utilise un tag de vérification pour s'assurer que le paquet appartient bien à l'association. A la différence de TCP l'utilisation de cookies est rendue obligatoire dans SCTP [43].

3.4.4 Etablissement et terminaison d'une association

L'initialisation d'une association est complète après l'échange de quatre messages. Les ressources coté serveur ne sont allouées qu'après l'arrivée et la validation du troisième message, assurant ainsi à l'hôte serveur que la requête d'initialisation provient bien du bon client. Ce mécanisme rend impossible des attaques comme le « blind spoofing », ou le « SYN flooding ».
Lorsqu'un client souhaite initialiser une connexion, il envoie un INIT chunk à une addresse de transport (ensemble formé par une adresse IP et un numéro de port). Le client démarre un timer d'initialisation qui enverra des paquets INIT chunk régulièrement jusqu'à la réception d'un INIT-ACK. Si après l'envoi d'un certain nombre de paquets (nombre configurable par l'utilisateur), aucun paquet INIT-ACK n'est reçu, l'hôte distant et considéré comme injoignable. Après l'envoi du premier INIT chunk, le client attend la réception d'un COOKIE.

A la réception d'une requête d'initialisation (INIT chunk), le serveur génère toutes les configurations nécessaires pour établir une association ainsi qu'une clé secrète, (avec l'algorithme MD5 ou SHA-1). Cette clé ainsi que les informations de configuration constituent un COOKIE. Ce COOKIE est renvoyé à l'instigateur de la connexion dans un paquet INIT-ACK. Le serveur efface ensuite toutes les informations relatives à cette initialisation de connexion.
Lorsque le client reçoit le chunk INIT-ACK du serveur, le timer est annulé, et place le COOKIE dans un chunk COOKIE-ECHO qu'il renvoie au serveur. Un nouveau timer est démarré, cadençant l'envoi de chunk COOKIE-ECHO, jusqu'à la réception d'un COOKIE-ACK de la part du serveur. Si après l'envoi d'un certain nombre de paquets COOKIE-ECHO (nombre configurable par l'utilisateur), aucun paquet COOKIE-ACK n'est reçu, le serveur est considéré comme injoignable.
Le serveur analyse les informations contenues dans le paquet COOKIE-ECHO reçu. Il vérifie ensuite la validité de la clè afin de s'assurer qu'il est bien l'émetteur d'origine de ce chunk COOKIE. Si la clé est valide, le serveur renvoie un chunk COOKIE-ACK au client est considère l'association comme établie.
Le client considère l'association comme établie dès la réception du chunk COOKIE-ACK (le chunk COOKIE-ECHO peut déjà transporter des données utilisateur). Chaque extrémité de la connexion peut décider de terminer l'association. L'association peut se terminer proprement, assurant ainsi qu'aucune donnée n'est perdue, ou brutalement, en ne faisant pas attention à l'autre extrémité de l'association.
Dans le cas d'un arrêt propre de l'association, l'extrémité envoie un chunk SHUTDOWN après avoir envoyé les données restantes dans le buffer de sortie. Cette étape est renforcée par un envoi périodique de chunk SHUTDOWN, au cas où un paquet serait perdu. A la réception d'un chunk SHUTDOWN, l'autre extrémité renvoie un chunk SHUTDOWN-ACK une fois que toutes les données sont acquittées. Cette étape est également renforcée par un timer. Quand l'hôte qui est à l'origine du processus de terminaison de l'association reçoit le chunk SHUTDOWN-ACK, il arrête le timer, renvoie un chunk SHUTDOWN COMPLETE, efface toutes les informations relatives à l'association et considère celle-ci comme terminée. Si le chunk SHUTDOWN COMPLETE est perdu, le second hôte considérera l'hôte distant comme injoignable après expiration du timer.
Dans le cas d'un arrêt brutal, une extrémité peut décider d'annuler l'association en considérant que les données restantes à envoyer, et non acquittées, ne sont pas importantes et peuvent être perdues. L'initiateur envoie un message ABORT, contenant le tag de vérification, en n'incluant absolument aucune donnée dans le paquet. L'hôte

qui reçoit le chunk ABORT ne renvoie aucun message, mais vérifie la validité du tag de vérification, et notifie les couches supérieures de l'abandon de l'association si le tag est valide. En cas de perte du message ABORT, l'émetteur du message ayant déjà fermé sa connexion, l'autre extrémité ne terminera l'association qu'après un temps relativement long (une fois que le compteur d'erreur aura atteint sa valeur maximale).

3.4.5 La gestion des flux de données

SCTP doit posséder des mécanismes de gestion des flux, et de contrôle de congestion, lui permettant d'être compatible avec les versions les plus répandues de TCP. Ces mécanismes sont décrits dans la RFC 2960. SCTP distingue plusieurs streams de messages à travers une seule associaton SCTP. Le réordonnancement des paquets selon leurs numéros de séquence, ne se fait qu'à l'intérieur du même stream. Ceci réduit les phénomènes de head of line blocking entre des streams de messages indépendants et met en œuvre la notion d'ordre partiel [44]. A travers une association, le transfert fiable des datagrammes IP est assuré par checksum, un numéro de séquence, et un mécanisme de retransmission sélectif des données. Mise à part la séquence d'initialisation, tous les chunks correctement reçus sont livrés à un nouveau niveau indépendant. Ce second niveau offre un mécanisme de livraison de données flexible, basé sur la notion de plusieurs streams à travers une seule association. Afin de détecter la perte ou la duplication, l'émetteur numérote tous les chunks de données avec un identifiant appelé le Transport Sequence Number (TSN). Les accusés de réception envoyés depuis le récepteur sont basés sur ces numéros de séquences. Les retransmissions sont déclenchées par l'expiration d'un timer, dérivé de mesures régulières du RTT. Lorsque le récepteur détecte un ou plusieurs manques dans une séquence de chunks de données, chaque paquet reçu est acquitté en envoyant un Selective ACKnowledgement (SACK) qui rapporte les paquets manquants. Si l'émetteur reçoit quatre SACK consécutifs relatifs à un même paquet, l'émetteur initie un mécanisme de Fast Retransmit, et retransmet le paquet immédiatement.

Le contrôle de flux

SCTP utilise un mécanisme de contrôle de flux de bout-en-bout basé sur un système de fenêtrage similaire à TCP. Le récepteur des données peut contrôler le débit de l'émetteur en spécifiant une taille de fenêtre en octets (Receiver Window) et en renvoyant cette valeur dans tous les chunks SACK. L'émetteur, de son coté, met également à jour une fenêtre de congestion (Congestion Window, CWND) qui représente

la quantité maximale de données qu'il peut envoyer avant de recevoir un accusé de réception. Chaque chunk reçu doit être acquitté, cependant l'émetteur attend un certain temps (200ms) avant de renvoyer l'accusé. Le mécanisme d'acquittement étant cumulatif, le dernier accusé de réception envoyé acquitte tous les paquets précédents. En attendant un certain temps, il est possible de réduire le nombre d'accusés de réception en renvoyant le SACK le plus récent. Cependant une limite maximale du nombre de paquets générés par la réception d'un SACK a été fixée à 2.

Le contrôle de flux pour des interfaces multiples

Par défaut toutes les données sont transmises à une seule adresse sélectionnée parmi l'ensemble des adresses disponibles dans l'association. Cette adresse est appelée adresse primaire (Primary Address). Les retransmissions se font vers une autre adresse IP, afin de ne pas surcharger le lien vers l'adresse primaire. Les accusés de réception doivent être envoyés vers l'adresse d'où les données sont originaires.

Le contrôle de congestion

Le contrôle de congestion de SCTP est très similaire à TCP ce qui fait de lui un protocole TCP friendly. SCTP peut donc cohabiter avec des flux *TCP Reno* sur de grands réseaux comme Internet. Bien qu'étant similaire à TCP-Reno, les mécanismes de contrôle de congestion ont été adaptés pour le multi-homing. Pour chaque adresse de destination, SCTP gère un ensemble de paramètres pour le contrôle de congestion ; ainsi une association peut être perçue comme un ensemble de connexions TCP, chacune correspondant à une adresse IP.

Les différentes phases du contrôle de congestion

Comme dans TCP, SCTP possède deux comportements différents vis-à-vis du contrôle de congestion, la phase slow start et la phase congestion avoidance. Durant la phase de Slow Start, la fenêtre de congestion de l'émetteur, CWND, augmente d'un MSS (Message Segment Size) pour chaque accusé de réception reçu, ce qui revient à doubler la taille de l'envoi précédent. La fenêtre de congestion subit donc un accroissement quasi-exponentiel jusqu'à un seuil spécifié, le slow start Threshold. Au delà de ce seuil, la fenêtre de congestion augmente de un MSS par RTT, cette phase est appelée congestion avoidance.

3.4.6 Le support multi-homing de SCTP

Le multi-homing peut être défini comme la capacité d'un hôte à être perçu à travers plusieurs adresses lors d'une même connexion. Une des propriétés les plus importantes de SCTP est son support du multi-homing, ainsi SCTP est capable de gérer plusieurs adresses IP différentes à travers une seule association. Cette caractéristique pourrait offrir de nombreuses opportunités et de nouvelles fonctionnalités pour l'amélioration des communications (tolérance aux fautes, transferts multiples en parallèle...).

Gestion des adresses lors de la phase d'initialisation

Si un client possède plusieurs interfaces IP, celles-ci peuvent être utilisées dans une association SCTP. Le client informe le serveur de l'ensemble de ses adresses dans le chunk d'initialisation (INIT chunk). Il suffit au client de ne connaître qu'une seule adresse du serveur, celui-ci renvoie l'ensemble de ses adresses dans l'accusé du message d'initialisation (INIT-ACK chunk). Par défaut, le protocole SCTP utilise l'adresse de l'interface servant à l'émission de données, comme adresse IP source. Ceci facilite l'utilisation de SCTP dans les réseaux traditionnels pouvant contenir des processus de translation d'adresse ou de sécurité (NAT, IP spoofing...)

Surveillance des liens IP

Une instance du protocole SCTP surveille l'ensemble des chemins IP contenus dans son association. Pour cela SCTP utilise des messages HEARTBEAT, qui sont envoyés périodiquement sur tous les chemins qui ne sont pas utilisés pour des transferts de données. Les messages HEARTBEAT-ACK accusent réception des messages HEARTBEAT. Chaque chemin peut être dans un état actif ou un état inactif. Si un message HEARTBEAT d'un chemin IP n'est pas acquitté, SCTP retransmet le message. Si au bout d'un nombre de fois prédéfini, le message n'est toujours pas acquitté, le chemin est considéré comme inactif. Si l'ensemble des chemins IP sont inactifs, l'association est abandonnée. Ces messages HEARTBEAT peuvent permettre de mesurer des paramètres comme le délai ou la gigue.

La sélection des chemins

A l'initialisation, une adresse de l'association est choisie comme adresse par défaut, adresse dite primaire. Les chunks de données sont envoyés avec cette adresse IP primaire. En cas de retransmission, un autre chemin IP peut être choisi, ceci peut

contribuer à ne pas surcharger un lien déjà congestionné. Afin d'assurer la mesure de RTT, les SACK sont renvoyés à l'adresse IP source du chunk de données correspondant. SCTP renvoie des messages de notification aux couches supérieures lors d'une modification de l'adresse primaire.

3.4.7 Les streams SCTP

TCP assure la bonne transmission des données et ce de façon strictement ordonnée. Ces deux fonctionnalités sont indissociables dans le fonctionnement de TCP. Le protocole SCTP sépare de façon bien distincte l'ordre de transmission des données, et le contrôle de cette transmission. SCTP offre ainsi de nouveaux modes de transmission de données, pouvant s'adapter plus facilement à des applications spécifiques : transmission en mode fiable, i.e. sans perte de paquets, mais non ordonnée, ou une transmission ordonnée, mais sans contrôle au niveau de la transmission (style UDP)).
Pour cela SCTP distingue différents flux de données (stream) à travers une seule association. La réorganisation des paquets dans le bon ordre de transmission est assurée au niveau des streams. Cette gestion par stream indépendant peut éviter dans certaines circonstances le phénomène de head of line blocking, qui apparaît notamment lors de transfert séquentiel de plusieurs fichiers à travers une seule connexion.
SCTP opère sur deux plans différents : Le premier consiste à s'assurer du contrôle de la transmission de données. Pour cela SCTP utilise des paramètres de checksum, de numéro de séquence, et un mécanisme de retransmission sélectif. Tous les chunks correctement reçus sont traités dans le second plan. Dans ce second niveau, SCTP traite des différents streams indépendamment les uns des autres et assure la réorganisation des paquets si nécessaire (transmission ordonnée ou non ordonnée).

Une transmission de données flexible

Une application reposant sur SCTP peut transmettre ses datagrammes sur un ou plusieurs streams. A l'initialisation de l'association, le nombre de streams disponible dans les deux directions est échangé entre les deux extrémités. Dans chaque stream, SCTP assigne un numéro de séquence indépendant à chaque datagramme. Ce numéro de séquence permet de réordonner les datagrammes dans le cadre d'une transmission de messages ordonnée. Avec TCP il aurait été nécessaire de créer plusieurs connexions, ce qui aurait entraîné un sur-coût en terme d'over-head, de signalisation

et de traitement au niveau applicatif.

3.5 Les fonctionnalités de la coopération de réseaux

La coopération de réseaux représente la capacité, pour un processus communiquant, à tirer avantage de la présence simultanée de plusieurs réseaux, pour améliorer le service fourni à l'utilisateur. Celle-ci peut se définir à travers quatre fonctionnalités : la redondance, l'agrégation, la sélection, et la combinaison.

3.5.1 La redondance

La notion de redondance de connectivité est très proche de la notion de mobilité, elle est en fait incluse dans le concept de mobilité. La redondance désigne la faculté de changer de type de réseaux d'accès (ie d'interface de communication) pour assurer une connectivité au niveau IP en cas de défaillance de l'interface courante utilisée et sans interruption des communications en cours. Effectivement, elle est équivalente à la réalisation de handovers verticaux ou extra-technologique. On peut classifier les hand-over, passage d'un réseau à un autre, en deux catégories :

- les hand-overs horizontaux, où l'interface de communication reste identique, mais le réseau d'accès IP change,
- les hand-overs verticaux, où les communications commutent d'une interface physique à une autre. (les réseaux IP pouvant éventuellement changer eux aussi).

Les mécanismes de mobilité intègrent en plus les hand-overs horizontaux et un système de gestion de la localisation, permettant au terminal mobile de rester joignable quelle que soit sa localisation topologique. Plusieurs protocoles sont susceptibles d'accomplir une redondance de connectivité : mSCTP, Mobile IPv6 (MIPv6), et HIP [45]. Les protocoles MIPv6 et HIP offre une *vraie* mobilité, puisqu'ils incluent des systèmes de localisation par le biais de l'agent mère pour MIPv6 et du serveur de *RendezVous* pour HIP. Ces protocoles apportent donc de façon native une redondance de connectivité au terminal. Le protocole mSCTP (mobile SCTP), qui repose sur l'extension ADDIP du protocole SCTP [46], autorisant l'ajout et la suppression d'adresse IP de façon dynamique constitue une troisième solution.

Le protocole mSCTP peut avertir son correspondant d'un changement dans la liste des adresses IP de l'association avec le message ASCONF (ASsociation CONFiguration). Ce message est acquitté par un message ASCONF-ACK. Lors de la détection d'une nouvelle adresse IP, suite à un changement de réseau d'accès et à

une autoconfiguration IPv6, le terminal informe son correspondant de sa nouvelle adresse. Une fois la mise à jour validée (ASCONF-ACK), le terminal spécifie sa nouvelle adresse comme adresse primaire. La communication bascule ainsi sur le nouveau réseau d'accès. Ensuite, l'ancienne adresse étant inactive, elle est supprimée de l'association, et l'hôte distant est prévenu par la procédure ASCONF.

Cependant, contrairement aux protocoles HIP et MIPv6, le protocole mSCTP est incapable de fournir l'ensemble des mécanismes de mobilité, puisque celui-ci ne possède pas de système de localisation. Le protocole SCTP ne peut être utilisé que dans le cas de smooth handover puisque l'hôte doit conserver son ancienne adresse IP assez de temps pour mettre à jour l'association avec la nouvelle adresse. Enfin, SCTP étant un protocole de la couche transport, l'introduction du concept de mobilité à ce niveau, ne serait pas en adéquation avec le modèle TCP/IP.

En effet, la couche réseau du modèle TCP/IP réalise l'interconnexion des réseaux (hétérogènes) distants sans connexion. Son rôle est de permettre l'injection de paquets dans n'importe quel réseau et l'acheminement de ces paquets indépendamment les uns des autres jusqu'à leur destination alors que la couche transport est responsable du bon acheminement des messages complets au destinataire. Le rôle principal de la couche transport est de prendre les messages de la couche session, de les découper s'il le faut en unités plus petites et de les passer à la couche réseau, tout en s'assurant que les morceaux arrivent correctement de l'autre côté. Cette couche effectue donc aussi le réassemblage du message à la réception des morceaux.

Ainsi, les protocoles HIP (couche réseau/transport) et MIPv6 (couche réseau) sont retenus dans l'architecture protocolaire afin de pourvoir aux fonctions de mobilité et par conséquent de redondance de connectivité.

3.5.2 L'agrégation

Les paramètres pouvant qualifier une voie de transmission sont la bande passante, le délai, la gigue et le taux d'erreurs. Par analogie avec le domaine de la thermodynamique, la bande passante peut être qualifiée de grandeur extensive. En effet, soit deux liens de communication α et β possédant respectivement une bande passante B_α et B_β, l'usage simultané de α et β offre une bande passante totale B_t égale à $B_\alpha + B_\beta$. En contrepartie, les autres paramètres peuvent être assimilés à des grandeurs intensives. Par conséquent, seules les valeurs de bande passante sont susceptibles d'être agrégées.

L'utilisation simultanée de différents liens de transmission est problématique pour les communications en mode fiable (*TCP like*). Si les liens de communication pos-

sèdent des caractéristiques différentes en terme de latence ou d'erreurs de transmission, les paquets arrivent aux récepteurs de façon désordonnée. Il n'existe actuellement aucun protocole pouvant offrir une agrégation efficace de plusieurs liens pour des transferts de données en mode fiable. Si le protocole SCTP est capable de gérer plusieurs adresses IP à travers une seule association, il n'utilise qu'une seule adresse, appelée adresse primaire pour l'envoi et la réception de paquets. Cependant, celui-ci apparaît comme le meilleur candidat pour réaliser des opérations de load sharing sur des communications IP. Si cet axe de développement est encore à l'état de recherche, deux approches différentes semblent apporter des solutions potentielles :

- CMT, *Concurrent Multipath Transfer*, qui modifie uniquement les algorithmes d'émission et de réception des paquets SCTP [47],
- LS-SCTP, *Load Sharing in Stream Control Transmission Protocol*, qui introduit de nouveaux champs dans les en-têtes des paquets SCTP [48].

La problématique du load-sharing est inhérente aux mécanismes de contrôle de flux et de congestion décrits dans la RFC2581, celle-ci est donc aussi bien applicable au protocole TCP qu'au protocole SCTP. Pour simplifier seul le trafic SCTP est pris en considération. Trois effets de bord indésirables ont été identifiés lors de transferts en load-sharing [49] :

- Un nombre important de *Fast Retransmission* inutiles,
- Une limite de l'accroissement de la CWnd (Congestion Window),
- Une augmentation importante du trafic d'accusé de réception sur la voie de retour.

Lorsque les paquets arrivent de façon désordonnée, suite à la différence entre les liens de transmission, le récepteur renvoie des messages GAB (Gap Ack Block), qui acquittent les paquets reçus et signalent les absences de paquets dans les séquences TSN. Ces discontinuités des TSNs sont perçues comme des erreurs de transmission et non comme un phénomène de congestion. Lorsque l'émetteur reçoit les GAB, il utilise alors le mécanisme de Fast Retransmit, et renvoie inutilement le paquet puisque celui-ci est en fait en transit sur un second lien. De plus les GAB ne contribuent pas à l'augmentation de fenêtre de congestion, avec le protocole SCTP, seuls les accusés de réception cumulatifs (CumAck, Acks qui acquittent les TSN suivants attendus sans discontinuité) permettent d'augmenter la CWND. Ainsi les paquets acquittés par les GAB ne contribuent pas à l'augmentation de la CWND. SCTP possède un mécanisme de DelayedAck, diminuant ainsi le trafic sur la voie de retour. Cependant cette technique ne s'applique que sur les accusés de réception cumulatifs et non sur les GAB. Pendant un transfert en load-sharing, les paquets étant fortement désordonnés,

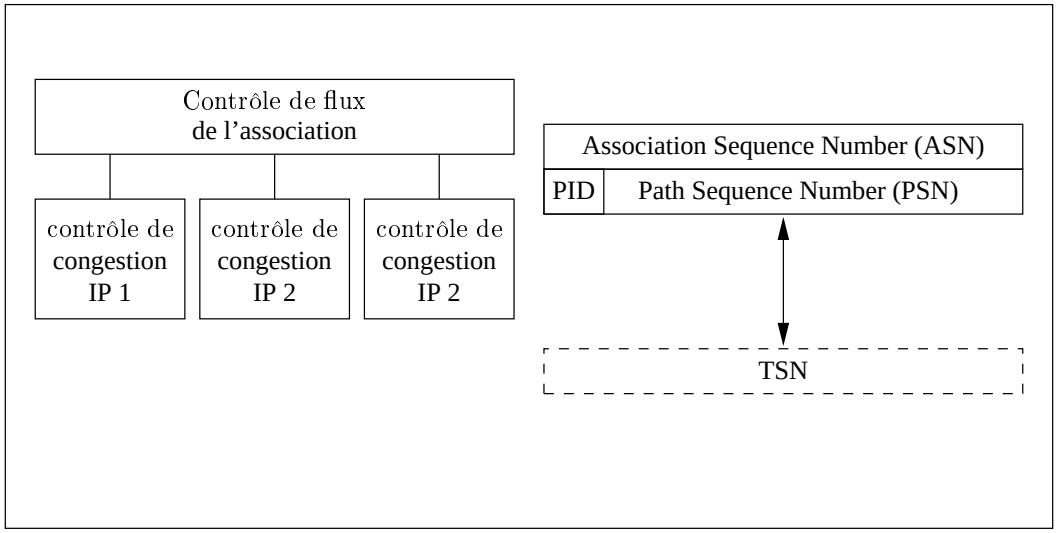

FIG. 3.8 – Architecture du protocole LS-SCTP

le récepteur renvoie principalement des GABs, rendant le DelayedAck pratiquement inopérant.

Les deux approches ne seront pas détaillées ici, nous en donnerons cependant un résumé afin de comparer ces deux solutions.

La solution CMT ne modifie pas les formats des paquets SCTP, elle introduit cependant trois nouveaux algorithmes, Split Fast Retransmit (SFR), Cwnd Update for CMT (CUC) et Delayed Ack for CMT (DAC), et de nouvelles variables d'état afin de gérer les communications concurrentes. Les variables d'état supplémentaires permettent d'associer chaque TSN envoyé avec un chemin IP, et sont utilisées par l'algorithme SFR pour éviter les Fast Retransmit abusifs, et par CUC pour avoir une mise à jour efficace de la fenêtre de congestion. CUC et SFR ne demandent que des modifications côté émetteur. L'algorithme DAC propose de retarder les GAB en plus des CumAcks, et utilise 2 bits, emprunté au champ Flag d'un chunk SACK (champ utilisé uniquement pour certain type de chunk, pour un chunk SACK, ce champ est inutilisé) pour avertir l'émetteur lorsque une phase de Fast Retransmit est nécessaire. DAC entraîne des modifications coté émetteur et récepteur.

Afin de séparer le contrôle de flux du contrôle de congestion, LS-SCTP utilise deux numéros de séquences différents (Fig. 3.8). Le premier numéro, appelé ASN (Association Sequence Number), est relatif à l'association, et est utilisé pour réordonner les chunks de données reçus dans le buffer de réception, quels que soient les chemins IP empruntés par ces données. Le second numéro appelé PSN (Path

Sequence Number) est utilisé pour le contrôle de congestion sur chaque chemin IP, identifié par le Path ID (PID). Le champ TSN, de 32 bits, est ainsi remplacé par ces trois nouveaux champs ayant une longueur totale de 64 bits. L'introduction de ces nouveaux paramètres ajoute donc un overhead de 4 octets aux chunks SCTP.

LS-SCTP offre une solution claire en séparant le contrôle de flux du contrôle de congestion. Cependant celle-ci demande la définition de nouveaux champs dans l'en-tête SCTP avec un overhead supplémentaire, et la modification des mécanismes d'initialisation avec ces nouveaux éléments. La solution CMT se base uniquement sur la définition actuelle du protocole SCTP et offre les performances attendues pour des transferts sur des liens agrégés [47].

L'agrégation de liens pour le transfert de données est donc tout à fait réalisable avec le protocole SCTP dans sa version actuelle. Les problèmes de performance dus aux asymétries entre les voies de transmission peuvent être résolus de plusieurs façons, la solution CMT étant la plus plausible. Ainsi l'introduction d'une fonctionnalité d'agrégation dans une architecture protocolaire orientée pour une coopération de réseaux est bien réalisable.

3.5.3 La sélection

Afin d'obtenir une meilleure qualité pour un service de communication, deux approches peuvent être mises en œuvre. Soit une adaptation du réseau vis-à-vis des critères de l'application, ce qui nécessite des infrastructures réseaux particulières [50][51]. Soit une sélection du meilleur réseau par le terminal, lorsque plusieurs moyens de communication sont disponibles. Bénéficiant de plusieurs réseaux à sa disposition, un terminal client d'une architecture réseaux coopérante doit pouvoir choisir le type de réseaux le plus adapté au service de communication souhaité par l'utilisateur. Cette sélection parmi les ressources réseaux nécessite :

- la spécification des besoins de l'application,
- la mesure des paramètres réseaux.

Dans un contexte de mobilité et de coopération de réseaux hétérogènes, où le nombre et les types de réseaux peuvent évoluer fréquemment, l'intégration de la sélection au niveau de l'application n'est pas envisageable. Ceci entraînerait un coût de développement important pour chaque nouvelle application. La couche transport a pour rôle de contrôler la congestion et l'intégrité du flux de données, elle n'est pas censée orientée les flux de communication selon divers critères. Cependant la couche transport est le premier niveau de la pile TCP/IP qui dispose d'un circuit logique de communication. A ce niveau il est possible d'évaluer des paramètres comme la

latence, la gigue ou la perte de paquets pour un chemin donné. Ainsi il est possible de caractériser les voies de communication indépendamment des couches sous-jacentes (mesure de signal, collisions...).

Nous avons étudié La solution d'une couche intermédiaire entre le niveau application et le niveau transport, appelée NML (Network Management Layer), est proposée afin de collecter les besoins de l'application à travers une interface prédéfinie, et de sélectionner les voies de communication à partir des paramètres procurés par la couche transport.

3.5.4 La combinaison

La combinaison de moyens dont plusieurs exemples, UDLR, HNIS, SCTPVAR, ont été analysés précédemment (§2 p. 13) autorise la répartition de la voie ascendante et de la voie descendante d'une communication sur deux réseaux différents. Cette fonctionnalité a pour principal intérêt de valoriser les liens de diffusion unidirectionnels et de proposer ainsi une nouvelle ressource, résultat de la combinaison de ressources élémentaires (UMTS, WiFi, DVB-T), plus efficace que chacune des ressources prises séparément. L'absence de notion de circuit logique au niveau de la couche réseau, autorise l'envoi et la réception de paquets sur des interfaces différentes, la reconstruction des messages et la vérification de l'intégrité de la communication (TCP like) étant à la charge de la couche transport. Cette flexibilité du routage de la couche réseau, associée à la validation de la communication par la couche transport permet la réalisation de la combinaison de réseaux. En l'absence d'infrastructure réseaux particulière (routeur d'interconnexion hybride), SCTP est le seul protocole offrant les moyens de cette coopération, par la modification de son adresse primaire (§2.2.2 p. 26).

3.5.5 Les protocoles de l'architecture

Les protocoles HIP (associé à IPv6), MIPv6 et SCTP, peuvent à eux trois fournir l'ensemble des fonctionnalités essentielles à la coopération de réseaux pour des communications unicast. Bien que les communications multicat ne soient pas traitées dans ce travail de thèse, nous pouvons noter que certains travaux proposent de développer des solutions multicast via le protocole SCTP [52], ce qui permettrait de les intégrer à l'architecture protocolaire proposée. Les caractéristiques de multihoming et/ou de mobilité de ces trois protocoles sont très intéressantes face aux problématiques liées à la coopération de réseaux. Cependant, la constitution d'une

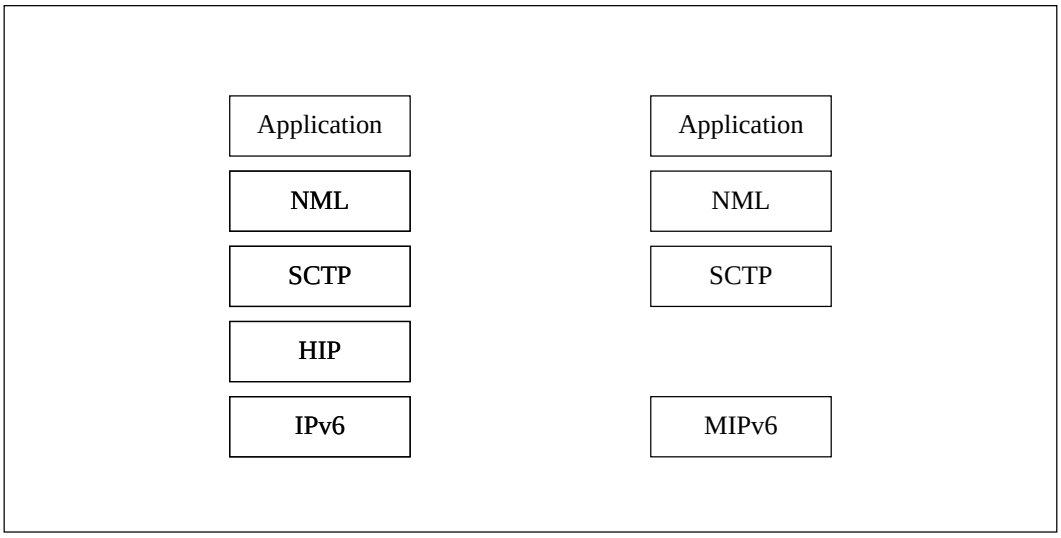

FIG. 3.9 – Modèles d'architectures coopérantes

pile protocolaire à partir de ces trois protocoles engendre certaines incompatibilités. Deux modèles peuvent être proposés (Fig. 3.9) :

- SCTP, HIP et IPv6,
- SCTP et MIPv6.

Le protocole HIP apporte une solution élégante pour implémenter la mobilité et le multi-accès en séparant l'identification et la localisation de l'hôte en utilisant respectivement un HIT et les adresses IP. Ainsi quels que soient les changements intervenant sur la ou les adresses IP, l'hôte reste toujours identifié par son HIT et les communications ne sont pas perturbées (§3.3 p. 47). HIP propose donc une couche d'abstraction supplémentaire entre la couche transport et la couche réseau. Au niveau de la couche transport, les communications sont définies par le quadruplet $\{\text{HIT}_{source}$; $\text{HIT}_{destination}$; port_{source} ; $\text{port}_{destination}\}$. La couche HIP génère donc une opacité totale entre la couche transport et la couche réseau et il est alors impossible de préciser une voie de communication particulière. Les fonctionnalités de sélection, de combinaison et d'agrégation ne peuvent donc plus être implémentées puisque chacune de ces fonctions nécessite de pouvoir spécifier une voie de communication (ie adresse IP) particulière. Aujourd'hui, le protocole HIP n'est encore qu'un domaine de recherche, et certains aspects comme le load balancing et les politiques de routage devront être spécifiés par la suite [35]. Cependant la définition de politiques de routage basées sur des critères tels que la latence, ou la perte de paquets exige la mise en place de protocoles de contrôle au niveau réseau, comme par exemple Inter-

net Control Message Protocol (ICMP). Ces mécanismes de contrôle fourniront des informations redondantes puisque celles-ci sont déjà accessibles au niveau transport grâce à la présence d'un canal logique de communication avec SCTP. De plus, l'exécution de mécanismes d'agrégation au niveau réseau nécessite impérativement une adaptation de la couche transport (problème de réorganisation des paquets). La définition de politiques de routage, basées sur des paramètres de qualité des réseaux doit donc se faire à un niveau supérieur, en recueillant les informations au niveau transport, en spécifiant l'orientation des flux sur les différentes voies de communication. Le protocole HIP rendant la couche transport totalement indépendante de la couche réseau, celui-ci ne peut pas être retenu dans une architecture protocolaire destinée à la coopération de réseaux. Le modèle proposé intègre donc les protocoles SCTP et MIPv6, la couche de gestion des réseaux Network Management Layer (NML) décrite dans le chapitre suivant repose sur cette pile protocolaire.

Chapitre 4

Proposition et modélisation d'une nouvelle couche protocolaire : NML

L'intégration des fonctionnalités de coopération de réseaux citées précédement doit offrir une gestion optimisée des ressources réseaux dans un contexte de coopération. Une implémentation au niveau applicatif nécessiterait un développement important pour chaque application, ainsi nous proposons de réaliser cette gestion de réseaux à travers une nouvelle couche protocolaire, située entre la couche transport et la couche application, appellée : Network Management Layer.
Cette nouvelle composante prend en considération les informations fournit par la couche applicative et la couche transport afin de définir la configuration de coopération à appliquer
La fonction principale du NML est donc de déterminer la meilleure coopération de réseaux possible satisfaisant les prérequis de qualité de service et/ou de facturation pour une application donnée et en fonction des ressources réseaux disponibles.

4.1 Les intéractions avec la couche applicative

Afin que le NML puisse effectuer les opérations de coopération de réseaux correctement, l'application doit fournir des informations supplémentaires relatives au service proposé à l'utilisateur.
Pour cela nous allons dans un premier temps caractériser les types d'applications les plus répandues à l'aide de contraintes qui seront transmises de l'application au NML.

Ces contraintes sont ensuite associées à des opérations de coopération de réseaux. Les échanges entre la couche applicative et le NML se font à travers une interface qui sera spécifiée par la suite.

4.1.1 La classification courante des applications

Valeur du champ TOS	*Classe d'application*
1000	minimisation du délai
0100	maximisation du débit
0010	maximisation de la reliabilité
0001	minimisation de la facturation
0000	service normal

TAB. 4.1 – Valeurs du champ TOS de IPv4

L'ancien champ Type Of Service (TOS) du protocole IPv4 définissait cinq catégories d'applications (Tab. 4.1) [53]. Cependant ce champ a été très peu utilisé dans sa définition initiale et a été remplacé par un nouvel ensemble de valeur appelé Diff-Serv Code Points (DSCP) [54]. Ce champ est désormais utilisé dans les architectures DiffServ pour appliquer différentes QoS selon les classes de trafics.
En s'inspirant de la classification précédente (TOS), nous allons définir des types d'applications auxquelles seront associées des contraintes relatives aux services fournis.

4.1.2 Les types d'applications

Les différentes applications peuvent être regroupées en deux grandes familles. Des applications que nous appellerons *symétriques* et des applications dites *asymétriques* (Tab. 4.2).
Pour les applications *symétriques*, les notions de serveur et de client, au sens émission et réception de données, sont confondues. Par exemple, des applications de VoIP ou de messagerie instantanée nécessitent autant de ressources réseaux en émission qu'en réception.
Les applications dites *asymétriques* correspondent au schéma classique des architectures client/serveur, dans lesquelles le terminal peut être soit serveur de données, et nécessite alors des ressources importantes en émission et de faibles ressources en réception, soit client et demande alors plus de ressources en réception qu'en émission.
Dans le cadre de cette thèse, le terminal est toujours considéré comme

client pour des applications *asymétriques*. L'application exprime ses caracté-

Applications	*Type*
Voix sur IP	symétrique
Messagerie instantanée	symétrique
FTP	asymétrique
Navigation web	asymétrique
Email	asymétrique
Diffusion vidéo	asymétrique
Vidéo conférence	symétrique

TAB. 4.2 – Classement des applications courantes

ristiques à travers une ou plusieurs contraintes qualifiant le service souhaité. A l'initialisation d'une communication, l'application spécifie ses contraintes au NML afin que celui-ci puisse opérer une coopération de réseaux efficace. En fonction du type de l'application, le NML prend en considération l'ensemble des réseaux disponibles (bi-directionnels et uni-diretionnels) ou uniquement les réseaux bi-directionnels. Les réseaux uni-directionnels ne sont jamais utilisés avec une application dite *symétrique* puisque celle-ci nécessite des ressources équivalentes dans les deux sens de transmission.

4.1.3 Spécification des contraintes

Les contraintes sont transmises par l'application à la couche protocolaire NML afin que celle-ci puisse éxecuter les fonctionalités de coopération de réseaux (sélection, combinaison...) adéquates pour satisfaire les besoins de l'application.

Le coût

La notion de facturation est particulièrement délicate à définir. Les différents modes de facturation possibles, facturation au temps ou à la quantité de données, les différences de coût selon les abonnements ou les périodes d'utilisation rendent l'expression d'une fonction coût pour un réseau donné quasiment impossible. Du moins cette fonction intègre des considérations économiques qui sortent du cadre de notre activité.
De plus les données de facturation sont presque toujours, à un moment donné, spécifiées par l'utilisateur, il est donc beaucoup plus simple, à la fois pour l'usager et pour l'algorithme de décision de remplacer la définition d'une fonction coût par un

paramètre de préférence qui influera la prise de décision dans la sélection des réseaux pouvant potentiellement satisfaire les critères de l'application.
Ce critère que nous appellons COST est associé à une interface réseaux et est représenté par une valeur arbitraire permettant de situer la préférence d'une interface réseaux par rapport aux autres. Le coût le plus faible correspond à la préférence la plus élevée. Le NML effectue ici une opération de sélection basée sur le critère COST.

Le délai

Le délai (DELAY) est une valeur seuil précisée par l'application afin que celle-ci puisse fournir un service de qualité suffisante (par exemple, 200ms pour une application de voix sur IP). Le NML sélectionne alors l'ensemble des réseaux ayant un délai inférieur à cette valeur seuil (une seconde sélection pouvant ainsi être opérée sur ce premier ensemble de réseaux). Si aucun réseau ne satisfait le critère DELAY, le NML sélectionne alors le réseau avec la plus faible latence. Le NML effectue ici une opération de sélection basée sur le critère DELAY.

La bande passante

L'application exprime le critère de bande passante, appelé BANDWIDTH, par un nombre β compris entre zéro et un. Ce nombre permet de définir une valeur seuil BWLIM calculée à partir de la bande passante maximale, BWMAX, assignée à une interface : BWLIM=β*BWMAX.
Le NML sélectionne ensuite l'ensemble des interfaces ayant une bande passante, supérieure ou égale à BWLIM, qui participe à une aggrégation de ressources. En effet, SCTP pouvant réaliser des opérations de load balancing, il est plus intéressant d'exploiter plusieurs interfaces réseaux pour accroître les ressources en bande passante, plutôt que de simplement sélectionner l'interface possédant la bande passante maximale. Cependant, les interfaces considérées étant fortement hétérogènes (GPRS$\simeq$ 30kbit/s, WiFi$\geq$ 5Mbit/s), le load balancing entre l'ensemble des interfaces n'apporterait pas systématiquement un gain notable (le load balancing entre GPRS et WiFi n'apporte qu'un gain de 0,6% par rapport au WiFi seul). La valeur β permet de préciser l'étendue des réseaux qui sont aggrégés, 0 pour prendre en compte l'ensemble des réseaux, 1 pour ne sélectionner que les interfaces possédant la valeur de bande passante maximale. L'opération effectuée ici par le NML est une opération de sélection, puis d'aggrégation de ressources. Il ne s'agit pas ici de prendre en compte la bande passante disponible le long du chemin IP, mais uniquement de considérer les débits offerts par chaque interface. Les valeurs réelles, par exemple dûes à des états de

congestion éventuels dans les cœurs de réseaux, ne sont pas prises en considération.

Les autres critères, gigue et taux d'erreur

Il existe de nombreux types de trafics applicatifs (*symétrique*/*asymétrique*, trafic à débit constant ou variable...) possèdant des contraintes variées et rendant une classification exhaustive des attributs applicatifs particulièrement difficile. Toutefois, des applications avec des contraintes simples peuvent facilement être associées avec des fonctionnalités de coopérations de réseaux :

- VoIP ↔ minimisation du critère DELAY ↔ sélection du réseau avec une latence minimale
- FTP ↔ maximisation du critère BANDWIDTH ↔ aggrégation des interfaces réseaux.

Cependant certaines applications possèdent plusieurs constraintes qui requièrent de multiples opérations de coopération de réseaux. Un service de vidéo-conférence a, par exemple, une contrainte sur le délai, la gigue et la bande passante. Le délai est la contrainte la plus significative pour une bonne interactivité, puis vient la gigue et enfin la bande passante puisque les données peuvent être encodées à différents débits. Avant tout le NML opère donc une sélection par rapport au critère de délai, puis une seconde par rapport au critère de gigue et enfin une dernière sélection sur la bande passante. Nous pouvons remarquer que cette application nécessiterait idéalement trois opérations de sélection successives selon trois critères différents spécifiques à cette application. De plus, nous pouvons constater que différentes opérations de coopération peuvent être appliquées pour la même constrainte. En effet, dans le cas d'un flux vidéo, une maximisation de la bande passante n'a pas de sens, la sélection d'une interface offrant une communication avec une qualité acceptable serait suffisante. Par conséquent, nous définissons un ensemble de services simplifiés afin de restreindre le domaine d'investigation concernant la caractérisation des applications. Dans cette première proposition du NML nous nous limitons aux trois critères cités

Contraintes	*Type d'application*	*Interfaces*	*Coopération*
DELAY	symétrique	bidirectionnelle	sélection
BANDWIDTH	asymétrique	bidirectionnelle ou unidirectionnelle	aggrégation
COST	—	bidirectionnelle ou unidirectionnelle	sélection ou combinaison

TAB. 4.3 – Correspondance entre services et opérations de coopération

précédemment (DELAY,COST, et BANDWIDTH) (Tab. 4.3). Ainsi des critères de sélection basés sur la gigue ou le taux d'erreur ne sont pas spécifiés dans le NML.

4.1.4 Interface de communication entre l'application et le NML

L'application caractérise ses besoins en termes de ressources réseaux au NML à travers une contrainte dite *primaire* et éventuellement d'une ou de plusieurs contraintes dites *secondaires*.
Les applications possèdant une contrainte primaire DELAY sont des applications à caractère interactif, par exemple un service de VoIP. Par conséquent ces applications sont généralement de type *symétrique* et les ressources réseaux en émission et en réception doivent être similaires. Ainsi pour une contrainte de type DELAY, les réseaux uni-directionnels ne sont pas sélectionnés par le NML.
La contrainte primaire BANDWIDTH caractérise des applications de type *asymétrique* correspondant à des téléchargements de contenu vers le terminal. Les réseaux uni-directionnels peuvent donc faire partie de l'aggrégation de réseaux demandée par la contrainte primaire BANDWIDTH.
Les applications demandant une minimisation du critère COST ne nécessitent pas de ressources réseaux particulières, le NML cherche donc à minimiser les coûts (exemple : réception d'e-mail, mise à jour de logiciel...).
Il serait également possible de spécifier d'autre contraintes secondaires, permettant pas exemple de sélectionner un ensemble de réseaux statisfaisant une contrainte primaire DELAY, et de chercher ensuite le ou les réseaux ayant le moindre coût. Dans cette proposition du NML nous nous limitons à l'emploi d'une seule contrainte primaire DELAY ou BANDWIDTH et d'une contrainte secondaire implicite COST.
En considérant le NML comme une couche protocolaire à part entière intégrée au système d'exploitation, une application pourrait communiquer avec la couche NML à travers un nouveau type de socket. Les sockets les plus communes aujourd'hui sont de type SOCK_STREAM pour le protocole TCP et de type SOCK_DGRAM pour le protocole UDP. Les informations transmises à travers ces sockets sont les suivantes :

- protocole : SOCK_STREAM ou SOCK_DGRAM
- adresse IP de destination
- port logique de destination

et éventuellement :

- port logique source
- adresse IP source

Les informations transmises à travers une socket créée au niveau du NML seraient les suivantes :

- protocole : SOCK_NML
- addresse IP de destination
- port logique de destination
- type de contrainte
- valeur de la contrainte
- type de communication : fiable, partiellement fiable, ou non fiable

et éventuellement :

- port logique source
- adresse IP source

Le NML reposant exclusivement sur le protocole SCTP, il est nécessaire de préciser le type de communication (relié ou non), qui est implicitement déclaré avec le type de protocole pour TCP et UDP.

Pour la réalisation d'un prototype, le NML a été développé comme un serveur proxy local acceptant en entrée des connexions UDP et TCP afin de récupérer le contenu des applications et utilisant ensuite le protocole SCTP pour établir la communication avec le processus distant intégrant aussi une couche NML(Fig. 4.1). Les types de

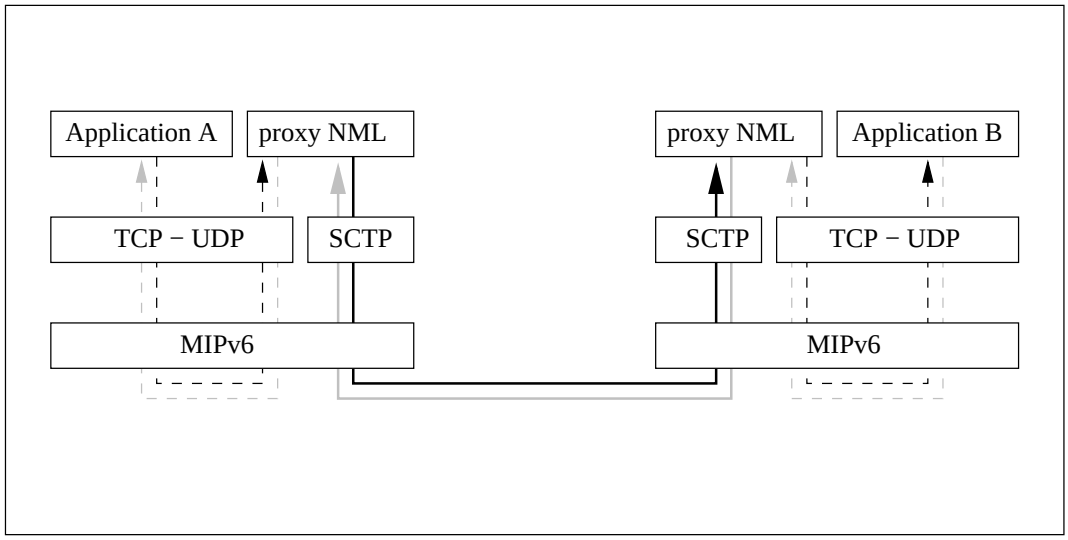

FIG. 4.1 – Proxy NML

contraintes et leurs valeurs sont associés aux ports logiques couramment utilisés par les applications (80=www,110=pop...).

Ainsi il n'est pas nécessaire de modifier les applications courantes pour exploiter les

processus de coopération de réseaux.

4.2 Interaction avec la couche transport

4.2.1 Paramètres spécifiés et paramètres mesurés

Les paramètres sur lesquels se base le NML sont fournis par la couche transport SCTP puisque celle-ci intègre un canal logique permettant de contrôler la communication de bout-en-bout.
Les critères sont certe limités, (absence de mesure de signal, de détection de cellules adjacentes...), mais ils procurent une complète indépendance vis-à-vis des couches matérielles sous-jacentes (UMTS, DVB-T, WiFi) et des infrastructures réseaux.
Certains paramètres peuvent être mesurés directement alors que d'autres paramètres nécessitent d'être précisés, partiellement ou totalement, par l'utilisateur.
La couche transport SCTP peut directement évaluer les paramètres suivants :

- le délai, fourni par le paramètre SRTT (Smooth Round Trip Time),
- la gigue, fourni par le paramètre RTTVAR (Round Trip Time VARiation),
- le taux d'erreur, fourni par le nombre de retransmission,
- le débit de transfert des données.

Le NML ne prenant pas en compte les critères de gigue et de taux d'erreur, les paramètres correspondants fournit par la couche SCTP sont donc inutiles au NML dans notre contexte d'étude.

Le paramètre de bande passante

La bande passante disponible peut difficilement être appréciée avant le début de la communication. Diverses techniques permettent de mesurer la bande passante disponible sur un chemin IP, mais celles-ci impliquent l'utilisation d'un trafic additionnel [55],[56],[57]. Ces solutions pourraient éventuellement être implémentées en utilisant les messages HEARTBEAT de SCTP.
Cependant, afin de ne pas surcharger les liens de communication la bande passante est précisée par l'utilisateur et non mesurée. Un paramètre de bande passante est donc assigné à chaque interface de communication et permet au NML de sélectionner de façon cohérente, à partir du critère BANDWIDTH, l'ensemble des interfaces qui participe à l'aggrégation de ressources.

Le paramètre de délai

Le délai sur chaque chemin IP est régulièrement apprécié par les chunks HEARTBEAT du protocole SCTP. L'obtention de ces valeurs n'entraîne donc pas de consommation de ressources supplémentaire par rapport à la version standard du protocole SCTP. Les messages HEARTBEAT sont envoyés, par défaut, toutes les trente secondes.

4.2.2 Communication avec la couche SCTP

Pour la communication avec la couche SCTP nous utilisons exclusivement les primitives décrites dans un draft défini à l'IETF [58]. La pile protocolaire SCTP n'est donc pas modifiée pour fournir les opérations de coopération de réseaux. Il est important de préciser que si le load balancing peut être réalisé avec le protocole SCTP dans sa version actuelle, ses performances sont fortement dégradées puisque les algorithmes de CMT ou de load-sharing ne sont pas implémentés (§3.5.2 p. 63).

4.2.3 Communication avec la couche MIPv6

Le protocole SCTP ne permet pas de sélectionner l'interface par laquelle le terminal transmet les paquets. Pour cela le NML doit spécifier des politiques de routage au niveau du terminal afin de commuter les paquets sur l'interface souhaitée. Plusieurs solutions, reposant sur des outils tels que Netfilter et IProute2, permettent de choisir une interface de communication en fonction de divers critères [59][60]. Cependant, elles ne permettent pas de dissocier la voie d'émission des données de la voie de réception comme nous proposons de le faire avec notre modèle (*combinaison*, *aggrégation*).
Les communications sont identifiées par leur association SCTP (port, protocole, addresses IP). Les politiques appliquées pour l'affectation de l'interface d'émission par défaut sont les suivantes :

1. Si la communication est relative à une application *symétrique* l'interface d'émission est identique à l'interface de réception des données.
2. Sinon l'interface d'émission sélectionnée est celle de plus faible coût.

Le NML initie les communications sur l'interface par défaut qui correspond à l'interface de plus faible coût. Les opérations de coopération de réseaux sont seulement appliquées après l'établissement de la connexion. La modélisation en Specification and Description Language (SDL) du NML est décrite dans la section suivante.

4.3 Description formelle

L'architecture protocolaire proposée présente une structure simple. C'est pourquoi nous limitons sa modélisation au seul niveau fonctionnel. Deux approches complémentaires ont été utilisées : l'approche interaction, modélisée avec SDL et l'approche fonctionnement interne modélisé par réseau de Petri. Les modélisations en langage SDL et en réseaux de Petri nous permettent de simuler et de valider en partie cette première proposition du NML. Le langage SDL est utilisé afin de décrire, simuler et valider les échanges inter-processus. La modélisation à l'aide des réseaux de Petri permet d'analyser et de vérifier le principe de fonctionnement du NML.

4.3.1 Réalisation d'un modèle SDL

Nous avons modélisé une communication entre une entité client et une entité serveur avec le standard SDLà travers l'outil de développement *Object Geode*. Avant tout il est important de rappeler le contexte d'utilisation de ce module de gestion des connexions réseaux. Cette solution est conçue pour être implémentée sur des hôtes d'extrémité. De nombreux réseaux sans fil, récemment déployés ou en cours de déploiement, et issus des domaines de la télécommunication (UMTS) ou de l'audiovisuel (DVB) supportent désormais des communications basées sur le protocole IP. Un utilisateur, mobile ou nomade, a donc potentiellement plusieurs réseaux de communication IP disponibles.
Le terminal utilisateur considéré intègre plusieurs interfaces de communication, UMTS, WLAN et DVB, qui représentent respectivement les trois grands domaines des réseaux de transport d'information, qui sont les réseaux de téléphonie, les réseaux informatiques et les réseaux de diffusion audiovisuels.
Le terminal utilisateur se limite à un rôle de client vis-à-vis du réseaux. Un usage « client »est défini par le fait que l'utilisateur consommera (recevra) beaucoup plus de données qu'il n'en fournira dans le cas d'applications *asymétriques*. Même dans le cas des nouvelles applications de type peer to peer (P2P), les transferts de données à destination des utilisateurs finaux restent largement supérieurs aux données transmises par les utilisateurs [61].
Par opposition, un serveur est défini comme un hôte distant dont l'activité principale est de fournir des contenus d'information. Ces hôtes du réseau sont considérés comme statiques et sont connectés à Internet par le biais de connexions fixes. Cette

configuration est totalement opposée à celle des terminaux qui sont mobiles et sont interfacés simultanément avec plusieurs réseaux hétérogènes.
Le but du NML est d'améliorer les services des usagers en sélectionnant le ou les réseaux adéquats selon les types d'applications. Les fonctions d'optimisation sont donc principalement situées sur le terminal. Le NML comporte deux modes de fonctionnement, un mode « client »qui recherche la coopération de réseau optimisée afin de fournir le meilleur service possible, et un mode serveur qui se limitera à l'application de la configuration réseaux défini au niveau du client.
Cette description SDL explicite deux points importants du fonctionnement du NML : la communication entre l'entité cliente et l'entité serveur au niveau NML, puis les interactions et les échanges entre les différentes couches systèmes et protocolaires pour les hôtes d'extrémité (client et serveur). L'annexe 3 propose un rappel succinct des termes employés dans le langage SDL.

Modélisation de la communication

Le système principal est composé de deux blocks représentant une entité cliente (`Terminal`) et une entité serveur (`Server`) (Fig. 4.2). Ces deux entités communiquent à travers deux canaux logiques de communication, le canal `IPcx` et le canal `Stream0`. Le service de la couche applicative est simulé par une application cliente envoyant une requête à l'application du serveur. Le serveur envoie alors des données, répondant ainsi au service souhaité. Une fois la tâche accomplie, le serveur initie la fermeture de la connexion. Nous n'avons pas modélisé l'envoi et la réception des paquets de données mais uniquement la signalisation entre les processus NML.
`IPcx` représente la connectivité IP, c'est sur ce canal que sont échangés les paquets SCTP d'intialisation. L'initiation de la connexion SCTP (INIT, INIT-ACK, COOKIE-ECHO et COOKIE-ACK), est simplifiée par les seuls messages `HandShakeStart`, émis par le `terminal` vers le `Server`, et `HandShakeStop`, renvoyé par le `Server` vers le `Terminal`.
SCTP supportant le multi-streaming, nous avons choisi arbitrairement le stream avec l'identifiant 0 pour les échanges de messages entre les processus NML client et serveur, les données étant transférées sur les autres streams. Les NML communiquent donc directement en utilisant le protocole SCTP. Les mécanismes d'encapsulation n'étant pas modélisés, le canal logique `Stream0` est utilisé pour représenter la communication SCTP entre les NML.
Les deux blocks peuvent recevoir des signaux de l'extérieur du système afin de transmettre des informations qui ne dépendent pas du système lui-même (nombre d'in-

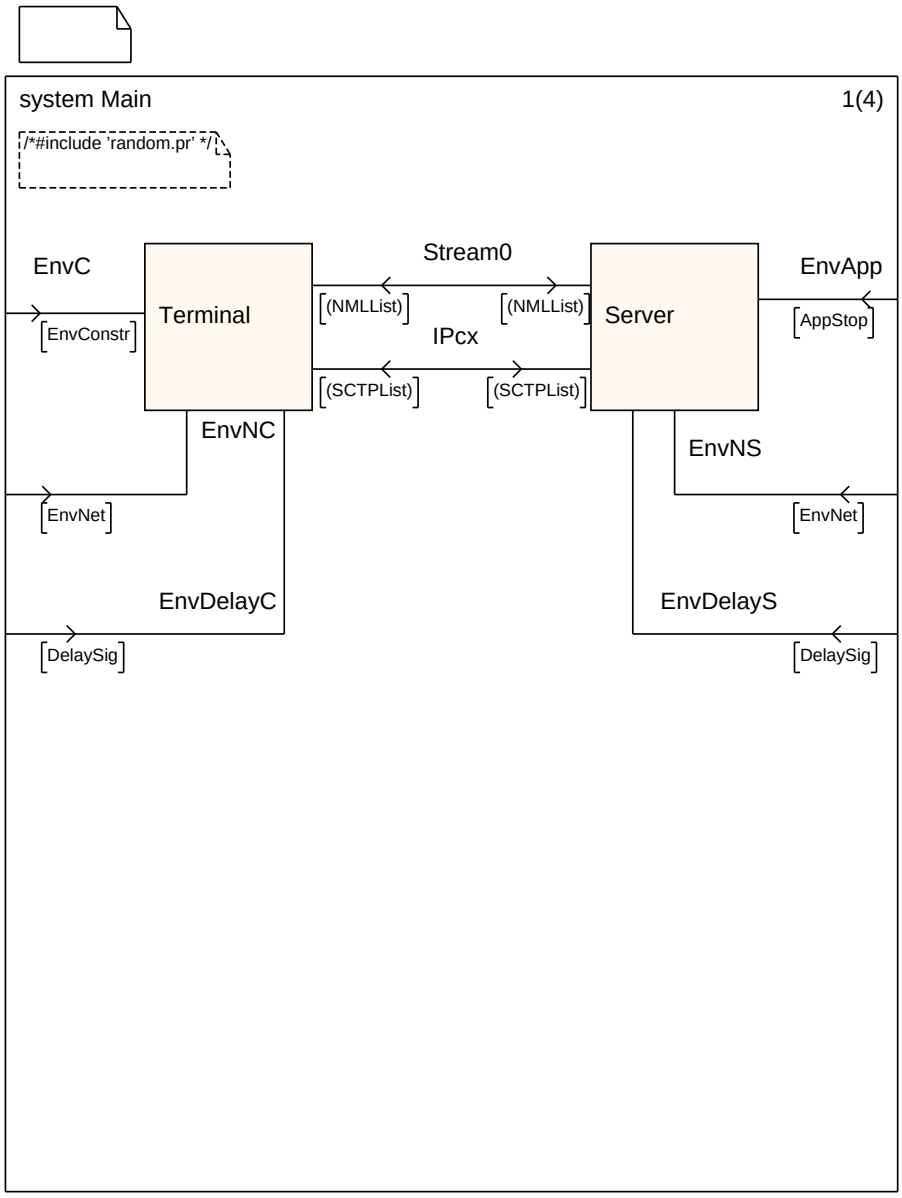

FIG. 4.2 – Modèle SDL de la communication

terfaces réseaux, type d'interface, delai...).
Pour le `Terminal`, la porte `EnvC` spécifie la contrainte définie par l'application. La porte `EnvNC` spécifie les caractéristiques des interfaces réseaux (nombre, bande pas-

sante, unidirectionnelle...). Les blocks `Terminal` et `Server` utilisant des processus NML de même type, la porte `EnvDelayC` est connectée au `Terminal`, mais celle-ci n'est pas utilisée.
Pour le `Server`, la porte `EnvApp` permet d'envoyer un signal `AppStop` pour arrêter l'application serveur. La porte `EnvDelayS` introduit les valeurs des délais mesurées sur les chemins IP par le protocole SCTP. La porte `EnvNS` n'est pas utilisée par le block `Server`.

Modélisation des hôtes réseaux

Chaque block est composé de quatre processus. Les processus applicatifs sont différents pour le serveur et pour le client. Le processus `AppClient` est de type `AppClientT` et le processus `AppServer` est de type `AppServerT`. Les trois autres processus du serveur et du client sont respectivement issus d'un même type (`NMLClient` et `NMLServer` sont de type `NMLT`, `NetworkStatus` est de type `NetworkStatusT`, `SCTPClient` et `SCTPServer` sont de type `SCTPT`).
Ormis les applications, qui permettent de déterminer le rôle serveur et le rôle client, les entités `Terminal` et `Server` utilisent donc des processus identiques qui constituent une pile protocolaire intégrant un module de gestion des ressources réseaux (NML). Chaque processus de cette pile a donc été décrit dans son ensemble comprenant à la fois la partie cliente et la partie serveur.

Les processus communs de la pile protocolaire

Le processus `NetworkStatusT` Ce processus répond à une requête du NML et informe celui-ci de l'état des interfaces réseaux disponibles. Les informations incluses dans le signal de réponse sont les suivantes :

- nombre de réseaux disponibles,
- bande passante de l'interface réseau,
- paramètre de coût,
- type d'interface, bidirectionnelle ou unidirectionnelle.

Le processus `SCTPT` Ce processus représente le protocole SCTP. Nous n'avons pas modélisé l'ensemble des fonctionnalités de SCTP, mais uniquement celles qui interagissent directement ou indirectement avec le processus NML.
La représentation Message Sequence Charts (MSC) (Fig. 4.5) montre les échanges entre la couche NML et la couche SCTP lors de l'initialisation d'une communication.

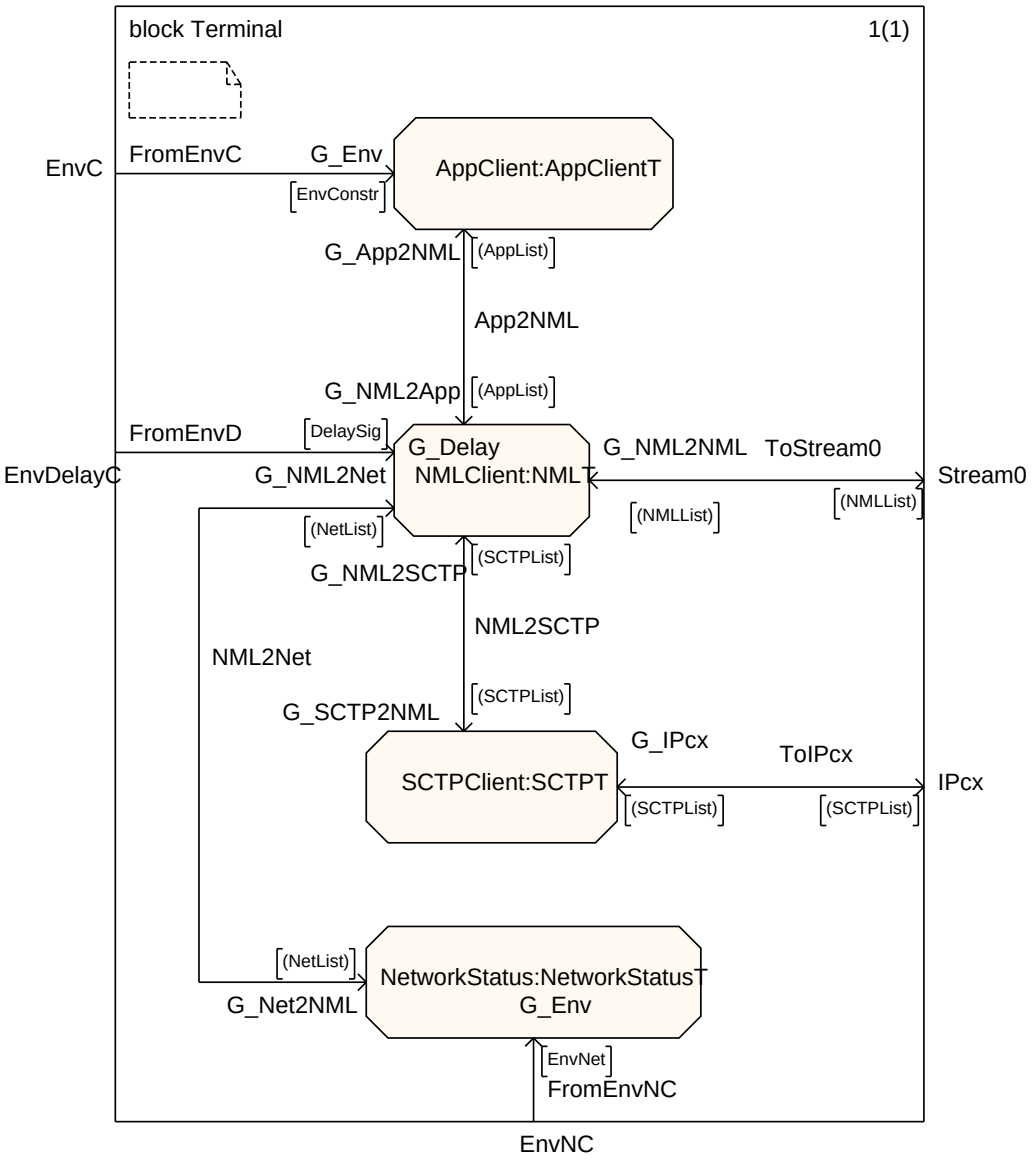

FIG. 4.3 – Modèle SDL du terminal

`NMLClient` envoie un signal `SCTPInit` à `SCTPClient`. Ce signal correspond à la création d'une `socket` SCTP. Les processus `SCTPClient` et `SCTPServer` établissent leur connexion à travers les échanges de messages `HandShakeStart` et `HandShakeStop`.

`SCTPClient` renvoie alors un signal `SCTPInitOk` à `NMLClient` précisant ainsi que la connexion au niveau SCTP est établie et fonctionne correctement. `NMLClient` transmet ensuite un signal `NMLInit`, auquel `NMLServer` répond par un `NMLInitOk` et attend la configuration à appliquer pour la communication. Cette configuration est spécifiée par les signaux `NMLCfg` et `NMLCfgOk`. Le protocole SCTP fournit également les mécanismes essentiels pour réaliser les fonctions de coopération de réseaux comme la sélection, la combinaison, l'agrégation ou la redondance. C'est à travers ce processus qu'est appliquée la coopération de réseaux pilotée par le processus NML .

Le processus `NMLT` Ce processus représente le pivot central pour la coopération de réseaux et communique directement avec les trois autres processus. `NMLT` gère les communications réseaux à travers le processus `SCTPT` selon la contrainte spécifiée par l'application et en fonction des ressources disponibles fournies par le processus `NetworkStatusT`.

Les figures 4.5, 4.6 et 4.7 montrent les échanges de signaux entre le processus `NMLT` et les autres processus lors de la simulation d'un transfert de données du serveur vers un terminal possédant trois interfaces réseaux et avec une contrainte DELAY.
Simulation :

1. Le processus `NetStatus` obtient une liste de réseaux avec leur caractéristiques (coût, bande passante, type, nom des réseaux), signal `NetList`.
2. L'application reçoit un signal `Constraint` contenant le type de contrainte ainsi que sa valeur, et le transmet au `NMLClient` à travers le signal `AppInit`.
3. Le processus `NMLClient` interroge alors le processus `NetworkStatus` via le signal `NetReq` et attend la réponse `NetResp`.
4. A la réception du signal `NetResp`, le `NMLClient` ouvre une `socket` SCTP et se connecte à l'hôte distant : signaux `SCTPInit, HandShakeStart, HandShakeStop`.
5. Une fois la connexion SCTP établie (réception du signal `SCTPInitOk`), le `NMLClient` transmet sa configuration au `NMLServer` : signaux `NMLInit, NMLInitOk, NMLCfg`. Le signal `NMLCfg` contient la liste des adresses IP relatives aux interfaces participant à la coopération de réseaux, ainsi que le type de contrainte et la valeur associée.
6. `NMLServer` se connecte à l'application serveur (signal `AppStart, AppStartOk`) et renvoie un signal `NMLCfgOk`.

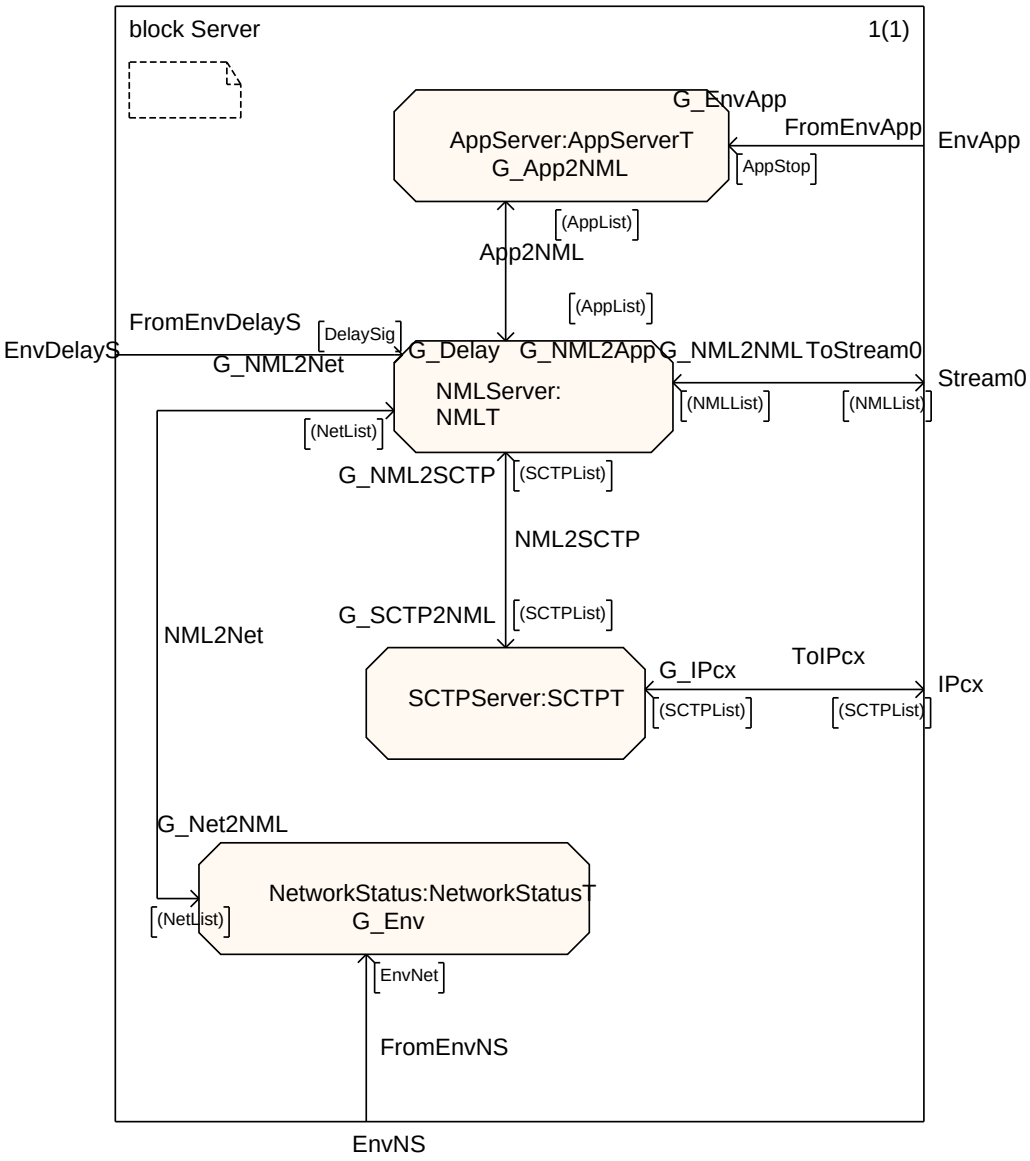

FIG. 4.4 – Modèle SDL du serveur

7. La contrainte appliquée est ici de type DELAY, le processus `NMLServer` atteint donc un état `DelayProcessing`. Les mesures de délai sur les divers chemins IP sont fournies au NML avec les signaux `DelaySig`.

8. Une fois l'adresse IP destination sélectionnée, désignant l'interface réseau avec la plus faible latence, le `NMLServer` oriente le flux d'information vers cette interface réseau grâce au processus `SCTPServer` et au message `SetPrimAddr`.
9. Au bout d'un certain temps, l'envoi du signal `AppStop` entraîne la fermeture de l'application, un message shutdown est donc envoyé aux processus `NMLServer, SCTPServer, SCTPClient, NMLClient` et enfin `Appclient`.

L'exemple précédent nous a permis de détailler les interactions entre les processus au sein d'un hôte d'extrémité et les échanges de messages entre les entités réseaux d'extrémité. Les principaux signaux transportent les informations suivantes :

`AppInit`
- Type de contrainte
- Valeur de la contrainte
- Adresse IP destination (non représentée dans le modèle)
- Port destination (non représenté dans le modèle)
- Fiabilité de la communication (non représentée dans le modèle)

`NetResp`
- Nombre de réseaux
- Adresse IP des réseaux
- Coût des réseaux
- Bande passante
- Type de réseaux

`NMLCfg`
- Nombre d'interface réseaux
- Coût
- Bande passante
- Type de contrainte
- Valeur de la contrainte

Validation à l'aide de scénarii

Nous avons effectué plusieurs simulations avec différents scénarii afin de vérifier le bon fonctionnement du modèle pour les principaux cas de figures. Dans cette modélisation, nous considérons également le critère coût dans la sélection des interfaces de plus faible latence. Une fois les interfaces satisfaisant la contrainte de latence sélectionnée, le NML sélectionne celle(s) ayant le plus faible coût. Pour une contrainte BANDWIDTH, le coût n'est pas pris en considération. Si aucune contrainte n'est spécifiée, le NML effectue par défaut une sélection par rapport au coût.

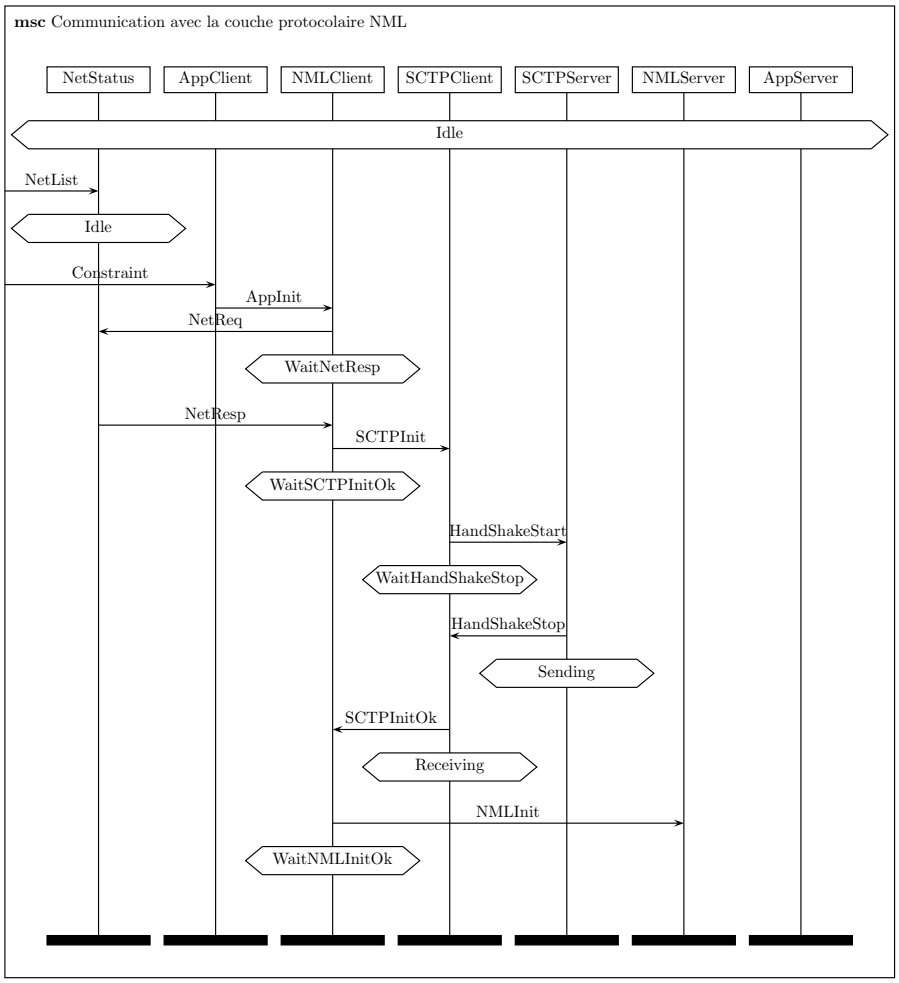

FIG. 4.5 – Message Sequence Chart 1

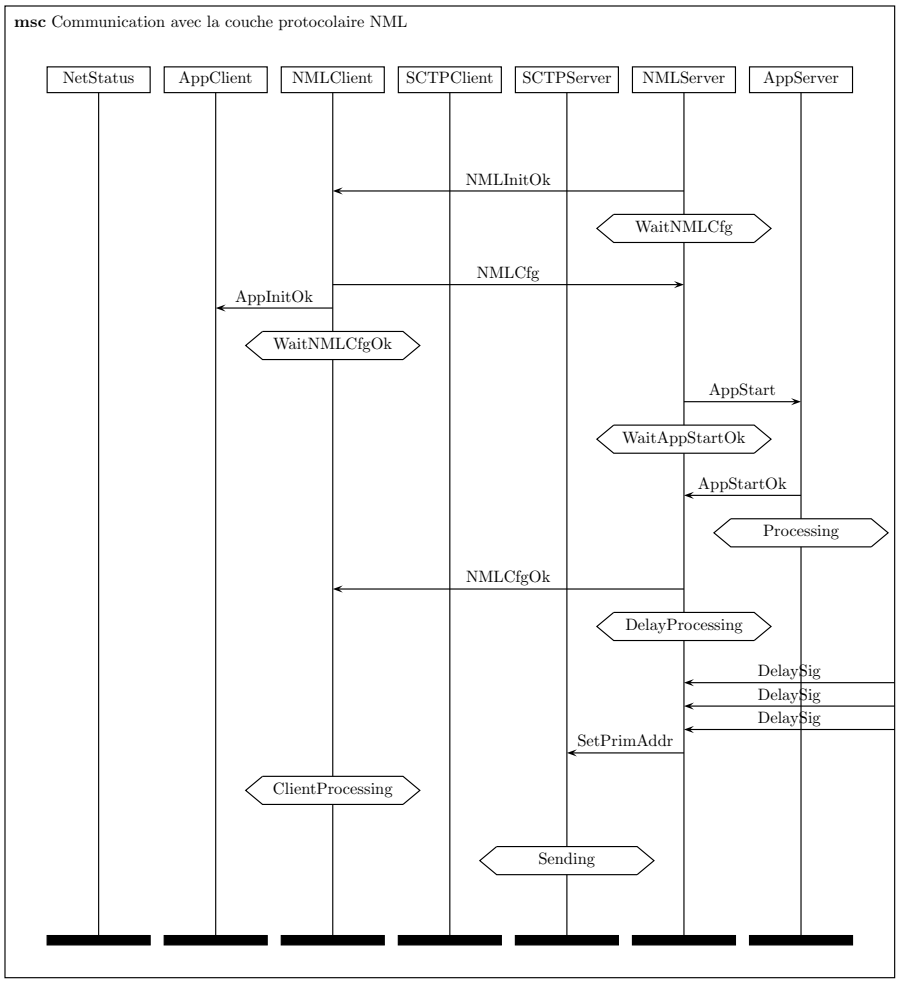

FIG. 4.6 – Message Sequence Chart 2

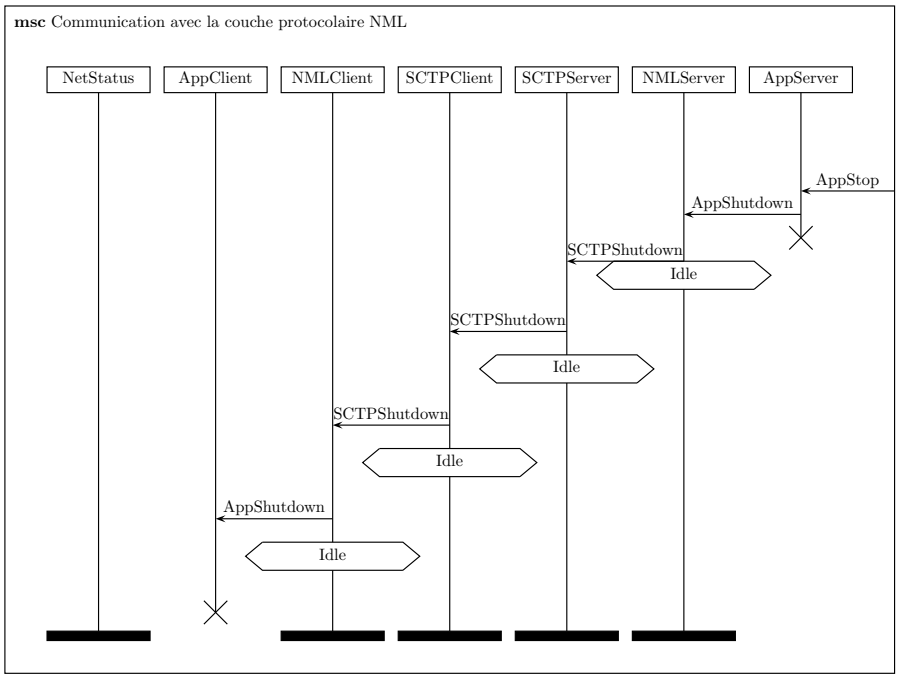

FIG. 4.7 – Message Sequence Chart 3

Les valeurs des paramètres de bande passante, de délai et de coût peuvent prendre des valeurs entières entre 0 et 10. La valeur de la contrainte BANDWIDTH peut varier de 0.0 à 1.0 et celle de la contrainte DELAY de 1 à 10. Les interfaces peuvent être de type unidirectionnel ou de type bidirectionnel. Nous nous sommes limités à trois interfaces réseaux pour le terminal.
Lors de la validation du modèle, le simulateur explore l'ensemble des états possibles. Etant donné le nombre de variables et le nombre de valeurs atteignables, nous sommes rapidement confrontés à un phénomène d'explosion combinatoire. Une validation exhaustive de l'ensemble des états est donc difficilement réalisable.
Un parcours aléatoire de l'arbre d'état avec une profondeur suffisament importante (10^9), permet d'atteindre au moins une fois chaque état, et de franchir au moins une fois chaque transition. Lors de cette validation avec exploration aléatoire, aucun état puits n'a été detecté.
L'usage des scénarii a permis de valider chaque fonctionnalité indépendamment. Exemple de scénarii :
Scénario 1 :

- Contrainte = BANWIDTH
- Valeur de la contrainte = 0.8
- Paramètres de bande passante des interfaces réseaux : NetA = 5, NetB = 1, NetC = 4
- Type d'interface réseau : NetA = Bidirect., NetB = Bidirect., NetC = Unidirect.

Résultat : La contrainte étant de type BANDWIDTH, le NML prend en compte les réseaux de type unidirectionnel. La valeur BWLIM=0.8*5=4, leNML sélectionne donc les réseaux possèdant un paramètre de bande passante supérieur ou égal à 4, soit NetA et NetB. Le NML opère donc une aggrégation avec les réseaux NetA et NetC.
Scénario 2 :

- Contrainte = DELAY,
- Valeur de la contrainte = 4,
- Paramètres de délai relatifs aux interfaces réseaux : NetA = 2, NetB = 3, NetC = 5,
- Paramètres de coût relatifs aux interfaces réseaux : NetA = 3, NetB = 2, NetC = 1,
- Type d'interface réseau : NetA = Bidirect., NetB = Bidirect., NetC = Bidirect.

Résultat : La contrainte étant de type DELAY, le NML prend en compte les réseaux de type bidirectionnel, soit NetA, NetB et NetC. La valeur de la contrainte étant 4, le NML sélectionne dans un premier temps les réseaux NetA et NetB. La valeur de coût du réseau NetB étant inférieure à celle du réseau NetA, et le réseau NetB satisfaisant la contrainte de l'application, le NML sélectionne donc le réseau NetB pour communiquer. Les fonctionnalités du modèle que nous avons vérifiées sont les suivantes :

- Sélection des réseaux en fontion de leur type et de la contrainte applicative. (tous les réseaux pour une contrainte BANDWIDTH et seulement les réseaux bidirectionnels pour les contraintes COST et DELAY)
- Agrégation cohérente des réseaux par rapport à la valeur de la contrainte BANDWIDTH,
- Sélection correcte d'un réseau par rapport aux paramètres coût et délai pour une contrainte DELAY,
- Sélection d'un réseau par rapport à une contrainte COST,
- Mise à jour de la configuration du NML lors d'une modification des interfaces réseaux sur le terminal,
- Redéfinition de la coopération de réseau suite à un changement de configuration du NML ou d'une modification des paramètres mesurés depuis le serveur.

Ce modèle SDL que nous proposons permet ainsi de décrire l'intégration du module de gestion des liens NML dans une pile protocolaire courante (couche réseau, transport et applicative) et explicite les interactions et les échanges nécessaires pour l'application d'une coopération de réseaux.

Principe de fonctionnement du NML

La représentation SDL précédente offre une description d'une communication (utilisant le NML) proche de la réalité dans le sens où nous pouvons séparer les entités serveur et client en deux blocks autonomes comprenant chacun des processus indépendants. Nous pouvons ainsi observer les communications entre les hôtes réseaux et entre les processus d'une même entité.
Cependant, afin de décrire le fonctionnement logique global du NML (comprenant la partie serveur et la partie cliente) nous utilisons les réseaux de Petri qui sont un autre outil de modélisation plus approprié à ce type de représentation (Fig. 4.8). Pour cette modélisation, nous avons utilisé le logiciel *CPNTools* qui permet de décrire et d'analyser des réseaux de Petri colorés et temporisés [62]. Les couleurs utilisées dans ce réseau de Petri sont les suivantes :

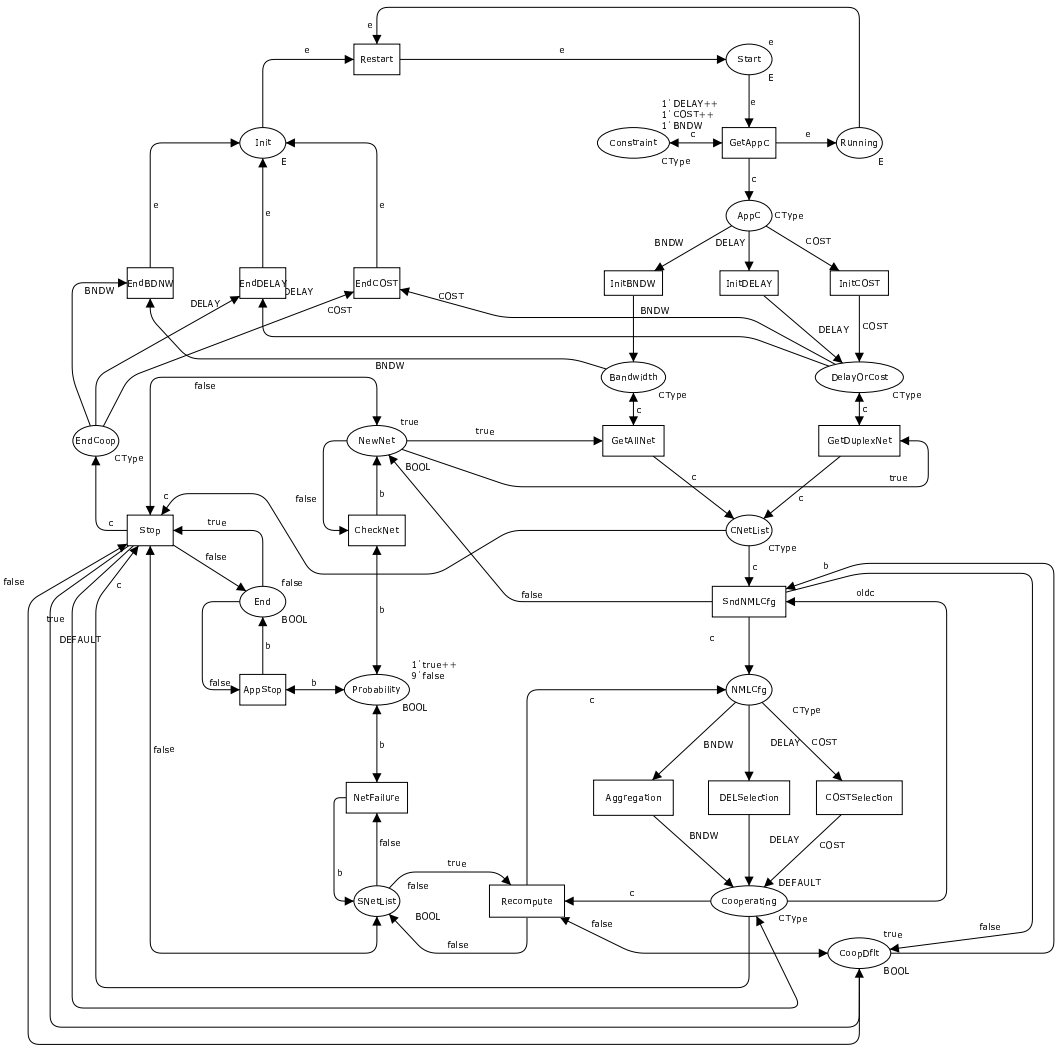

FIG. 4.8 – Principe de fonctionnement du NML

- `BOOL` : booléen
- `E` : sans couleur
- `CType` : cette couleur traduit le type de contrainte et peut prendre les valeurs `BNDW` (`BANDWIDTH`), `DELAY`, `COST` et `DEFAULT`. La valeur `DEFAULT` caractérise une absence de coopération.

Les variables :

- `c, oldc` : variable de type `CType`
- `e` : variable de type `E` (jeton sans couleur)
- `c` : variable de type `BOOL`

Au démarrage du système, place `Start`, le NML récupère la contrainte de l'application : transition `GetAppC`. Un jeton de couleur `CType` est placé sur la place `AppC`. Pour la contrainte BANDWIDTH le NML prend en compte l'ensemble des réseaux (`GetAllNet`) et pour les contraintes COST ou DELAY, le NML liste uniquement les interfaces unidirectionnelles (`GetDuplexNet`).
La liste des réseaux, ainsi que la contrainte peuvent alors être envoyées au NML distant : transition `SndNMLCfg`. Le NML sort de son état `DEFAULT` (la place `Cooperating` perd un jeton) et configuration est appliquée (transition `Aggregation`, `DELSelection`, ou `COSTSelection`. La place `Cooperating` reçoit alors un jeton de couleur `CType` avec la valeur de la contrainte.
La place `Probability`, de type `BOOL`, contient 9 jetons de valeur `false` et 1 jeton de valeur `true`. Cette place permet d'introduire une probabilité de 0.1 sur les transitions `AppStop`, `CheckNet` et `NetFailure`.
La transition `CheckNet` simule une reconfiguration des interfaces coté terminal, ce qui entraîne une resélection et l'envoi d'une nouvelle configuration NML.
La transition `NetFailure` modélise une défaillance sur un chemin IP vers une interface du terminal. Cette défaillance peut être détectée par le serveur à l'aide des messages HEARTBEAT du protocole SCTP. Une nouvelle configuration NML ne prenant pas en compte l'interface défaillante est alors appliquée.
La transition `AppStop` conditionne l'arrêt de la communication suite à la fin de l'application. Les jetons de couleur `CType` sont alors retirés des places caractérisant le mode de coopération, transition : `Stop, EndBNDW, EndDELAY`, et `EndCOST` et places : `Bandwidth, DelayOrCost` et `Cooperating`.
La transition `Restart` permet au système de revenir à son état initial.

Analyse des états Lors de l'analyse du modèle, l'ensemble des états sont parcourus (Tab. 4.4), Status : Full. La recherche des graphes Strongly Connected Components (SCC) montre qu'il n'existe qu'une seule composante fortement connexe. Le système peut donc toujours revenir dans son état initial. De plus il n'existe aucun marquage bloquant dans cette représentation. Le schéma de principe ainsi décrit doit permettre un fonctionnement correct du NML et offre donc une coopération de réseaux efficace lors d'une communication. Ces deux modèles forment donc une base pour la construction d'une pile protocolaire intégrant un processus de gestion des liens, tirant ainsi profit d'accès à de multiples réseaux, à travers une coopération de réseaux efficace.

```
 Statistics
----------------------------------------------------------------------
  State Space
    Nodes:  12
    Arcs:   35
    Secs:   0
    Status: Full

  Scc Graph
    Nodes:  1
    Arcs:   0
    Secs:   0


 Home Properties
----------------------------------------------------------------------
  Home Markings:  All


 Liveness Properties
----------------------------------------------------------------------
  Dead Markings:  None
```

TAB. 4.4 – Résultat de simulations en réseau de Petri

Dans ce chapitre nous avons décrit le processus de coopération de réseaux résidant dans la couche NML située entre le niveau applicatif et le niveau transport.
Le NML utilise d'une part des contraintes limitées et relatives à l'application , DELAY, BANDWIDTH et COST, et d'autre part des caratéristiques réseaux spécifiées ou mesurées par la couche transport afin de définir la configuration de coopération adéquate au service souhaité.
Les modélisation en réseaux de Petri et SDL ont respectivement permis de décrire de façon précise le principe de fonctionnement du NML et la signalisation associée.

Chapitre 5

Expérimentations de l'architecture protocolaire

5.1 Implémentation C++

Nous avons développé des implémentations de certaines fonctionnalités de coopération de réseaux afin d'évaluer les propositions précédentes à travers des expérimentations. La couche Network Management Layer (NML) n'a pas été développée dans sa totalité, notamment la partie signalisation entre les entités d'extrémité est absente des prototypes développés.
Les deux fonctionalités que nous avons intégrées sont :

- la sélection d'une interface pour une latence minimale,
- la combinaison de liens de communication pour une diffusion vidéo.

Nous avons réalisé ces développements en C/C++ dans un environnement GNU/Linux intégrant le protocole Stream Control Transmission Protocol (SCTP) et les fonctionnalités de mobilité IPv6.

Environnement de développement :

- Debian GNU/Linux 3.1 (stable) [63]
- LKSCTP version 1.0.2 pour kernel 2.6.10 [64]
- Linux kernel vanilla version 2.6.11 [65]
- MIPL version 2.0-rc3 pour kernel 2.6.11 [66]

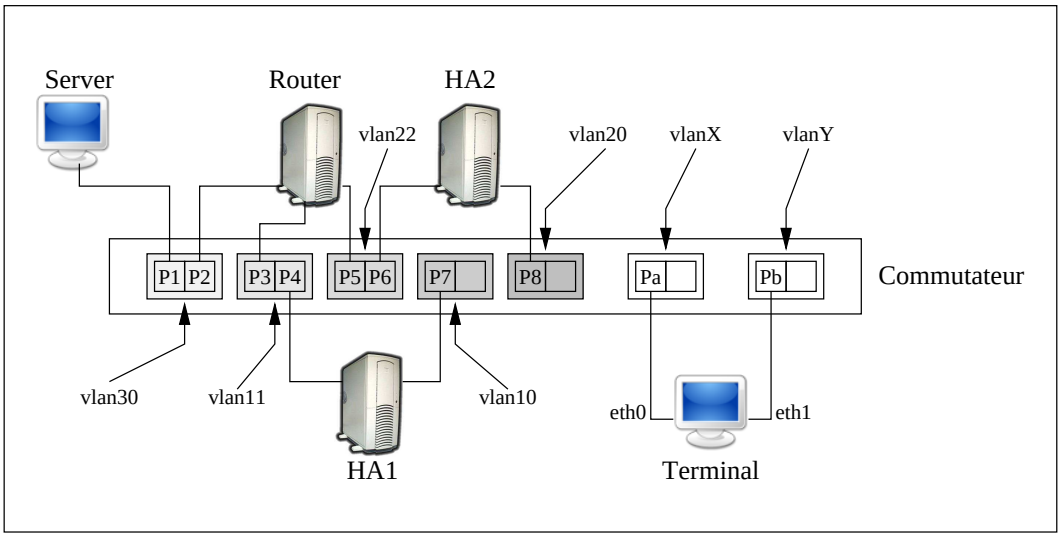

FIG. 5.1 – Maquette : structure physique

5.2 Sélection du réseau de plus faible latence

5.2.1 Présentation de la maquette

Pour la réalisation de cette maquette nous avons utilisé des ordinateurs de type PC pour toutes les entités réseaux : serveur, client et routeurs.
Ces ordinateurs sont connectés à un commutateur (CISCO catalyst 2950) intégrant la gestion de Virtual Local Area Network (VLAN)(Fig. 5.1). Tous les équipements sont reliés entre eux avec des interfaces ethernet RJ45, 10/100Mbits.
Les PCs `Terminal` et `Server` représentent respectivement le client et le serveur. Les PCs `Router`, `HA1` et `HA2` sont utilisés en guise de routeurs. Le PC `Router` fait office de routeur central, il est connecté au `VLAN30`, `VLAN11` et `VLAN22` par l'intermédiaire des ports `P2`, `P3` et `P5`. Les routeurs `HA1` et `HA2` interconnectent respectivement les `VLAN11` et `VLAN10` et les `VLAN22` et `VLAN20`. `HA1` et `HA2` font également office d'agent mère (Home Agent (HA)) pour les réseaux `VLAN10` et `VLAN20`. Le `Terminal` possède deux interfaces `eth0` et `eth1` connectées respectivement aux ports `Pa` et `Pb`. Chacun des ports `Pa` et `Pb` peut appartenir au `VLAN10`, `VLAN20` ou `VLAN30`. Le changement de VLAN pour les ports `Pa` et `Pb` permet de réaliser un handover entre les différents réseaux. Nous avons désactivé les modules de gestion liés aux ports du commutateur (Spanning Tree Protocol (STP), auto-configuration...) afin de réduire au minimum les temps de handover au niveau ethernet, qui sont de l'ordre de quelques micro-secondes.

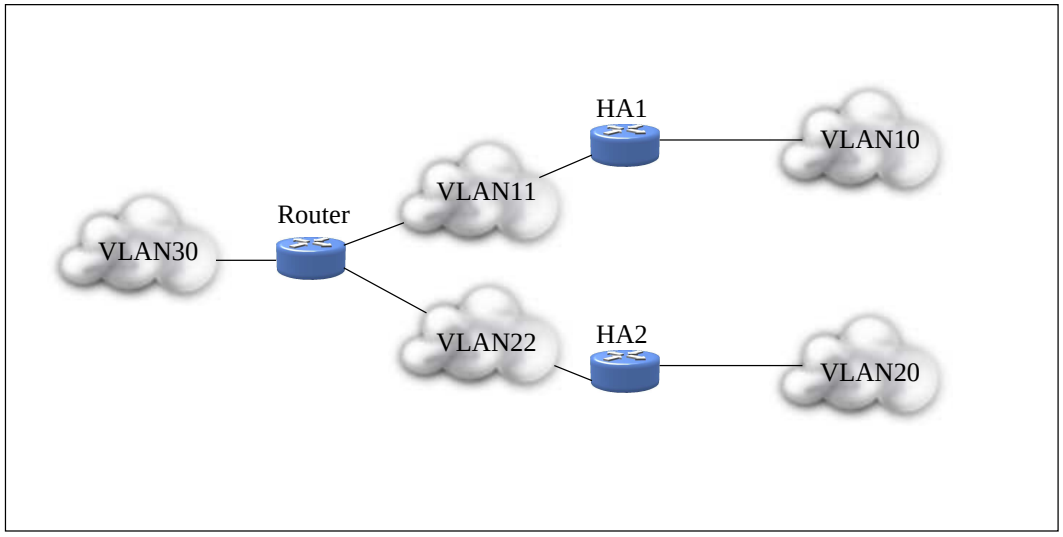

FIG. 5.2 – Maquette : structure logique

La structure logique de la maquette est représentée sur la figure 5.2. Le `Server` est toujours connecté sur le `VLAN30` et chacune des interfaces du `Terminal` peut potentiellement se connecter au `VLAN10`, `VLAN20` et `VLAN30`.
Nous avons assigné un sous-réseau IPv6 différent à chaque VLAN. Le `VLAN30` correspond au réseau IPv6 3ffe :30 : :/64, le `VLAN10` au réseau 3ffe :10 : :/64....

5.2.2 Expérimentation

Description : Sur la figure 5.3 nous voyons que le terminal mobile `MT` possède deux interfaces `eth0` et `eth1`. Chacune de ces interfaces peut potentiellement se connecter au réseau `Net1`, `Net2` et `Net3`. Le lien entre `Router` et `HA1` possède une latence de 200ms et le lien entre `Router` et `HA2` une latence de 400ms. Nous utilisons l'émulateur réseau *netem* [67] sur `Router` afin d'introduire ces délais vers les routeurs `HA1` et `HA2`. L'application de test du `Server` transfère des données à destination du `Terminal` à un débit de 500Kbit/s. La taille des paquets est de 1076 octets (IPv6 + SCTP + payload = 40 + 32 + 1024 = 1096).
Le flux SCTP peut donc être envoyé sur les réseaux `Net1`, `Net2` ou `Net3` avec une latence de communication respectivement égale à 200ms, 400ms, ou 0ms (la latence du réseau `Net3` est de l'ordre de quelques millisecondes, ce qui est négligeable par rapport aux délais vers `Net1` et `Net2`).

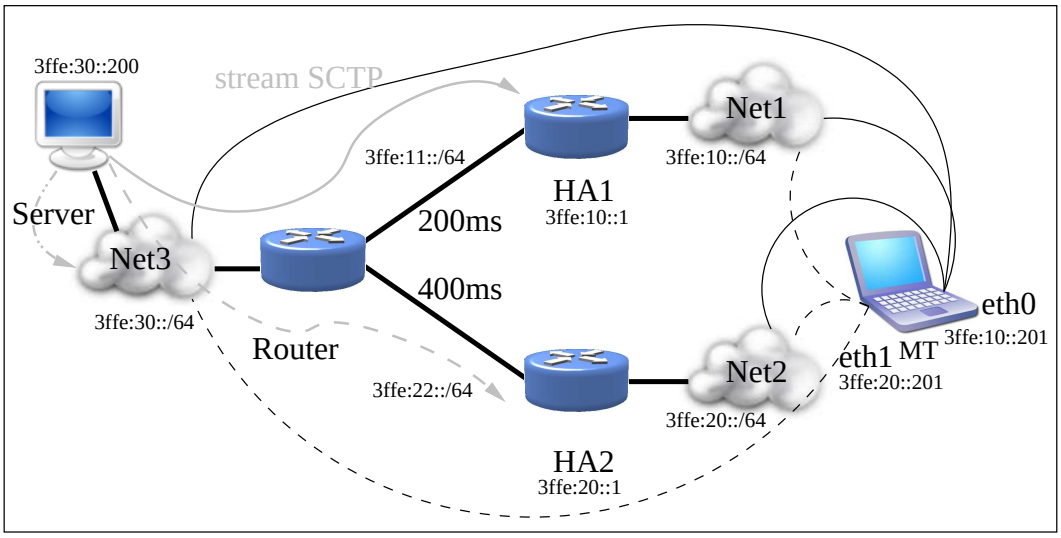

FIG. 5.3 – Maquette : sélection du meilleur réseau

Un script de commande permet de contrôler le commutateur et de modifier régulièrement l'attribution des ports `Pa` et `Pb` aux VLANs `VLAN10`, `VLAN20` ou `VLAN30`. La commande appliquée sur le port `Pa` est représentée à travers le troisième graphique de la figure 5.5.

Ce graphique représente le délai d'une communication entre le `Server` et le `Terminal` passant par l'interface `eth0`. De 0s à 120s, le délai est de 200ms, ce qui signifie que l'interface `eth0` est connectée au `VLAN10` et appartient donc au réseau `Net1`. A t=120s, le script de contrôle bascule le port `Pa` sur le `VLAN20`, le délai est alors de 400ms, et l'interface `eth0` appartient au réseau `Net2`. De 240s à 360s, l'interface `eth0` est connectée au réseau `Net3` avant de se rattacher de nouveau au réseau `Net1` à t=360s.

Toutes les 120s, le script de commande du commutateur exécute donc une permutation circulaire entre les réseaux `Net1`, `Net2` et `Net3`.

Il en est de même pour l'interface `eth1`, où lorsque le délai a une valeur de 0ms, 200ms et 400ms, l'interface `eth1` appartient respectivement aux réseaux `Net3`, `Net1` et `Net2`. Pour l'interface `eth1`, le script de contrôle effectue une permutation toutes les 240s. Les `handovers` sont effectués avec un décalage d'au moins 60s. Ainsi, l'ensemble des combinaisons possibles sont évaluées durant l'expérimentation. Le tableau 5.1 rassemble toutes les configurations réseaux rencontrées par le terminal, ainsi que les résultats théoriques attendus concernant la sélection des interfaces par rapport à la latence minimale.

Dans cette expérience, le terminal est constamment en situation de mobilité, simu-

Date	*eth0*		*eth1*		*Résultats attendus*	
0s	Net1	200ms	Net2	400ms	eth0	200ms
60s	Net1	200ms	Net1	200ms	eth0 ou eth1	200ms
120s	Net2	400ms	Net1	200ms	eth1	200ms
240s	Net3	0ms	Net1	200ms	eth0	0ms
300s	Net3	0ms	Net2	400ms	eth0	0ms
480s	Net2	400ms	Net2	400ms	eth0 ou eth1	400ms
540s	Net2	400ms	Net3	0ms	eth1	0ms
600s	Net3	0ms	Net3	0ms	eth0 ou eth1	0ms
720s	Net1	200ms	Net3	0ms	eth0 ou eth1	0ms
780s	Net1	200ms	Net1	200ms	eth0 ou eth1	200ms
840s	Net2	400ms	Net1	200ms	eth1	200ms
960s	Net3	0ms	Net1	200ms	eth0	0ms
1020s	Net3	0ms	Net2	400ms	eth0	0ms
1080s	Net1	200ms	Net2	400ms	eth0	200ms
1200s	Net2	400ms	Net2	400ms	eth0 ou eth1	400ms
1260s	Net2	400ms	Net3	0ms	eth1	0ms
1320s	Net3	0ms	Net3	0ms	eth0 ou eth1	0ms

Tab. 5.1 – Tests et prévisions

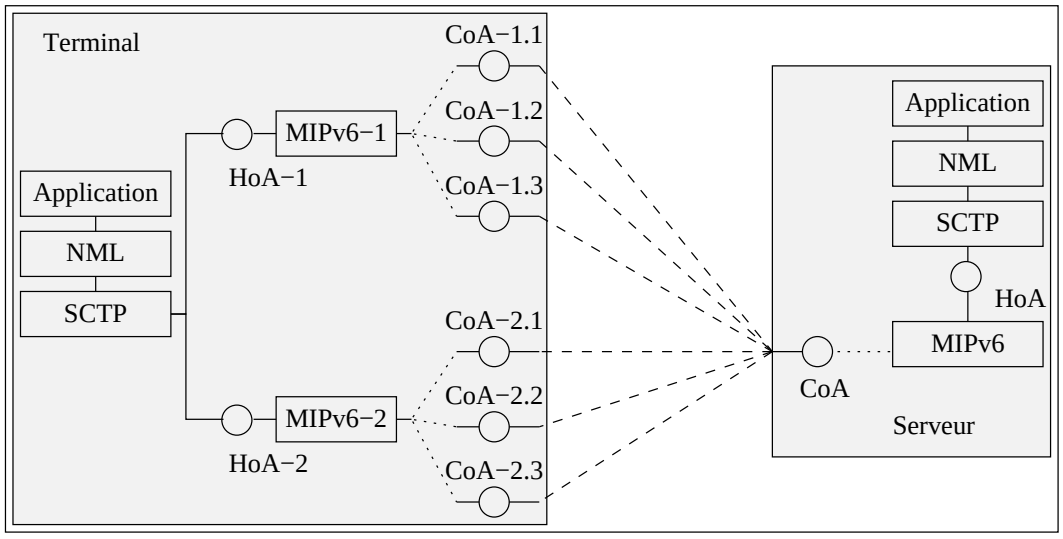

Fig. 5.4 – Pile protocolaire NML, SCTP et IPv6

lée par des handovers réguliers sur ses deux interfaces. Les réseaux auquels peuvent se connecter les interfaces ont des caractéristiques de délai fortement hétérogènes (respectivement 200ms, 400ms, et 0ms pour `Net1`, `Net2` et `Net3`).

Le terminal et le serveur possèdent tous les deux une pile protocolaire identique constituée du protocole Mobile IPv6 (MIPv6) et SCTP. Le serveur est dans une situation statique, son interface réseaux restant toujours connectée sur le réseau `Net3`. La fonctionnalité de coopération de réseaux *sélection* est implémentée au sein même du programme serveur. La figure 5.4 schématise les connexions logiques possibles entre le serveur et le terminal.
Dans cette expérimentation, le serveur et le terminal établissent l'association SCTP en utilisant leurs adresses mères {(HoA)↔(HoA-1 ;HoA-2)}. En situation de mobilité, le processus MIPv6 assure alors la connectivité et optimise le routage des paquets en utilisant l'adresse temporaire adéquate (CoA). Chaque adresse mère étant assignée à une interface, (HoA-1 pour `eth0` et HoA-2 pour `eth1`), les opérations de `handovers` verticaux et horizontaux sont alors respectivement prises en charge par le processus SCTP et par le processus MIPv6.
Une autre approche consisterait à n'utiliser qu'une seule adresse mère, et à établir l'association SCTP avec les adresses temporaires. Cependant cette solution possède deux inconvénients majeurs :

1. la détection des apparitions ou des suppressions des adresses temporaires IPv6 devraient se faire au niveau du NML afin que celui-ci mette à jour l'association SCTP. Ceci occasionnerait des traitements supplémentaires qui sont déjà effectués au niveau de la couche IPv6,
2. la suppression au niveau de la couche réseau, d'une adresse temporaire IPv6, également adresse primaire de l'association SCTP, avant que celle-ci ne soit retirée de l'association, génère un blocage du système. Cette solution serait donc limitée à des opérations de soft-handover.

La figure 5.5 montre le trafic sur les interfaces `eth0` et `eth1` en fonction de la latence appliquée sur chaque interface. Nous constatons que la sélection effective est bien conforme au modèle théorique, et que, par conséquent, les données sont toujours transmises vers l'interface offrant la latence la plus faible.
Nous pouvons remarquer des baisses de débit régulières (t=360s). Ces chutes de trafic sont occasionnées par les `handovers` IP horizontaux des interfaces.
Par exemple, juste avant t=360s, le trafic est envoyé sur l'interface `eth0` avec une latence de 0ms. A t=360s, cette interface passe du réseau `Net3` au réseau `Net1` avec une latence de 200ms. La latence de l'interface `eth1` étant toujours de 400ms, le flux devrait toujours être transmis sur `eth0`. Cependant la détection du nouveau réseau IP, l'autoconfiguration et l'augmentation de latence de 0ms à 200ms conduisent le protocole SCTP à retransmettre quelques paquets sur l'interface `eth1`. Une fois la

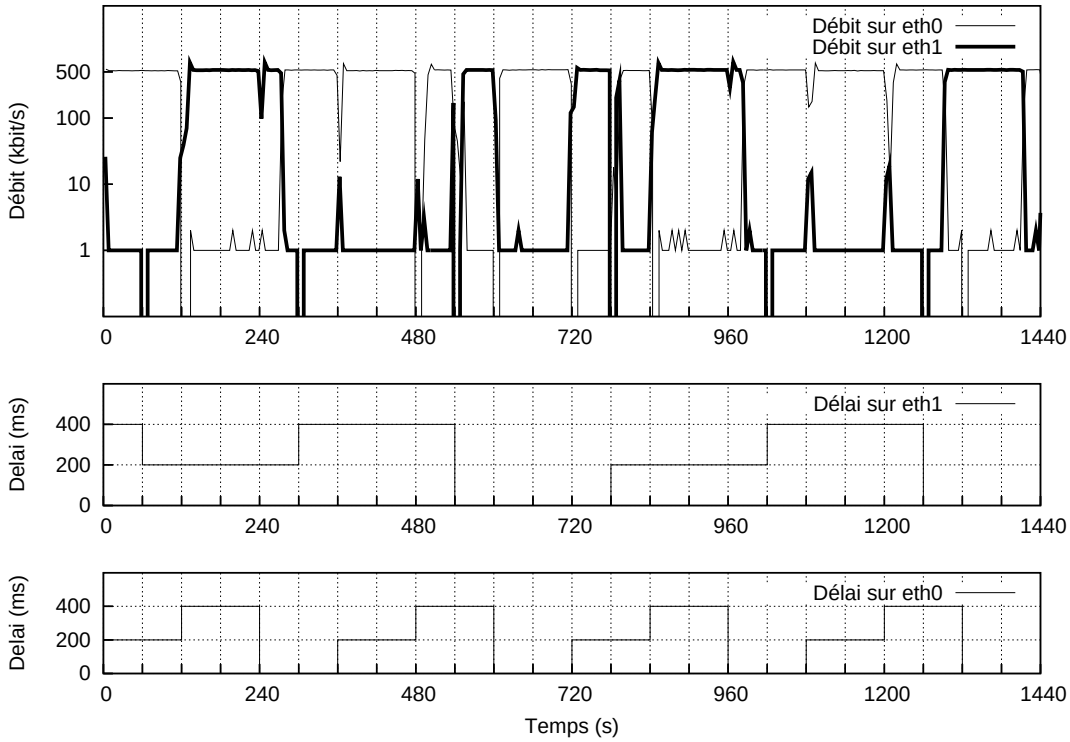

FIG. 5.5 – Sélection du réseau de plus faible latence

connectivité IP rétablie sur l'interface `eth0` le flux est réorienté vers cette interface, puisque celle-ci possède alors toujours la latence la plus faible. Les handovers horizontaux (changement de réseaux IP) entraînent une interruption dans la transmission de l'ordre de 2s à 3s. Ces durées pourraient être réduites en utilisant des mécanismes de Fast handovers de MIPv6 [68].
En revanche, les handovers verticaux (passage d'une interface à une autre) s'opèrent sans perte de paquets et sans interruption de trafic. A t=240s et t=960s nous constatons un retard dans le basculement du trafic vers l'interface de plus faible latence. Ce retard est dû à l'évaluation du SRTT, qui repose sur une décroissance exponentielle, formule couramment utilisée afin de lisser les mesures.

$$SRTT_n = \alpha * RTT_n + (1 - \alpha) * SRTT_{n-1} \ , \ avec \ 0 < \alpha < 1 \qquad (5.1)$$

La mesure de latence étant fondée sur les messages heartbeat, envoyés toutes les 3 secondes, il faut un certain temps afin que le SRTT estimé converge vers la mesure

exacte. Un autre mode de calcul, ou une adaptation dynamique de la fréquence d'envoi des messages heartbeat pourrait améliorer le temps de détection, cependant ce point ne sera pas abordé dans cette étude.

5.3 Diffusion vidéo à travers une architecture hybride

Dans cette expérimentation, nous utilisons une fonctionnalité de coopération de réseaux qui est la combinaison afin de fournir un service de diffusion vidéo vers un terminal. Cette architecture comprend un lien Digital Video Broadcasting Handheld (DVB-H) utilisé pour acheminé les paquets vidéo vers le terminal et un lien Universal Mobile Telecommunications System (UMTS) qui fait office de voie de retour. Nous évaluons les performances de ce service à travers des simulations réalisées avec Network Simulator 2 (NS2).
Le terminal est équipé de deux interfaces de communication possédant chacune des caractéristiques très différentes : une interface UMTS (60kbit/s, 240ms, bidirectionnelle) et une interface DVB-H (diffusion en time slicing, unidirectionnelle).
Le service de diffusion vidéo repose sur le protocole SCTP ainsi que sur la combinaison des réseaux de type UMTS et DVB-H. A travers plusieurs simulations nous allons comparer cette architecture protocolaire et réseaux à d'autres solutions de diffusion (User Datagram Protocol (UDP)et Transmission Control Protocol (TCP)).
Dans un premier temps, nous avons développé, pour des besoins de simulation, plusieurs fonctionnalités qui n'étaient pas présentes dans la version standard du simulateur (diffusion DVB-H et Partially Reliable SCTP (PR-SCTP)).
Par la suite nous avons étudié les mécanismes de contrôle de congestion des protocoles SCTP et TCP afin de pouvoir analyser les résultats de simulations dans l'architecture complète.
Enfin nous avons évalué les performances de ces diverses solutions de diffusion vidéo dans un modèle simulant une architecture de coopération de réseaux UMTS/DVB-H.

5.3.1 Simulation d'un lien DVB-H

La bande passante du lien DVB-H est de 2560Kbit/s, avec une période de burst de 100ms et une période de repos (idle) de 400ms. La capacité du canal sur le lien DVB-H est donc de 512Kbit/s. Nous avons développé un nouveau type de lien dans le simulateur NS2 afin d'intégrer le type de diffusion DVB-H. Des simulations ont permis de valider le bon fonctionnement de l'implémentation DVB-H (Fig. 5.6, 5.7).

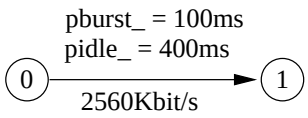

FIG. 5.6 – Validation DVB-H

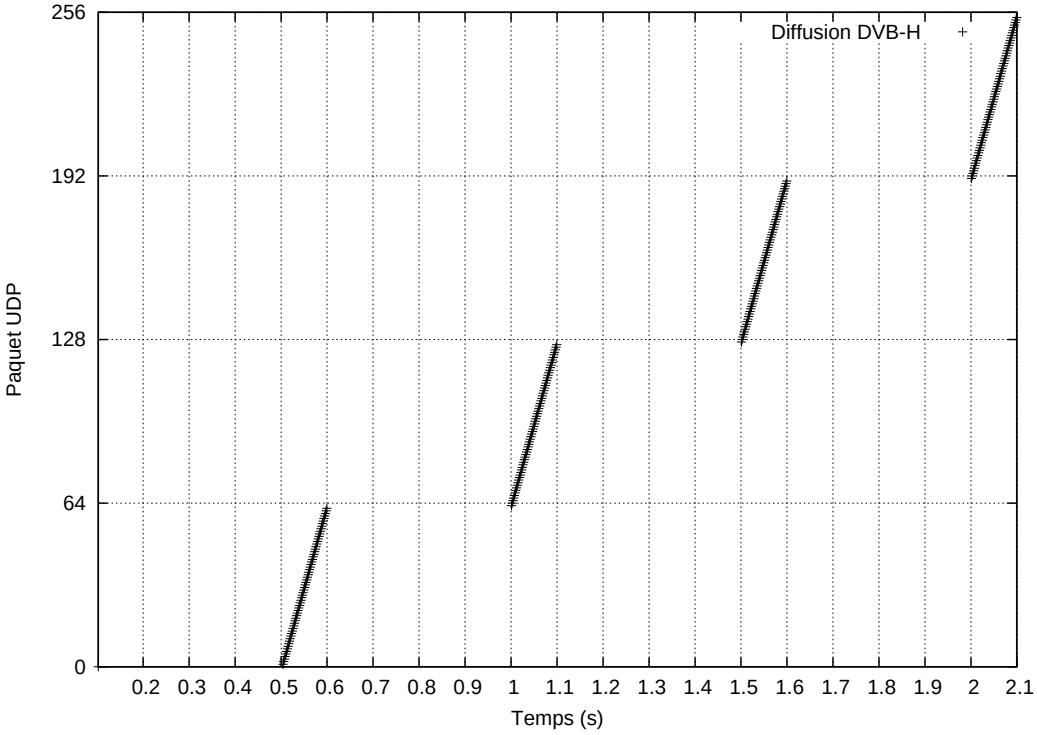

FIG. 5.7 – Diffusion DVB-H

Dans le modèle (Fig. 5.6), le nœud 0 diffuse un trafic constant à un débit de 512Kbit/s avec une taille de paquet de 500 octets, ce qui correspond à une émission de 128 paquets par seconde.
La figure 5.7 montre la réception des paquets au niveau du nœud 1 et nous constatons que le terminal (nœud 1) reçoit 64 paquets durant une période de 100ms. Le terminal peut ensuite se placer dans un état oisif (idle) pendant une période de 400ms, période pendant laquelle il ne reçoit aucun paquet. Ainsi sur une période de 1s, le terminal traverse deux périodes de burst de 100ms chacune et reçoit au total 128 paquets, soit 512Kbit/s.

5.3.2 Le protocole PR-SCTP

PR-SCTP est une extension du protocole SCTP permettant de spécifier la durée de vie d'un message SCTP [69]. Ce protocole comporte deux principales nouveautés :

- un nouveau paramètre dans les échanges INIT/INIT-ACK, indiquant le support de l'extension PR-SCTP,
- un nouveau type de message, FORWARD-TSN indiquant au récepteur de mettre à jour la valeur du numéro d'acquittement cumulatif. Ce message permet ainsi de ne pas tenir compte de la perte d'un ou de plusieurs paquets.

Le RFC3758 précise que la notion de fiabilité partielle (ie. partial reliability) peut se présenter sous différentes formes mais encourage l'utilisation d'une limite temporelle pour les retransmissions des paquets. Cette limite temporelle est également utilisée dans le draft décrivant l'API SCTP [58], ainsi que dans certaines implémentations [64].

L'implémentation PR-SCTP du simulateur NS2 propose une fiabilité partielle reposant sur le nombre maximal de retransmissions des paquets qui traduit par le paramètre `reliability_`. Nous avons intégré un nouveau paramètre `prttl_` représentant la durée de vie maximale des paquets SCTP exprimée en millisecondes. Une fois cette durée de vie écoulée, et si le paquet n'est toujours pas acquitté, l'émetteur envoie un message FORWARD-TSN au récepteur afin de passer outre le paquet manquant.
En combinant l'ancien paramètre `reliability_` et le paramètre `prttl_`, nous pouvons ainsi préciser le nombre maximal de retransmission d'un paquet dans une certaine limite de temps. Nous utilisons principalement le paramètre `prtll_` en fixant le nombre de retransmission `reliability_` à une valeur très élevée. Le comportement du protocole PR-SCTP est ainsi très proche des implémentations logicielles.

5.3.3 Le contrôle de congestion des protocoles SCTP et TCP

Pour cette étude nous simulons un transfert FTP entre deux nœuds avec des liens ayant une bande passante de 10Mbit/s et une latence de 100ms. Les mécanismes de contrôle de flux des protocoles SCTP et TCP sont très semblables et reposent sur la RFC2581 [24]. La figure 5.8 montre l'évolution de la fenêtre de congestion pour les protocoles SCTP et TCP tel qu'elle devrait être selon la RFC 2581.
Durant les quatre premières secondes, la fenêtre de congestion est en phase Slow Start, identifiable par son accroissement exponentiel, le Slow Start Threshold étant fixé à quarante cinq paquets. Le protocole TCP est légèrement plus rapide, puisque celui-ci ne nécessite que trois échanges de paquets pour l'initialisation, au lieu de

quatre pour SCTP.
Lorsque le contrôle de flux est en phase de Congestion Avoidance (accroissement linéaire), les paquets 2000 (t=16s) et 10000 (t=54s) sont perdus. Les fenêtres sont alors réduites de moitié et augmentent de nouveau d'un paquet tous les RTT. Cependant le protocole SCTP comporte quelques changements mineurs par rapport à TCP qui peuvent accroître ses performances dans certaines circonstances.

1. le protocole SCTP augmente sa Congestion WiNDow (CWND) de la quantité de donnée acquittée alors que TCP augmente sa CWND d'un segment pour chaque accusé reçu.
2. SCTP utilise les mécanismes de Selective ACKnowledgement (SACK) et ne possède pas de limite pour le nombre d'accusés notifiés dans le SACK. TCP est limité à 3 SACKs dans le meilleur des cas.
3. En phase de Congestion Avoidance, SCTP met à jour sa CWND tous les Round Trip Time (RTT) alors que TCP dépend d'une approximation basée sur les accusés de réception.

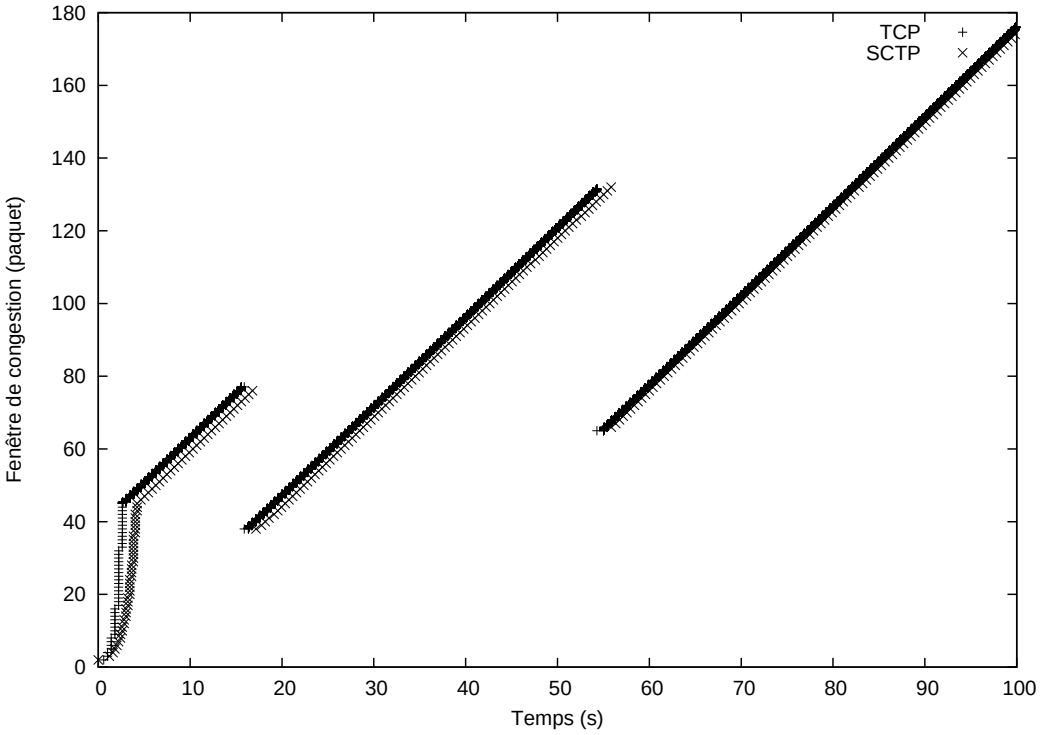

FIG. 5.8 – Contrôle de flux, SCTP et TCP

Le point 3 résulte d'une formule (Eq. 5.2) couramment utilisée dans les implémentations de TCP qui fournit une bonne approximation pour augmenter la fenêtre de congestion d'un segment tous les Round Trip Time (RTT) si le récepteur TCP acquitte tous les segments TCP.

$$CWND_n = CWND_{n-1} + SMSS * \frac{SMSS}{CWND_{n-1}} \tag{5.2}$$

Si le récepteur utilise l'option TCP Delayed Ack (DelAck), celui-ci ne renvoie qu'un accusé de réception sur deux dans la limite d'un certain interval de temps (100ms). Par conséquent l'accroissement sera inférieur à 1 segment par RTT en phase de congestion avoidance.
Le protocole SCTP utilise une nouvelle variable d'état `partial_bytes_acked` pour gérer l'évolution de la Congestion WiNDow (CWND).
Algorithme de gestion de la fenêtre de congestion en phase de congestion avoidance :

$partial\_bytes\_acked \leftarrow 0$
while CWND $>$ SSTHRESH **do**
  if SACK augmente le Cumulative TSN Ack Point **then**
    $partial\_bytes\_acked += Nombre\ d'\ octet\ acquittés\ (CumAck + GAB)$
  end if
  if partial_bytes_acked $\geq$ CWND **then**
    $CWND = CWND + MTU$
    $partial\_bytes\_acked -= CWND$
  end if
end while

Le protocole SCTP ne dépend donc pas du nombre d'accusés de réception émis pas le récepteur. Le protocole SCTP utilise par défaut la méthode DelAck avec un interval de temps de 200ms.
La figure 5.9 montre bien la limitation de l'algorithme de gestion de la CWND de TCP lorque celui-ci utilise l'option DelAck. Cette option étant couramment activée dans de nombreuses implémentations logicielles, nous ferons, par la suite, toujours référence à la version DelAck de TCP, le contrôle de flux de la version *théorique* étant très proche de celle du protocole SCTP.

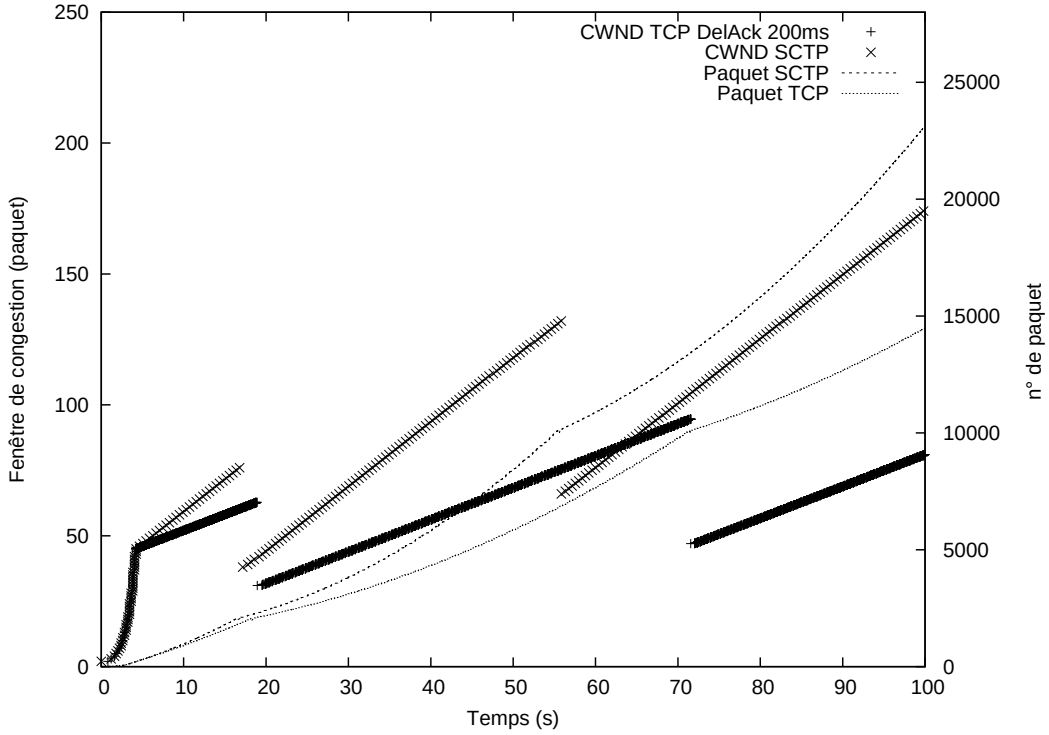

FIG. 5.9 – Contrôle de flux, SCTP et TCP DelAck (transfert FTP)

5.3.4 Modèle de simulation

La figure 5.10 représente le modèle de simulation utilisé. Les nœuds `D` et `U` correspondent respectivement aux interfaces DVB-H et UMTS du `Terminal`. Le nœud `E` permet d'introduire un taux d'erreur sur le lien DVB-H et le nœud `H` opère le time slicing. La connexion UMTS est simulée par le lien de `U` à `S` (60Kbits/, 240ms (RTT), bidirectionnnelle) et les liens de `S` et `D` (`S`→`E`→`H`→`D`) caractérisent une diffusion DVB-H.

L'application serveur simule la diffusion d'un flux vidéo à un débit constant de 384Kbit/s avec une taille de paquets de 1000 octets. L'application cliente considérée est un lecteur de flux vidéo possédant une mémoire tampon de 1s en plus du buffer nécessaire à la diffusion DVB-H. En effet, la diffusion DVB-H s'opérant en time slicing, l'application cliente doit mémoriser l'ensemble des paquets reçus pendant la période de burst. Dans ces simulations, le lien DVB-H transmet des bursts de 100ms et passe dans un état oisif pendant 400ms. L'application doit donc posséder un buffer d'au moins 500ms pour la réception DVB-H auquel nous rajoutons un buffer de 1s.

En supposant que la vidéo soit jouée par le client à un débit constant, nous pouvons calculer la date théorique dt_i d'arrivée du prochain paquet. Le débit étant de 384Kbit/s avec une taille de paquets de 1000 octets, le temps interpaquet (Inter Packet Gap (IPG)) est donc de 21ms.

$$IPG = \frac{1}{\frac{384*1000}{1000*8}} = 0.0208 \tag{5.3}$$

$$dt_i = dt_0 + IPG * pkt_i \tag{5.4}$$

Nous calculons ensuite le retard entre l'arrivée effective du paquet et sa date d'arrivée théorique. Si cet écart est supérieur à 1s, nous considérons ce paquet comme perdu. C'est ce que nous appellons le *taux d'erreur vidéo*.
Nous utilisons un `prttl_` que nous avons arbitrairement fixé à :

$$\texttt{prttl\_} = \texttt{pidle\_} + \frac{RTT_{UMTS}}{2} \tag{5.5}$$

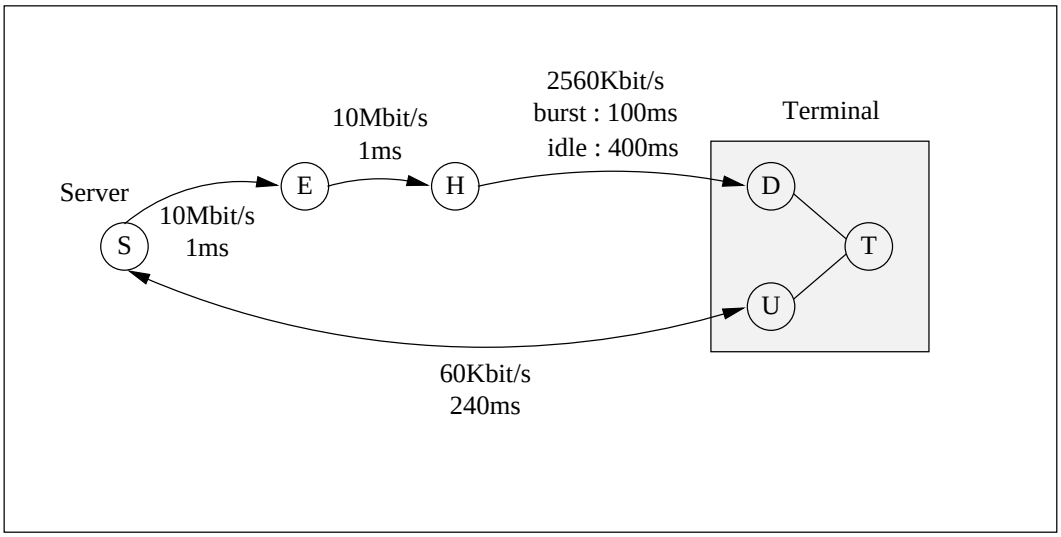

FIG. 5.10 – Modèle de simulation

5.3.5 Simulations

Nous avons réalisé des simulations avec les trois protocoles TCP, UDP et SCTP. Nous prenons le protocole UDP comme référence, étant donné que celui-ci est le plus

couramment employé pour des services de video streaming. La durée de diffusion du flux vidéo est de 1000s.

Le protocole TCP

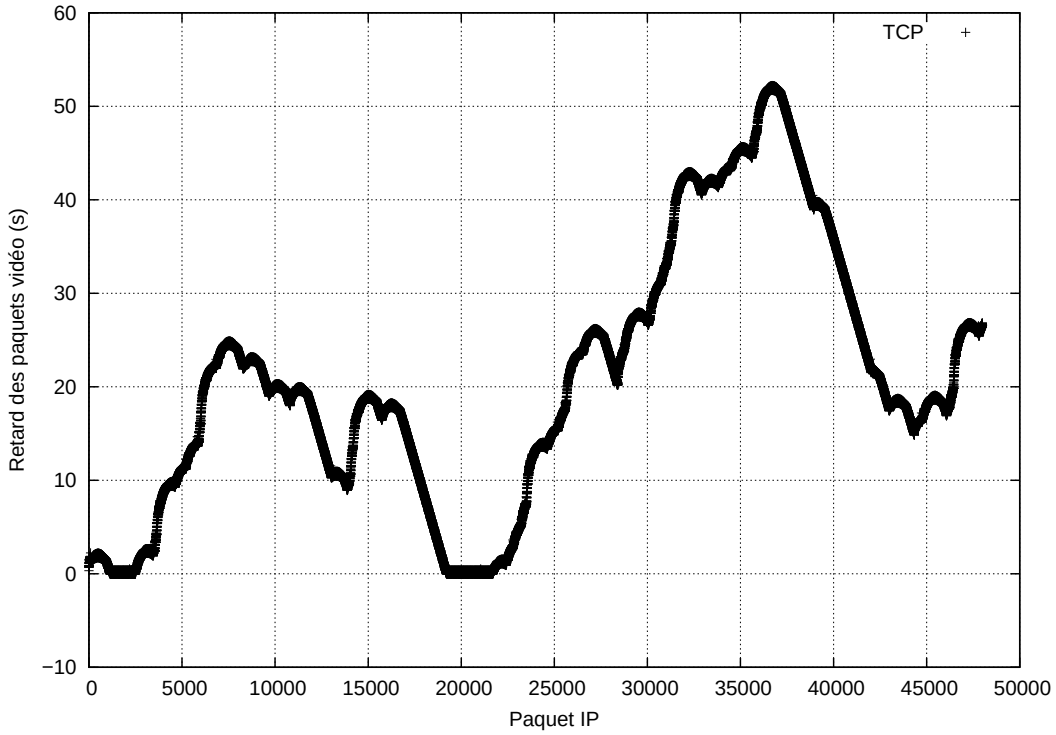

FIG. 5.11 – Dérive temporelle des paquets avec 0.1% d'erreur

La version du protocole TCP comprend les options Selective ACKnowledgement (SACK) et DelAck. Le figure 5.11 montre le retard des paquets vidéo par rapport à leur date d'arrivée théorique pour un taux d'erreur sur les paquets IP de 0.1%. Nous constatons que le client vidéo reçoit la grande majorité des paquets avec un retard supérieur à 1s, ce qui induit un taux d'erreur sur les paquets vidéo de 90%. Les facteurs responsables de ces contre performances sont les suivants :

1. Obligation de retransmission des paquets
2. Accroissement insuffisant de la fenêtre de congestion dû à l'option DelAck.
3. Retransmission des paquets perdus sur le lien DVB-H, ce qui peut entraîner des timeouts si le lien DVB-H entre dans un état idle juste avant la retransmission

Le protocole n'est bien évidemment pas adapté à la diffusion de flux vidéo, ces résultats fournissent cependant un point de comparaison avec le protocole SCTP.

Le protocole UDP

Le protocole UDP n'offre aucun mécanisme de retransmission ou de contrôle de congestion. Les pertes de paquets vidéo sont donc équivalentes aux pertes de paquets IP. UDP est ici considéré comme le protocole de référence auquel seront comparées les différentes configurations de SCTP.

Le protocole SCTP

Le protocole SCTP posséde deux caractéristiques fondamentales qui le différencie des protocoles TCP et UDP dans un contexte de diffusion vidéo.
D'une part, le mode partiellement relié autorise des retransmissions dans une certaine limite, ce qui place SCTP à mi-chemin entre TCP et UDP en terme de fiabilité. Nous utilisons le protocole SCTP en mode de fiabilité partielle, que nous notons PR-SCTP, avec une durée de vie de 520ms. Nous utilisons également le mode de fiabilité totale, que nous notons SCTP, et qui est similaire à TCP.
D'autre part SCTP offre deux possibilités de retransmission pour les paquets erronnés. Par défaut, celui-ci retransmet les paquets perdus sur un chemin IP autre que celui de l'adresse primaire. Pour les simulations, nous employons deux configurations du protocole SCTP : `RtxToSame` où les retransmissions sont faites vers l'adresse primaire, et `RtxToAlt` où les paquets sont renvoyés sur un autre chemin IP (en l'occurence UMTS dans notre modèle).

La figure 5.12 représente les retards des paquets vidéo reçus par le client pour une diffusion PR-SCTP `RtxToAlt` avec un taux d'erreur de 0.1% sur les paquets IP. Nous constatons que la dérive temporelle des paquets vidéo est limitée par rapport à celle occasionnée par TCP. La majorité des paquets a un retard négatif compris entre 0 et 400 ms dû à la diffusion DVB-H. Les paquets ayant un retard positif ont été retransmis pas UMTS.
Pour cette simulation, le taux d'erreur vidéo est de 0.06%, soit presque deux fois moins important que le taux d'erreur appliqué sur les paquets IP.

Nous avons simulé plusieurs diffusions vidéo avec un taux d'erreur compris entre 0.1% et 5% en utilisant les différentes configurations possibles pour SCTP. La figure 5.13 montre le taux d'erreur vidéo perçu au niveau du terminal en fonction du taux d'erreur des paquets IP.

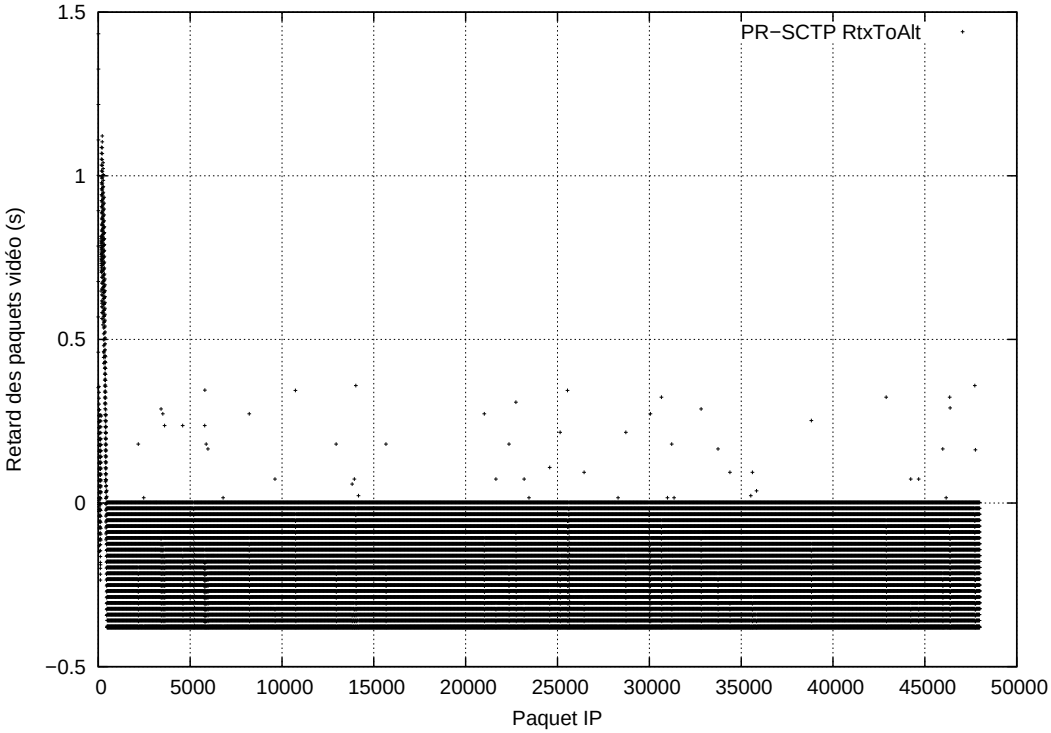

FIG. 5.12 – Dérive temporelle des paquets avec 0.1% d'erreur

Les deux modes SCTP et PR-SCTP utilisant des retransmissions sur l'adresse primaire (`RtxToAlt`) entraîne un taux d'erreur vidéo supérieur à 10% pour un taux d'erreur IP supérieur ou égal à 1,5%. Le problème est identique à celui cité précédemment avec TCP, étant donné que les retransmissions peuvent être retardées par une phase oisive du lien DVB-H. SCTP offre cependant de meilleurs résultats que TCP étant donné que sa fenêtre de congestion évolue plus rapidement en phase de congestion avoidance. (cf. 5.3.3, Fig. 5.8)
La configuration SCTP `RtxToAlt` génère un taux d'erreur vidéo inférieur à 10^{-4} pour un taux d'erreur vidéo inférieur à 1%. Au-delà, le taux d'erreur vidéo est proche de celui du protocole UDP.
En revanche, le mode PR-SCTP `RtxToAlt` limite le taux d'erreur vidéo à une valeur relativement faible pour l'ensemble des taux d'erreur IP appliqués. Les pertes vidéo sont inférieures à 0.1% pour un taux d'erreur IP inférieur à 1%, et reste en moyenne autour de 0,4% pour un taux d'erreur IP variant de 2% à 5%.

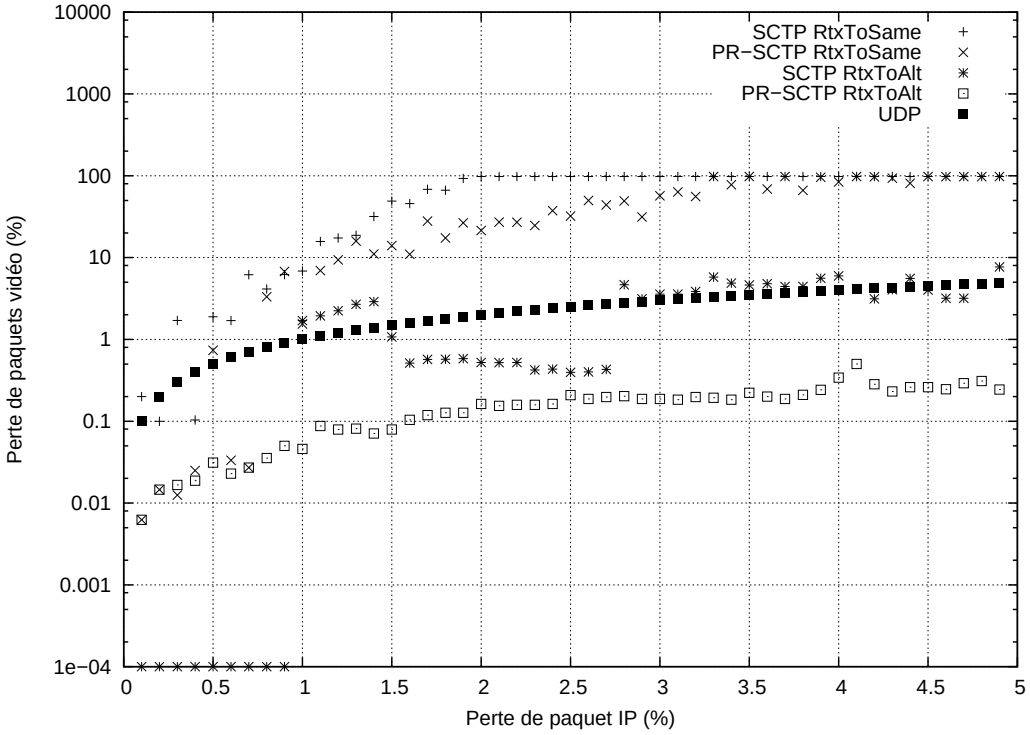

FIG. 5.13 – Performance de la coopération de réseaux pour une diffusion vidéo

5.3.6 Remarques

La combinaison des réseaux UMTS et DVB-H

Dans cette étude, les simulations montrent l'apport d'une architecture de coopération de réseau intégrant le protocole SCTP comme couche transport dans sa pile protocolaire pour un service de diffusion vidéo. Dans cette architecture, la connexion est initialisée par le client sur le lien UMTS, puis le type d'application étant de type asymétrique (diffusion vidéo), le flux de données est renvoyé sur l'interface DVB-H du terminal en changeant son adresse primaire.
L'extension PR-SCTP permet de limiter les retards occasionnés par les retransmissions lors de pertes de paquets tout en réduisant dans une certaine limite le taux d'erreurs perçu au niveau du terminal.
De plus la coopération avec le lien UMTS pour les retransmissions améliore fortement les performances pour des taux de perte importants.

Le protocole DCCP

Dans ces simulations, nous n'avons pas abordé le protocole Datagram Congestion Control Protocol (DCCP) qui permet de transmettre des datagrammes en mode non relié tout en intégrant des mécanismes de contrôle de congestion. En effet ce protocole n'apporte pas de mécanismes de retransmission (ormis pour les datagrammes comprenant certaines options), il est donc aussi sensible aux erreurs de transmission que le protocole UDP. Cependant DCCP peut utiliser différents profils de congestion, Congestion Control ID (CCID), afin d'être compatible avec des flux TCP (TCP Friendly).

Les mécanismes de contrôle de congestion

La version du protocole SCTP actuel utilise les mécanismes de contrôle de congestion décrits dans la RFC 2581 qui ne sont pas adaptés pour des trafics de diffusion. Cependant, de nombreux travaux montrent l'intérêt porté à l'usage du protocole SCTP comme moyen de transport vidéo [70],[71],[72],[73]. Il serait intéressant d'évaluer les performances avec d'autres modèles de contrôle de flux pour des trafics à débit constant (diffusion vidéo) :

- High Speed TCP
- TCP Vegas
- TCP Westwood
- BIC TCP
- H-TCP
- Fast TCP
- TCP Hybla

5.4 Résultats des expérimentations

A travers l'expérimentation de sélection du réseau de plus faible latence, et les simulations de streaming vidéo dans une architecture hybride, nous avons pu valider en partie certaines fonctionnalités du modèle protocolaire proposé.
La fonctionnalité de coopération de réseaux *sélection*, mise en œuvre dans la première expérience montre la faisabilité d'une gestion des liens basée sur les protocoles MIPv6 et SCTP.
La *combinaison* de réseaux, utilisée dans les simulations, souligne les avantages que peut offir la coopération de réseaux dans certaines situations. Dans notre cas de figure, il s'agit d'une amélioration de la fiabilité par des retransmissions limitées.

Ces simulations illustrent également l'usage du protocole SCTP pour divers types d'application telle que la diffusion vidéo, alors que celui-ci était à l'origine un protocole de signalisation pour les réseaux IP.

L'introduction de la nouvelle couche protocolaire NML remet en cause un des paradigmes du modèle TCP/IP en introduisant des interactions entre des couches non adjacentes. En effet le NML peut éventuellement modifier la configuration de la couche réseau, par exemple dans un cas de sélection de réseaux, afin d'appliquer son mode de coopération. De plus le NML se place en partie dans le plan de contrôle de la pile protocolaire (signalisation/configuration entre les couches) et dans le plan de données puisque celles-ci sont transmises au NML avant d'être délivrées à la couche transport. Il serait envisageable de placer le processus NML totalement dans le plan de contrôle, ce qui permettrait de retrouver une certaine indépendance entre les couches courantes du modèle TCP/IP (application, transport et réseau). Le NML résiderait alors en parallèle de l'empilement protocolaire classique, assurant uniquement la configuration des diverses couches. Cependant, cela impliquerait certaines modifications des protocoles actuels, par exemple SCTP afin de gérer la répartition des flux de données sur plusieurs chemins IP lors d'une agrégation de réseaux, ce qui irait à l'encontre de l'approche retenue qui est basée sur l'usage de protocoles strandards actuels.

Conclusion

Dans ce mémoire de thèse, nous avons présenté les travaux effectués autour d'une problématique de coopération de réseaux, du point de vue des hôtes d'extrémité, dans un environnement sans fil, hétérogène et en situation de mobilité. Cette problématique de coopération de réseaux résulte de trois phénomènes majeurs dans le domaine des communications multimédia.
Dans un premier temps, la convergence des services multimédia vers un support full IP fait de ce dernier un protocole incontournable pour le transport de différents types de données. Ensuite, les innovations concernant les nouveaux terminaux mobiles comme l'autonomie, la puissance de traitement, ou l'intégration de plusieurs interfaces permettent le développement de nouveaux services dans un contexte de mobilité. Enfin, le déploiement de nombreux réseaux numériques sans fil tels que Wireless Fidelity (WiFi), DVB ou l'UMTS, capables de transporter des communications IP contribue à l'apparition du phénomène d'Internet ambiant offrant un important potentiel de connectivité IP pour un client léger équipé de plusieurs interfaces. Une seule technologie ne pouvant assurer la couverture totale d'un territoire, un modèle de coopération de réseaux peut apparaître comme une solution alternative pour fournir un accès haut débit efficace vers des terminaux mobiles et multi-interfaces.
Dans ce contexte de réseaux sans fil, multiples et hétérogènes, nous avons proposé une architecture de coopération de réseaux intégrée au niveau des hôtes d'extrémité indépendamment des types de réseaux d'accès. Afin d'assurer un déploiement possible sur des infrastructures réseaux existantes, les principales contraintes étaient de définir une solution indépendante des cœurs de réseaux, des technologies matérielles employées, et des réseaux d'accès.

Dans la recherche de solutions alternatives à l'architecture « classique » UniDirectional Link Routing (UDLR), nous avons montré que la coopération de réseaux peut être réalisée à différents niveaux de la pile protocolaire. En déplaçant les

mécanismes de coopération de réseaux vers la couche transport avec le protocole SCTP, nous avons reporté les processus de coopération de réseaux vers les hôtes d'extrémité tout en conservant des performances similaires voire supérieures au protcole TCP.
Un état de l'art nous a permis de définir le concept de coopération de réseaux et d'identifier quatres fonctionnalités qui peuvent être incluses dans la notion globale de réseaux coopérants.
Tout d'abord, la présence de plusieurs liens de communication doit offrir la possibilité d'un choix, selon divers critères de performances, du réseau le plus adapté au service souhaité. Cette faculté est appelée *sélection*.
De même, l'accès à plusieurs liens doit également pouvoir améliorer la robustesse des communications en cas de défaillance et donc offrir des moyens de *redondance*.
Ensuite, certaines caractéristiques réseaux, comme la bande passante, peuvent être cumulées entre elles. Ainsi dans le cas d'un terminal multi-interfaces, un modèle de coopération de réseaux peut permettre une *agrégation* de bande passante des différents liens disponibles.
Enfin, la *combinaison* de différentes connexions complémentaires, supportant chacune un aspect de la communication (séparation des voies ascendante et descendante entre UMTS et DVB-H), peut former un nouveau type d'architecture de réseaux « hybrides » offrant des performances supérieures à chaque réseau considéré de manière isolée.
Ces quatres fonctionnalités nécessitent de façon implicite la notion de connexion logique afin de caractériser la communication à travers des critères comme le taux de pertes, la latence, la gigue... Dans la recherche de solutions, nous avons donc défini un champ d'investigation comprenant la couche réseau et la couche transport du modèle TCP/IP.
Nous avons défini la coopération de réseaux comme une gestion optimisée des différentes ressources réseaux et connexions potentielles disponibles auprès d'une entité terminale, afin d'accroître les performances d'une communication pour un service donné.

Aucun protocole, de couche réseau ou transport, ne peut apporter à lui seul une solution à notre problèmatique de coopération de réseaux. Cependant la conjonction du protocole IPv6 et du protocole SCTP fournit tous les moyens nécessaires à la réalisation de réseaux coopérants et pouvant être exploités par une entité de niveau supérieur. Les différentes tâches utiles au bon fonctionnement de notre architecture

ont été clairement réparties entre le protocole IPv6, avec son extension de mobilité, et le procotole SCTP, avec ses extensions de fiabilité partielle et de reconfiguration dynamique d'adresses.
D'une part, le protocole MIPv6 assure les fonctions courantes de mobilité que sont la configuration automatique, fournissant une connectivité dans un réseau étranger, la localisation, et la continuité de service lors d'un handover horizontal.
D'autre part, le protocole SCTP assure les handovers verticaux, et prend en charge l'application des quatre opérations de coopération de réseaux : *sélection*, *redondance*, *combinaison* et *agrégation*. Certains paramètres permettant de caractériser les différentes interfaces réseaux sont également récupérés à travers la couche SCTP. D'autres paramètres difficilement évaluables sont assignés par l'utilisateur à chaque interface de communication.

Pour réaliser la coopération de réseaux, nous avons intégré une nouvelle couche système, que nous appellons Network Management Layer, située entre le protocole de transport et l'application. Cette nouvelle entité définit les politiques de coopération de réseaux à partir des contraintes fournies par l'application et des caractéristiques issues des liens de communication. Le NML applique ensuite ces politiques par une configuration de la couche SCTP en utilisant des primitives définies dans l'API standard.
L'API SCTP défini auprès de l'IETF fait office d'interface entre le NML et le protocole, ne nécessitant donc aucun développement pour la communication entre le NML et SCTP. Cependant, il a été indispensable de définir une interface entre la couche NML et l'application. Etant donné le nombre important de profils de trafic applicatif, nous avons limité la caractérisation des applications selon des contraintes simples permettant de qualifier des applications courantes.
Une description SDL du processus NML a permis d'expliciter et de valider les deux modes de fonctionnement proposés, serveur et client, ainsi que la signalisation, permettant d'échanger les informations relatives à la coopération de réseaux, entre le NML serveur et le NML client. Cette signalisation est transportée au sein même de l'association en utilisant le stream 0 du protocole SCTP, protocole qui était à l'origine prévu à cet effet. Nous avons également utilisé une modélisation en réseau de Petri afin de présenter et de valider l'activité globale du système, comprenant les entités serveur et cliente, du processus de coopération de réseaux.

Enfin nous avons évalué notre proposition d'architecture protocolaire à travers

une expérimentation, et des simulations sous NS2.
L'expérimentation consistait en une sélection du réseau de plus faible latence pour un terminal multi-interfaces en situation de mobilité dans un environnement hétérogène. Nous avons ainsi pu montrer la faisabilité du modèle proposé, et valider son bon fonctionnement par rapport aux résultats théoriques attendus.
Les simulations de diffusion vidéo à travers une architecture hybride ont permis de mettre en valeur les bénéfices de la coopération de réseaux en employant les mécanismes de combinaison pour éviter un routage « hybride », et de redondance afin d'améliorer les performances de diffusion pour des taux d'erreur importants.

Dans ce travail de thèse, nous avons défini de façon générale le concept de coopération de réseaux et montré la possibilité d'une intégration sur les hôtes d'extrémité, sans modification des entités réseaux intermédiaires. Un premier modèle de processus de coopération a été proposé à travers le NML. Cependant, plusieurs points nécessitent encore d'être approfondis.
Dans cette première version du NML, les moyens d'évaluation des paramètres réseaux comme le délai, la gigue, ou la bande passante, sont relativement rudimentaires. La recherche de solutions de mesures dynamiques plus efficaces, basées sur les messages HeartBeat, permettrait une prise de décision plus juste au niveau du NML.
De même, l'interface entre l'application et le NML est aujourd'hui limitée à des profils de trafics simples, et ne permet pas de prendre en compte l'ensemble des applications. Une étude sur la possibilité d'une caractérisation des applications à travers une liste exhaustive de contraintes, et leurs associations à des opérations de coopération de réseaux pourrait étendre l'usage du NML pour l'ensemble des types d'applications.
Notons aussi que notre étude a été limitée à des communications bi-points. Une extension au multicast permettrait d'intégrer d'autres applications. Il faudrait alors examiner les aptitudes des protocoles de niveau réseau et transport au multicast dans un contexte de coopération de réseaux, par exemple le protocole de transport Scalable Reliable Multicast Transport Protocol (SRMTP) [74].

Table des figures

Liste des tableaux

Glossaire

best effort

Principe des réseaux IP selon lequel les ressources réseaux ne sont pas réservées à un usage particulier. Les services sont donc fournit au mieux selon les disponibilités. 44

bluetooth

Technologie radio courte distance destinée à simplifier les connexions entre les appareils électroniques. 6

DiffServ

DifServ ou differentiated services est un modèle de mécanisme de qualité de service basé sur la marquage des paquets IP. 84

ethernet

Standard de communication 802.11 permettant d'établir un réseau local informatique. 5

handover

Transfert d'une communication d'un réseaux à un autre. 45, 114–116

indoor

Intérieur. Réception sans antenne extérieure xxii

IP datacasting

Diffusion de donnée IP sur des liens DVB. xxii

load balancing

Technique visant à répartir une certaine charge de travail entre plusieurs processus 12, 86, 91

qualité de service

Mécanisme permettant de favoriser et de protéger des flux d'information qualifiés de fragile (flux vidéo ou audio) 4

réseaux hybrides

Architecture comprenant des réseaux de nature différentes pour former un canal de communication bidirectionnel (ex : RTC/satellite) 12

smooth handover

Handover pendant lequel les réseaux sont temporairement disponibles simultanément. 75

socket

Interface logique de communication pemettant d'exploiter les services d'un protocole réseau. 88, 98, 100

time slicing

Découpage temporel, utilisé pour la diffusion DVB-H permettant de réduire la consommation d'énergie. xxii, 117, 123

triple play

Offre proposant des service de TV, téléphonie et Internet à travers une seule connexion 6

Liste des acronymes

Advanced Research Project Agency (ARPA)

3

Asynchronous Transfert Mode (ATM)

45

Congestion Control ID (CCID)

Profil de contrôle de congestion utilisé par le protocole DCCP. 129

Coded Othogonal Frequency Divisional Multiplexing (COFDM)

Codage d'une transmission à l'aide d'un multiplexage fréquentiel de sous porteuses orthogonales entre elles, séparées par un intervalle de garde. xxi

Common Open Policy Service (COPS)

45

Congestion WiNDow (CWND)

Paramètre utilisé pour le contrôle de flux de TCP et SCTP. 120, 121, 123

Datagram Congestion Control Protocol (DCCP)

Protocole de couche transport permettant de transmettre des donnée en mode non relié tout en intégrant des mécanismes de contrôle de congestion. 128, 129

Delayed Ack (DelAck)

121–125

Differentiated Services (DiffServ)

4

DiffServ Code Points (DSCP)

84

Digital Video Broadcasting (DVB)

Standard de diffusion vidéo numérique. xxi, xxii, 93

Digital Video Broadcasting Cable (DVB-C)

Standard de diffusion vidéo numérique par réseaux cablés. xxi

Digital Video Broadcasting Handheld (DVB-H)

Standard de diffusion vidéo numérique vers des terminaux portable. xxii, 6, 117, 118, 123, 125, 126, 128

Digital Video Broadcasting Satellite (DVB-S)

Standard de diffusion vidéo numérique satellite. xxi, 7, 8

Digital Video Broadcasting Terrestrial (DVB-T)

Standard de diffusion vidéo numérique terrestre. xxi, xxii, 5–7, 10, 89

Digital Video Disc (DVD)

xx

General Packet Radio Service (GPRS)

Technologie de téléphonie mobile de génération 2.5. 6, 10, 86

Global System for Mobile (GSM)

Norme numérique de seconde génération pour la téléphonie mobile. 6

Home Agent (HA)

Routeur du réseau mère connaissant à chaque instant la position du mobile 111

Host Identity Protocol (HIP)

Protocole de couche 3,5 placé entre la couche réseau et la couche transport offrant des servies d'identification, d'authentification et de mobilité 75, 80

Hybrid Network Interconnection System (HNIS)

Routeur assurant l'interconnexion au niveau IP de trois type de réseaux (télécom, informatique, diffusion) 12, 42, 43

Institut National de l'Audiovisuel (INA)

xix

Mobile IPv6 (MIPv6)

Version d'IPv6 comprenant la gestion de la mobilité 75, 80, 82, 114–116, 129

Message Sequence Charts (MSC)

MSC est un langage graphique et textuel pour la description et la spécification des interaction en les divers composants d'un système 98

Network Management Layer (NML)

Couche protocolaire située entre la couche transport et la couche applicative permettant offrant une gestion optimale des ressources réseaux dans un contexte de coopération 82, 83, 85–91, 94, 96–98, 100, 104–106, 108–110, 115

Network Simulator 2 (NS2)

Outils de simulation réseaux courament utilisé dans le domaine de la recherche 117–119

Office de Radio Télévision Français (ORTF)

xix

Personal Digital Assistant (PDA)

6

Partially Reliable SCTP (PR-SCTP)

Extension du protocole SCTP fournissant une fiabilité partielle pour le transport des données. 117, 119, 120, 126–128

peer to peer (P2P)

Modèle de réseau informatique dont les éléments (les nœuds) ne jouent pas exclusivement les rôles de client ou de serveur mais fonctionnent des deux façons, en étant à la fois clients et serveurs des autres nœuds de ces réseaux. 93

Quadrature Amplitude Modulation (QAM)

Modulations d'amplitude à porteuse supprimée en quadrature. xxi

Quality of Service (QoS)

4, 45

Quaternary Phase Shift Keying (QPSK)

Modulation à déplacement de phase à 4 états. xxi

Resource ReSerVation Protocol (RSVP)

4, 45

Réseau Téléphonique Commuté (RTC)

9

Round Trip Time (RTT)

120, 121, 123

Selective ACKnowledgement (SACK)

120, 124

Strongly Connected Components (SCC)

Composante fortement connexe. Une composante d'un graphe est fortement connexe si deux de ses sommets distincts peuvent être joints l'un à l'autre dans les deux sens. 108

Stream Control Transmission Protocol (SCTP)

Protocole de couche transport. 9, 43, 75, 80–82, 86, 88–91, 94, 96, 98, 100, 108, 110, 112, 114, 115, 117, 119–127, 129

SCTP Variable Ack Rate (SCTPVAR)

Version du protocole SCTP modifié afin d'améliorer les performances dans des réseaux fortement asymétrique. 42, 43

Specification and Description Language (SDL)

Langage de spécification et de description de processus communiquant définit par la norme Z.100 auprès de l'ITU-T. 93, 94, 104, 106, 109

Software Defined Radio (SDR)

6

Session Initiation Protocol (SIP)

45

Spanning Tree Protocol (STP)

Protocole de niveau 2 permettant aux commutateurs de détecter de gérer les boucle de commutation. 111

Transmission Control Protocol (TCP)

Protocole de couche transport comprenant un contrôle d'erreur et un contrôle de flux 88, 117, 120, 121, 123, 124, 126, 129

Télédiffusion De France (TDF)

Ancien nom de l'entreprise qui aujourd'hui s'appelle simplement Groupe TDF. xix–xxi

Télévision Numérique Terrestre (TNT)

xx

Type Of Service (TOS)

Le champs service Type Of Service est codé sur 8 bits, il permet la gestion d'une qualité de service traitée directement en couche 3 du modèle OSI. 84

Television over IP (TVoIP)

Utilisation du protocole IP pour diffuser des programmes télévisuels 5

UniDirectional Link Routing (UDLR)

Protocole définissant l'utilisation de liens hétérogènes, avec flux descendants par satellite et flux montants par réseau terrestre. 9, 12, 42, 43

User Datagram Protocol (UDP)

Protocole de couche transport simple ne possédant ni contrôle d'erreur ni contrôle de flux 88, 117, 124–127, 129

Unlicensed Mobile Access (UMA)

6

Universal Mobile Telecommunications System (UMTS)

Technologie de téléphonie mobile de troisième génération. 5, 6, 10, 44, 89, 93, 117, 123, 126, 128

Virtual Local Area Network (VLAN)

Un réseau local virtuel est un réseau local regroupant un ensemble de machines de façon logique et non physique. 111–113

Video on Demand (VoD)

1

Voice over IP (VoIP)

Utilisation du protocole IP pour transporter des conversations téléphoniques 3, 5, 44, 84, 87

Wireless Fidelity (WiFi)

Nom donnée à la certification d'inter-opérabilité des équipements WLAN délivrée par la Wi-Fi Alliance. 5, 6, 10, 86, 89

Publications personnelles

Liste des publications

1. Davy Darche, *Procédé et dispositif de communication de données par paquets à haut débit.*
   - Brevet français
     - Numéro de publication : FR2868642
     - Date de publication : 2005 -10-07 (BOPI 2005 - 40)
     - Numéro de dépôt : FR0403328
     - Date de dépôt : 2004-03-30
   - Brevet européen
     - Numéro de publication : EP1583286
     - Date de publication : 2005 -10-05 (Bulletin 2005 - 40)
     - Numéro de dépôt : EP05290567
     - Date de dépôt : 2005-03-15
     - Numéro de date de priorité : FR0403328 2004-03-30
2. Davy Darche, Francis Lepage, Eric Gnaedinger, *Mobile and Wireless Communication Networks.* IFIP TC6 / WG6.8 Conference on Mobile and Wireless Communication Networks (MWCN 2004) October 25-27, 2004, Paris, France. ISBN : 0-387-23148-X, pages 35 - 45
3. Davy Darche, Francis Lepage, René Kopp, Eric Gnaedinger, Bertrand Mazières, *Using SCTP to improve performances of hybrid broadcast / telecommunication network system.* IEEE Consumer Communications and Networking Conference 2006 (CCNC06) January 8-10, 2006, Las Vegas, Nevada, USA.

Rappels SDL

Le SDL permet de décrire, de simuler et de valider des systèmes comprenant des processus communiquant. Ce langage est principalement utilisé dans le domaine des télécommunications pour concevoir des protocoles de signalisation ou de transport de données. Il existe deux formes de représentation du SDL, la forme GR (Graphical Representation, Fig. 4.3) et la forme PR (Phrase Representation, p. 139).

Liste non exhaustive de composants SDL :

- **block** : élément permettant de regrouper d'autres composants (block, processus,...) au sein d'une même unité.
- **processus** : élément pouvant contenir des états, exécuter des actions, envoyer ou recevoir des signaux.
- **signal** : élément utilisé pour transmettre de l'information entre les processus.
- **channel** : canal logique de communication sur lequel transitent les signaux.
- **gate** : interface d'entrée/sortie entre un processus et un canal.
- **procedure** : routine pouvant être appellée par un processus pour traiter un événement.

Code SDL

nmlt.sdl

```
process type NMLT;
gate G_NML2App out with (AppList);
in with (AppList);
gate G_NML2SCTP out with (SCTPList);
in with (SCTPList);
gate G_NML2NML out with (NMLList);
in with (NMLList);
gate G_NML2Net out with (NetList);
in with (NetList);
gate G_Delay in with DelaySig;
DCL NumNet Integer;
DCL Networks NetTable;
DCL C Constraint;

DCL NetworksOld NetTable;
DCL NumNetSel Integer;
DCL NetworksSel NetTable;
DCL NetworksPrev NetTable;

DCL DelayVal Digit;
DCL i  INTEGER;
start ;
nextstate Idle;
state Idle;
input SCTPShutdown;
nextstate -;
input AppInit(C);
output NetReq VIA G_NML2Net;
nextstate WaitNetResp;
input NMLInit;
output NMLInitOk VIA
G_NML2NML;
nextstate WaitNMLCfg;
endstate;
state WaitNetResp;
input NetResp(NumNet,Networks);
call SelNet(NumNet,
Networks,C);
decision NumNet;
(0):
output AppShutdown VIA
G_NML2App;
nextstate Idle;
else:
output SCTPInit(NumNet,Networks)
VIA G_NML2SCTP;
nextstate WaitSCTPInitOk;
enddecision;
endstate;
state WaitSCTPInitOk;
input SCTPInitOk;
output NMLInit VIA
G_NML2NML;
nextstate WaitNMLInitOK;
endstate;
state WaitNMLInitOK;
input NMLInitOk;
output AppInitOk
VIA G_NML2App;
output NMLCfg(C,NumNet,Networks)
VIA G_NML2NML;
nextstate WaitNMLCfgOk;
endstate;
state WaitNMLCfg;
input NMLCfg
(C,NumNet,Networks)
/* serveur */
```

nmlt.sdl

```
;
output AppStart VIA
G_NML2App;
nextstate WaitAppStartOk;
endstate;
state WaitAppStartOk;
input AppStartOk;
output NMLCfgOk VIA
G_NML2NML;
L5:
decision C!CType;
(Bandwidth):
join L6;
(Delay):
join L3;
enddecision;
endstate;
state DelayProcessing;
input DelaySig
(DelayVal);
task Networks(i)!Delay
:= DelayVal,
i := i+1;
grst1:
decision i = NumNet +1;
(false):
nextstate DelayProcessing;
(true):
call SelNetDelay
(C,NumNet,Networks,
NumNetSel, NetworksSel);
decision NetworksSel =
NetworksPrev;
(false):
output SetPrimaryAddr
(NetworksSel)
VIA G_NML2SCTP;
(true):
enddecision;
task NetworksPrev :=
NetworksSel;
nextstate DelayProcessing;
enddecision;
endstate;
state DelayProcessing;
input AppShutdown;
output SCTPShutdown
VIA G_NML2SCTP;
nextstate Idle;
input NMLCfg
(C,NumNet,Networks);
output NMLCfgOk VIA
G_NML2NML;
join L5;
input none;
join L3;
endstate;
connection L3:
task i := 1;
join grst1;
endconnection L3;
connection L1:
output NMLCfg(C,NumNet,Networks)
VIA G_NML2NML;
nextstate WaitNMLCfgOk;
endconnection L1;
state WaitNMLCfgOk;
```

nmlt.sdl

```
input SCTPShutdown;
output AppShutdown
VIA G_NML2App;
nextstate Idle;
input NMLCfgOk;
grst2:
task NetworksOld :=
Networks;
nextstate ClientProcessing;
endstate;
state ClientProcessing;
input NONE;
output NetReq VIA G_NML2Net;
nextstate ClProcWNetResp;
input SCTPShutdown;
output AppShutdown
VIA G_NML2App;
nextstate Idle;
endstate;
state ClProcWNetResp;
input NetResp(NumNet,Networks);
call SelNet(NumNet,
Networks,C);
decision NetworksOld =
Networks;
(true):
(false):
decision NumNet;
(0):
output AppShutdown VIA
G_NML2App;
nextstate Idle;
else:
join L1;
enddecision;
enddecision;
join grst2;
endstate;
state BandwidthProcessing;
input AppShutdown;
output SCTPShutdown
VIA G_NML2SCTP;
nextstate Idle;
input NMLCfg
(C,NumNet,Networks);
join L5;
endstate;
connection L6:
call SelNetBandwidth
(C,NumNet,Networks,
NumNetSel, NetworksSel);
decision NetworksSel =
NetworksPrev;
(false):
output SetPrimaryAddr
(NetworksSel)
VIA G_NML2SCTP;
(true):
enddecision;
task NetworksPrev :=
NetworksSel;
nextstate BandwidthProcessing;
endconnection L6;
endprocess type NMLT;
```

SelNetDelay.sdl

```
procedure SelNetDelay
; FPAR
IN C Constraint,
IN NumNet Integer,
IN Networks NetTable,
IN/OUT NumNetSel Integer,
IN/OUT NetworksSel NetTable;
DCL i,NumNetTmp INTEGER;
DCL NetTmp NetType;
DCL NetworksTmp NetTable;
start ;
task i:= NumNet;
grst6:
decision i;
(0):
task i := 1,
NumNetTmp :=0;
grst7:
decision i = NumNet+1;
(true):
decision NumNetTmp;
(0:1):
join L2;
else:
join L3;
enddecision;
(false):
decision Networks(i)!Delay<=
C!Value;
(true):
task NumNetTmp :=
NumNetTmp+1,
NetworksTmp(i) :=
Networks(i);
(false):
enddecision;
task i := i+1;
enddecision;
join grst7;
else:
decision Networks(i)!Delay
< Networks(i-1)!Delay;
(false):
task i:=i-1;
(true):
task NetTmp := Networks(i-1),
Networks(i-1) := Networks(i),
Networks(i) := NetTmp,
i:=NumNet;
enddecision;
join grst6;
enddecision;
connection L3:
task NumNet := NumNetTmp,
Networks := NetworksTmp,
i := NumNet;
grst8:
decision i;
(0):
task NumNetSel := 1,
NetworksSel(0) := Networks(1);
return ;
else:
decision Networks(i)!Cost
< Networks(i-1)!Cost;
(false):
task i:=i-1;
```

SelNetDelay.sdl

```
(true):
task NetTmp := Networks(i-1),
Networks(i-1) := Networks(i),
Networks(i) := NetTmp,
i:=NumNet;
enddecision;
enddecision;
join grst8;
endconnection L3;
connection L2:
task NumNetSel := 1,
NetworksSel(0) := Networks(1);
return ;
endconnection L2;
endprocedure SelNetDelay;
```

AppClientT.sdl

```
process type AppClientT;
gate G_App2NML out with (AppList);
in with (AppList);
gate G_Env in with EnvConstr;
DCL C Constraint;

start ;
nextstate Idle;
state Idle;
input EnvConstr(C);
output AppInit(C) VIA
G_App2NML;
nextstate WaitAppInitOk;
endstate;
state WaitAppInitOk;
input AppInitOk;
nextstate Processing;
input AppShutdown;
grst3:
stop ;
endstate;
state Processing;
input AppShutdown;
join grst3;
endstate;
state Processing;
endstate;
endprocess type AppClientT;
```

AppServerT.sdl

```
process type AppServerT;
gate G_App2NML out with (AppList);
in with (AppList);
gate G_EnvApp in with AppStop;
start ;
nextstate Idle;
state Idle;
input AppStart;
output AppStartOk
VIA G_App2NML;
nextstate Processing;
endstate;
state Processing;
input AppStop;
output AppShutdown
VIA G_App2NML;
stop ;
endstate;
state Processing;
endstate;
endprocess type AppServerT;
```

NetworkStatusT.sdl

```
process type NetworkStatusT;
gate G_Env in with EnvNet;
gate G_Net2NML out with (NetList);
in with (NetList);
DCL NumNet Integer;
DCL Networks NetTable;
DCL NetArg NetListType;
start ;
task NumNet := 0;
grst4:
nextstate Idle;
state Idle;
input EnvNet(NetArg);
task NumNet :=
NetArg!Id;
decision NumNet;
(1):
task Networks(1) :=
NetArg!Net1,
Networks(1)!Name :=
NetA;
(2):
task Networks(1) :=
NetArg!Net1,
Networks(1)!Name :=
NetA,
Networks(2) :=
NetArg!Net2,
Networks(2)!Name :=
NetB;
(3):
task Networks(1) :=
NetArg!Net1,
Networks(1)!Name :=
NetA,
Networks(2) :=
NetArg!Net2,
Networks(2)!Name :=
NetB,
Networks(3) :=
NetArg!Net3,
Networks(3)!Name :=
NetC;
else:
enddecision;
nextstate Idle;
input NetReq;
output NetResp
(NumNet,Networks)
VIA G_Net2NML;
join grst4;
endstate;
endprocess type NetworkStatusT;
```

sctpt.sdl

```
process type SCTPT;
gate G_SCTP2NML out with (SCTPList);
in with (SCTPList);
gate G_IPcx out with (SCTPList);
in with (SCTPList);
DCL NumNet Integer;
DCL Networks NetTable;
DCL DestAddr NetTable;
start ;
nextstate Idle;
state Idle;
input SCTPInit
(NumNet, Networks);
output HandShakeStart
(NumNet,Networks)
VIA G_IPcx;
nextstate WaitHandShakeStop;
input HandShakeStart
(NumNet,Networks);
output HandShakeStop
VIA G_IPcx;
grst0:
nextstate Sending;
endstate;
state WaitHandShakeStop;
input HandShakeStop;
output SCTPInitOk
VIA G_SCTP2NML;
nextstate Receiving;
endstate;
state Receiving;
input SCTPShutdown;
output SCTPShutdown
VIA G_SCTP2NML;
nextstate Idle;
endstate;
state Sending;
input SCTPShutdown;
output SCTPShutdown
VIA G_IPcx;
nextstate Idle;
input SetPrimaryAddr
(DestAddr);
task DestAddr :=
DestAddr;
join grst0;
endstate;
endprocess type SCTPT;
```

SelNetBandwidth.sdl

```
procedure SelNetBandwidth
; FPAR
IN C Constraint,
IN NumNet Integer,
IN Networks NetTable,
IN/OUT NumNetSel Integer,
IN/OUT NetworksSel NetTable;
DCL BwMax Digit;
DCL BwLim Digit;
DCL i Integer;
start ;
task i := 1,
BwMax :=1;
grst9:
decision i = NumNet+1;
(false):
decision Networks(i)!Bandwidth
> BwMax;
(true):
task BwMax :=
Networks(i)!Bandwidth;
(false):
enddecision;
task i := i+1;
join grst9;
(true):
task i := 1,
NumNetSel :=0,
BwLim := (C!Value*BwMax)/5+1
;
grst10:
decision i = NumNet+1;
(false):
decision Networks(i)!Bandwidth
>= BwLim;
(true):
task NumNetSel :=
NumNetSel+1,
NetworksSel(NumNetSel)
:= Networks(i);
(false):
enddecision;
task i := i+1;
(true):
return ;
enddecision;
join grst10;
enddecision;
endprocedure SelNetBandwidth;
```

SelNet.sdl

```
procedure SelNet
; FPAR
IN/OUT NumNet Integer,
IN/OUT Networks NetTable,
IN/OUT C Constraint;
DCL NumNetSel Integer;
DCL NetworksSel NetTable;
DCL i Integer;
start ;
task NumNetSel :=0,
i:= 1;
grst5:
decision i = NumNet+1;
(true):
decision NumNetSel;
(0):
else:
decision C!CType;
(Delay):
task NumNet :=
NumNetSel,
Networks :=
NetworksSel;
(Bandwidth):
enddecision;
enddecision;
return ;
(false):
decision Networks(i)!Unidir;
(false):
task NumNetSel :=
NumNetSel+1,
NetworksSel(NumNetSel) :=
 Networks(i);
(true):
enddecision;
task i := i+1;
enddecision;
join grst5;
endprocedure SelNet;
```

IPv6

L'adressage

Les adresses IPv6 sont codées sur 128 bits contre seulement 32 pour le protocole IPv4. Cette adresse est découpée en deux parties, une adresse de réseau et un identifiant d'interface, toutes deux codées sur 64 bits (l'identifiant d'interface est défini à partir de l'adresse Media Access Control (MAC) pour une interface ethernet). Le nombre d'adresses disponibles est donc colossal par rapport à celui d'IPv4. Ce nouveau format d'adresse résout donc le problème de la pénurie d'adresses IPv4, palliée temporairement par les techniques de translation d'adresse. Le plan d'adressage IPv6 est hiérarchisé et agrégé, ce qui signifie que le nombre d'adresse IPv6 est donc inférieur à 2^{128}. Cependant cette structuration des adresses optimise le routage des paquets en réduisant la taille des tables de routage, qui est actuellement un problème majeur dans l'Internet avec le protocole IPv4. L'agrégation comporte trois niveaux, le Top Level Aggregator (TLA) qui le plus élevé, le Next Level Aggregator (NLA) et enfin le Site Level Aggregator (SLA) qui le niveau le plus bas.
Les adresses IPv4 sont souvent exprimées en notation décimale en séparant l'adresse en quatre mots de 8 bits (0-255.0-255.0-255.0-255). Étant donnée la taille des adresses IPv6 et afin de faciliter leurs représentations, on utilise la notation hexadécimale en découpant l'adresse en huit mots de seize bits (FEDC :0000 :0000 :EDBC :A987 :6543 :210F). Une suite de zéros peut être agrégée par " : : " . Naturellement afin d'éviter toute ambiguïté, cette abréviation ne peut être utilisée qu'une seule fois.
Afin de faciliter la gestion des réseaux, le protocole IPv6 permet d'automatiser la phase de renumérotation des hôtes du réseau. A cette fin, les adresses IPv6 possèdent une durée de vie limitée (extensible à l'infini) et ne sont pas attribuées définitivement, mais prêtées pendant un certain laps de temps. Les adresses IPv6 passent par trois états après leur allocation. Le premier de ces

états est qualifié de préféré : l'utilisation n'est aucunement restreinte. Peu avant son invalidation, l'adresse passe dans un état déprécié. Dans cet état, l'utilisation de l'adresse est déconseillée mais pas interdite. L'adresse dépréciée ne doit plus être utilisée comme adresse de source pour de nouvelles communications, mais peut encore servir d'adresse source pour des communications existantes, afin de ne pas interrompre une session (UDP ou TCP) en cours. A l'expiration de la durée de vie de l'adresse, celle-ci passe dans un état invalide. Une interface IPv6 peut donc utiliser simultanément plusieurs adresses pour des communications différentes.
Le protocole IPv6 utilise trois type d'adresses :unicast, multicast et anycast. Les adresses de type broadcast ont été abandonnées et remplacées par des groupes d'adresses multicast prédéfinies (FF02 : :1 = ensemble des nœuds du lien local, FF05 : :2 = ensemble des routeurs du site...). Comme en IPv4, les adresses unicast et multicast désignent respectivement un hôte unique, et un groupe d'hôte identifié par une seule adresse. Comme en multicast, une adresse anycast représente un groupe d'hôte, à la différence qu'un paquet envoyé à une telle adresse sera transmis à l'hôte le plus proche, en terme de métrique de routage, et non à l'ensemble des hôtes.

Format des trames IPv6

Contrairement aux en-têtes IPv4, qui pouvaient avoir une taille comprise entre vingt et quarante octets, l'entête IPv6 est fixe et a une longueur de quarante octets. Ce nouvel entête a été défini afin d'optimiser le traitement des paquets IPv6 au niveau des routeurs.

- L'en-tête IPv6 ne contient plus de checksum, qui devait être modifié par chaque routeur en raison de la décrémentation du champ de durée de vie. Cette fonctionnalité est assurée par les couches supérieures, qui intègre un pseudo-entête contenant l'adresse source et destination IPv6 dans leurs calculs de checksum.
- L'entête étant fixe, la zone de données utiles est clairement définie.
- Les options sont retirées de l'en-tête elle-même, et sont placées dans de nouveaux entêtes, appelés extensions et qui sont situés après l'entête principal. A part l'extension de proche en proche, elles sont ignorées par les routeurs intermédiaires, et ne sont traitées que par les hôtes d'extrémités.
- Les champs sont alignés sur des mots de 64 bits qui optimisent leur traite-

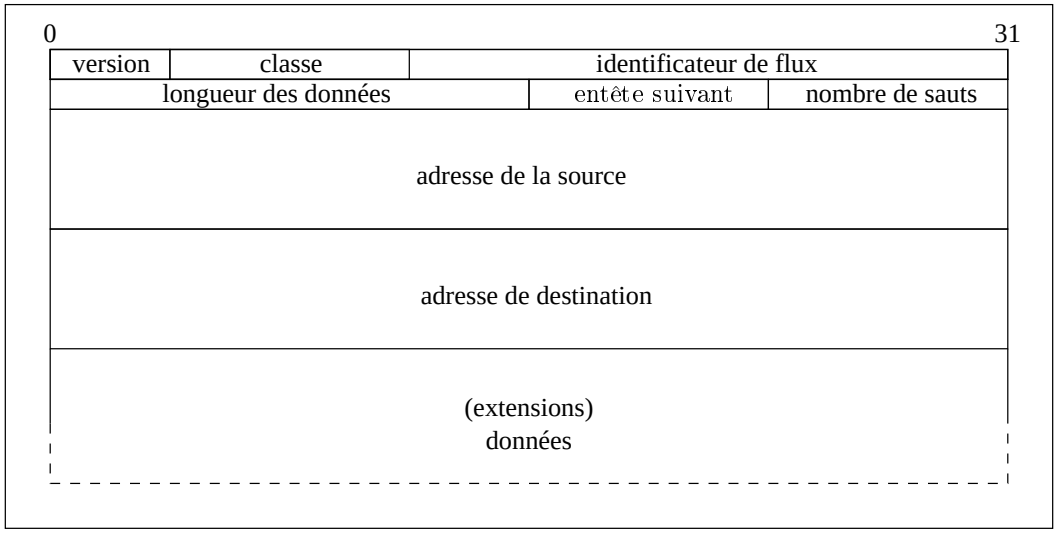

FIG. 14 – Format d'un paquet IPv6

ment, en particulier avec des architectures 64 bits.

- La taille minimale des MTU (Maximum Transmission Unit) est de 1280 octets. Le choix de 1280 comme MTU minimal en IPv6 a été pris pour permettre l'encapsulation des paquets IPv6. En effet, la taille de 1 500 octets est généralement admise car elle correspond à la valeur imposée par Ethernet.
- La fonction de fragmentation a été retirée des routeurs. Les champs qui s'y rapportent (identification, drapeau, place du fragment) ont été supprimés. Normalement les algorithmes de découverte du PMTU (Path MTU) évitent le recours à la fragmentation [32].

La figure 14 représente le format d'une trame IPv6. Le premier champ, codé sur 4 bits, indique la version du protocole, sa valeur est 6. Le champ classe spécifie la classe de trafic, et est équivalent au champ DiffServ des paquets IPv4 (lui-même remplaçant l'ancien champ Type Of Service (TOS)). L'identificateur contient un numéro unique choisi par la source qui a pour but de faciliter la mise en œuvre des fonctions de qualité de services comme Resource ReSerVation Protocol (RSVP). L'utilisation de cet identifiant optimise la commutation des paquets dans les routeurs. En IPv4 les routeurs définissent un contexte à partir de la session spécifiée par le paquet (adresse source et destination et port source et destination). La longueur des données spécifiée dans l'en-tête IPv6 représente uniquement la taille des données utiles, contrairement à IPv4 qui prenait en compte la longueur de l'entête. Le champ en-tête suivant a une

fonction similaire au champ protocole du paquet IPv4. Il identifie le prochain entête. Il peut s'agir d'un protocole de niveau supérieur ou de la désignation d'une extension.

Bibliographie

[1] DVB Project. http ://www.dvb.org.

[2] Peter Shelswell. COFDM : The modulation system for digital radio, November 18 1999.

[3] J. H. Stott. The how and why of COFDM, November 18 1998.

[4] A. G. Burr. Performance of COFDM for multimedia transmission on the personal communication channel. In *International Conference on Universal Personal Communications*, volume 1, pages 269–273, San Diego, CA, USA, Octobre 1997. IEEE.

[5] Wolfgang Eberle, Veerle Derudder, Geert Vanwijnsberghe, Mario Vergara, Luc Deneire, Liesbet Van der Perre, Marc G. E. Engels, Ivo Bolsens, and Hugo De Man. Digital Modulation :OFDM Solves Mobility and High Rate Problems. *IEEE journal of solid-state circuits*, 36, Novembre 2001.

[6] Zhang Di. Performance analysis and comparison of OFDM based packet transmission system with QAM modulation.

[7] Andres Arjona. Internet Protocol DataCasting. Helsinky University of Technology.

[8] Wei Li, Hong Liu, and Gilles Gagnon. Integration of an interactive multimedia datacasting system. In *IEA/AIE*, pages 325–334, 2004.

[9] Jean Michel Cornu. Deux visions de l'après-internet : NGN et STP/SP. www.fing.org, mars 2005.

[10] Milton L. Mueller. Digital convergence and its consequences. *Javnost The Public*, 1999.

[11] David G. Messerschmitt. The Future of Computer and Telecommunications Integration. *IEEE Communications Magazine*, pages 66–69, Avril 1996.

[12] Decina Maurizio and Trecordi Vittorio. Convergence of Telecommunications and Computing to Networking Models for Integrated Services and Applications. *Proceedings of the IEEE*, 85(12) :1887–1914, Décembre 1997.

[13] DARPA. Internet Protocol. RFC 791, septembre 1981.

[14] DARPA. Transmission Control Protocol. RFC 793, septembre 1981.

[15] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog and S. Jamin. Resource ReSerVation Protocol (RSVP). RFC 2205, septembre 1997.

[16] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss . An Architecture for Differentiated Services. RFC 2475, décembre 1998.

[17] UMA Technology. http ://www.umatechnology.org.

[18] The Official Bluetooth Website. http ://www.bluetooth.com/.

[19] M. Berg, S. Butterfi eld, J. Cosmas, P. Casagranda, D. Garrec, M.Guiraudou, G. Martinez, E. Launay, B. Mazieres, and D. Milanesio. Cismundus : Convergence of digital broadcast and mobile telecommunications, 2004. http ://dea.brunel.ac.uk/project/Cismundus/.

[20] D. Johnson, C. Perkins, and J. Arkko. Mobility support in ipv6. RFC 3775, juin 2004.

[21] E. Duros, W. Dabbous, H. Izumiyama, N. Fujii and Y. Zhang. A Link-Layer Tunneling Mechanism for Unidirectional Links. RFC 3077, mars 2001.

[22] Davy Darche, Francis Lepage and Eric Gnaedinger. TCP performances in a hybrid broadcast/telecommunication system. In *Proceedings of the Sixth IFIP IEEE International Conference on Mobile and Wireless Communication Networks 2004*, october 2004.

[23] H. Balakrishnan, V. N. Padmanabhan, G. Fairhurst, and M. Sooriyabandara. TCP Performance Implications of Network Path Asymmetry. RFC 3449, décembre 2002.

[24] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581, avril 1999.

[25] Seok Joo Koh, Moon Jeong Chang and Meejeong Lee. mSCTP for Soft Handover in Transport Layer. *IEEE COMMUNICATIONS LETTERS, VOL. 8, NO. 3*, mar 2004.

[26] Rajesh Rajamani, Sumit Kumar and Nikhil Gupta. SCTP versus TCP : Comparing the Performance of Transport Protocols for Web Traffic, juillet 2002. http ://www.cs.wisc.edu/sumit/extlinks/sctp.pdf.

[27] Sourabh Ladha and Paul D. Amer. Improving Multiple File Transfers Using SCTP Multistreaming, mai 2003. http ://www.cis.udel.edu/amer/PEL/poc/pdf/TR2003-06.FTP.over.SCTP.Ladha.pdf.

[28] Karl-Johan Grinnemo, Torbjorn Andersson, and Anna Brunstrom. Performance benefits of avoiding head-of-line blocking in SCTP. In *ICAS/ICNS*, 2005.

[29] International Organization for Standardization. Open system interconnection. norme ISO IS7498, 1978.

[30] S. Bradner and A. Mankin. The Recommendation for the IP Next Generation Protocol. RFC 1752, janvier 1995.

[31] S. Deering and R. Hinden. Internet Protocol Version 6 (IPv6) Specification. RFC 2460, décembre 1998.

[32] J. McCann and S. Deering and J. Mogul. Path MTU Discovery for IP version 6. RFC 1981, août 1996.

[33] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. Host identity protocol, septembre 2006. draft-ietf-hip-base-05.

[34] R. Moskowitz and P. Nikander. Host Identity Protocol Architecture, février 2006. draft-ietf-hip-arch-03.

[35] T. Henderson. End-host mobility and multihoming with the host identity protocol, jul 2005.

[36] T. Bova and T. Krivoruchka. Reliable UDP Protocol. draft-ietf-sigtran-reliable-udp-00.txt, février 1999.

[37] Gene Ma. T/UDP : Udp for TCAP. draft-ma-tudp-00.txt, mai 1999.

[38] David Sanchez and Miguel A. Garcia. A Simple SCCP Tunneling Protocol (SSTP). draft-sanchez-garcia-SSTP-v1r0-00.txt, juillet 1999.

[39] David Sanchez. Connectionless SCCP over IP Adaptation Layer (CSIP). draft-sanchez-CSIP-v0r0-00.txt, mai 1999.

[40] K. Toney. Reliable Transport Extensions on UDP. draft-toney-purdet-00, septembre 1999.

[41] R. Stewart and Q. Xie. Multi-Network Datagram Transmission Protocol. draft-ietf-sigtran-mdtp-06.txt, jan 1999.

[42] R. Stewart, Q. Xie and K. Morneault, C. Harp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang and V. Paxson. Stream Control Transport Protocol. RFC 2960, octobre 2000.

[43] L. Coene. Stream Control Transmission Protocol Applicability Statement. RFC 3257, avril 2002.

[44] P. Amer, T. Connolly, C. Chassot, P. Conrad and M. Diaz . Partial order transport service for multimedia and other applications . *IEEE/ACM Trans on Networking*, octobre 1994.

[45] Mika Ratola. Which Layer for Mobility ? - Comparing Mobile IPv6, HIP and SCTP, mar 2004.

[46] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad. *Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration*, jun 2005.

[47] Janardhan R. Iyengar, Paul D. Amer and Randall Stewart. Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths. *IEEE/ACM transactions on networking*, 2006. à paraître.

[48] T. Saadawi A. Abd El Al and M. Lee. Load Sharing in Stream Control Transmission Protocol. draft-ahmed-lssctp-01.txt, mai 2005.

[49] J. Iyengar, P. Amer, and R. Stewart. Receive Buffer Blocking In Concurrent Multipath Transfer. In *Globecom*. IEEE, Novembre 2005.

[50] Zeina Jrad, Badr Benmammar, Joeseph Correa, Francine Krief, and Nader Mbarek. A userassistant for QoS negotiation in a dynamic environment using agent technology. In *Proceedings of second IFIP International Conference on Wireless and Optical Communications Networks WOCN'05*, mars 2005.

[51] Francine Krief. Self-aware management of IP networks with QoS guarantees. *Journal of Network Management*, 2004.

[52] Foo Kong Yong and Wan Tat Chee and Sureswaran Ramadass. M-SCTP : Transport Layer Multicasting Protocol, juin 2005.

[53] P. Almquist. Type of service in the internet protocol suite. RFC 1349, juillet 1992.

[54] K. Nichols, S. Blake, F. Baker and D. Black . Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474, décembre 1998.

[55] Van Jacobson. pathchar - a tool to infer characteristics of Internet paths. In , avril 1997.

[56] Robert L. Carter and Mark E. Crovella. Measuring Bottleneck Link Speed in Packet-Switched Networks. Boston University Computer Science Department, mars 1996.

[57] Manish Jain and Constantinos Dovrolis. End-to-End Available Bandwidth : Measurement methodology, Dynamics, and Relation with TCP Throughput, août 2002.

[58] R. Stewart and Q. Xie and L. Yarroll and J. Wood and K. Poon and M. Tuexen . Sockets api extensions for stream control transmission protocol (sctp), septembre 2005. draft-ietf-tsvwg-sctpsocket-11.txt.

[59] "Jukka Ytialo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, and Jaakko Laine". Dynamic Network Interfaces Selection in Multihomed Mobile Hosts, 2003.

[60] Yasuyuki TANAKA and Mitsunobu KUNISHI and Fumio TERAOKA. PM-PATH : A Policy Routing System for Multihomed End-Hosts, January 2006.

[61] IPSOS. Sondage profiling 2003, dec 2003.

[62] University of Aarhus. CPNTools. http ://wiki.daimi.au.dk/cpntools/cpntools.wiki.

[63] Debian. http ://www.debian.org/.

[64] Linux Kernel SCTP. http ://lksctp.sourceforge.net/.

[65] The Linux Kernel Archive. http ://www.kernel.org/.

[66] Mobile IPv6 for Linux. http ://www.mobile-ipv6.org/software/.

[67] Debian. http ://linux-net.osdl.org/index.php/Netem/.

[68] R. Koodli. Fast Handovers for Mobile IPv6. RFC 4068, juillet 2005.

[69] "R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad ". Stream Control Transmission Protocol (SCTP) Partial Reliability Extension . RFC 3758, mai 2004.

[70] Mohamed N. El Derini and Amr A. Elshikh. MPEG-4 Video Transfer with SCTP-Friendly Rate Control. In *Proceedings of the second International Conference on Innovations in Information Technology, IIT'05*, 2005.

[71] M. Molteni and M.Villari. Using SCTP with Partial Reliability for MPEG-4 Multimedia Streaming. In *Proceedings of the BSDCon*, 2002.

[72] H. Huang and J. Ou and D. Zhang (PRC). Efficient Multimedia Transmission in Mobile Network by using PR-SCTP. In *Proceedings of the Communications and Computer Networks*, october 2005.

[73] Antonios Argyriou. A novel end-to-end architecture for H.264 video streaming over the internet. *Telecommunication Systems*, 28(2) :133–150, 2005.

[74] Stephan Block, Ken Chen, Philippe Godlewski and Ahmed Serhrouchni. Some Design Issues of SRMTP, a Scalable Reliable Multicast Transport Protocol. In *Proceedings of Multimedia Applications, Services and Techniques 4th European Conference*, mai 1999.

[75] Davy Darche, Francis Lepage, René Kopp, Eric Gnaedinger and Bertrand Mazières. Using SCTP to improve performances of hybrid broadcast/telecommunication network system. In *Proceedings of the IEEE Consumer Communications and Networking Conference 2006*, january 2006.

[76] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden. A survey of active network research. *IEEE Communications Magazine*, 35(1) :80–86, 1997. citeseer.ist.psu.edu/tennenhouse97survey.html.

[77] Allen B. Downey. Using pathchar to estimate internet link characteristics. In *SIGCOMM*, pages 241–250, 1999.

[78] D. Durham, Ed., J. Boyle, R. Cohen, S. Herzog, R. Rajan and A. Sastry. The COPS (Common Open Policy Service) Protocol. RFC 2748, janvier 2000.

[79] J Rosenberg, H. Schulzrine, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler. SIP : Session Initiation Protocol. RFC 3261, juin 2002.

[80] Gustavo Carneiro and Carlos Garcia and Pedro Neves and Zhikui Chen and Michelle Wetterwald and Manuel Ricardo and Pablo Serrano and Susana Sargento and Albert Banchs. The DAIDALOS Architecture for QoS over Heterogeneous Wireless Networks. Technical report, IST, juin 2005.

[81] J.H. Stott. The how and why COFDM. *EBU Technical review*, 1998.

[82] James Noonan, Philip Perry, and John Murphy. Client controlled network selection, 2004. Computer Science Department, University College Dublin, Dublin, Ireland.

[83] J. Laganier and L. Eggert. Host identity protocol (hip) rendezvous extension, jul 2005.

[84] Ed. C. Perkins. Ip mobility support for ipv. RFC 3220, janvier 2002.

[85] François Baccelli and Ki Baek Kim. Tcp throughput analysis under transmission error and congestion losses. Technical report, INRIA, octobre 2003.

[86] E. Altman and K. Avrachenkov and C. Barakat. TCP Network Calculus : The case of large delay-bandwidth product. In *Proceedins of IEEE INFOCOM Conference*, juin 2002. http ://citeseer.ist.psu.edu/altman02tcp.html.

[87] S. Dawkins and G. Montenegro and M. Kojo and V. Magret and N. Vaidya. End-to-end Performance Implications of Links with Errors, août 2001. RFC 3155.