Architecture de Communication pour les Applications Multimédia Interactives dans les Réseaux Sans Fil

Frédéric Nivor

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse III – Paul Sabatier

Discipline ou spécialité : Systèmes Informatiques

Présentée et soutenue par Frédéric NIVOR

Le 15 Juillet 2009

Titre : ***Architecture de Communication pour les Applications Multimédia Interactives dans les Réseaux Sans Fil***

JURY

Abdelmalek BENZEKRI

Olivier FOURMAUX

Olivier ALPHAND

Pascal BERTHOU

Slim ABDELLATIF

Invité : Fabrice ARNAL

École Doctorale : École Doctorale Systèmes

Unité de Recherche : LAAS-CNRS

Directeur de thèse : Michel DIAZ

Rapporteurs : Francine KRIEF, Véronique VÈQUE

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse III – Paul Sabatier

Discipline ou spécialité : Systèmes Informatiques

Présentée et soutenue par Frédéric NIVOR

Le 15 Juillet 2009

Titre : ***Architecture de Communication pour les Applications Multimédia Interactives dans les Réseaux Sans Fil***

JURY

Abdelmalek BENZEKRI

Olivier FOURMAUX

Olivier ALPHAND

Pascal BERTHOU

Slim ABDELLATIF

Invité : Fabrice ARNAL

École Doctorale : École Doctorale Systèmes

Unité de Recherche : LAAS-CNRS

Directeur de thèse : Michel DIAZ

Rapporteurs : Francine KRIEF, Véronique VÈQUE

AUTEUR :
Frédéric NIVOR

TITRE :
Architecture de Communication pour les Applications Multimédia Interactives dans les Réseaux sans Fil

DIRECTEUR DE THÈSE :
Michel DIAZ

LIEU ET DATE DE SOUTENANCE :
LAAS-CNRS, Toulouse France, 15 Juillet 2009

RÉSUMÉ en français

Les travaux de cette thèse s'inscrivent dans le contexte des réseaux sans fil et des réseaux d'accès par Satellite en particulier. Ces derniers présentent certains inconvénients lorsqu'il s'agit de déployer des applications multimédia interactives qui requièrent un délai de bout en bout aussi faible que possible et plus généralement exigent une meilleure Qualité de Service (QdS) du système de communication.

Dans ces travaux de thèse, nous proposons d'utiliser les informations de signalisation de session des applications multimédia basées sur le protocole de session SIP afin d'ajuster le paramétrage du système de communication selon une approche « cross-layer » qui permet alors d'améliorer de façon significative la réactivité du système.

Afin de faciliter l'intégration et l'implémentation des solutions proposées dans un système de communication réel, un mécanisme de communication inter-couches d'optimisation est proposé et développé.

Les solutions proposées sont évaluées dans des environnements sans fil émulés et réels.

MOTS-CLÉS :
Réseaux sans Fil, Satellite, DVB-S2/RCS, WiFi, 802.11e, WiMax, 802.16, Qualité de Service, QoS, SIP, Cross-Layer, Applications Multimédia, VoIP

DISCIPLINE ADMINISTRATIVE :
Systèmes Informatiques

LABORATOIRE DE RATTACHEMENT :

LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse Cedex

TABLE DES MATIERES

Liste des figures et tables

Introduction générale

Les applications multimédia interactives fournissent des environnements de communication très complets pouvant être utilisés dans le cadre du travail collaboratif synchrone ou encore dans un contexte ludique et familial. Cela va de la communication audio et/ou vidéo jusqu'aux applications partagées, en passant par les tableaux blancs. Ces applications, par leur aspect interactif, requièrent un service de qualité du système de communication sous-jacent afin de fonctionner correctement. Au niveau réseau, ce service, variable selon les applications, s'exprime par exemple en termes de garanties sur le délai de bout en bout, le débit de transmission ou encore le taux de perte d'information.

Jusque là, les applications multimédia ont été largement déployées et supportées sur les infrastructures filaires telles que la fibre optique comme lien en cœur de réseau, ou encore la technologie xDSL (Digital Subscriber Line) pour la distribution finale.

De nos jours, de nouvelles technologies réseau ont fait leur apparition. Il s'agit en particulier de la technologie « sans fil » qui permet un déploiement facile et rapide d'infrastructures réseau sur des sites géographiques non accessibles à la technologie filaire (xDSL). Ces technologies réseau sans fil d'accès sont complémentaires par leur portée : on retrouve le Wifi à l'échelle LAN (Local Area Network ou réseau local), le Wimax à l'échelle MAN (Metropolitan Area Network ou réseau métropolitain) et les réseaux d'accès Satellite à l'échelle WAN (Wide Area Network ou réseau étendu).

Cependant ces technologies réseau sans fil souffrent de certains points faibles. Par exemple, dans les systèmes satellite, les ressources radio sont à la fois limitées et coûteuses. De plus le délai de transmission est relativement long. Une utilisation optimale et efficace du système de communication est alors nécessaire pour assurer la diffusion des services satellite au grand public.

En déployant les applications multimédia interactives sur de tels réseaux, de nombreux problèmes apparaissent. Dans ces travaux de thèse, nous adressons plus particulièrement ceux concernant l'interopérabilité des couches de communication. Bien que non exhaustifs, ces problèmes constituent une première barrière au déploiement des applications multimédia interactives sur les réseaux sans fil.

Problématique adressée

L'interconnexion des réseaux à technologie sans fil avec le reste du réseau de communication Internet nécessite l'utilisation des protocoles actuellement déployés sur Internet (pile TCP/IP). L'ensemble de ces protocoles, tout comme le modèle de référence ISO/OSI [1] sont basés sur le paradigme de conception en couche : chaque couche est chargée de fournir un ou plusieurs services spécifiques à la couche située au dessus. Les communications s'effectuent uniquement et directement entre couche adjacente.

Il y a plusieurs avantages à cette approche : la modularité, la robustesse et la conception sont facilement réalisées :

- Sachant qu'une couche doit remplir certaines fonctions, l'effort de conception est uniquement concentré sur ces fonctions sans se préoccuper des couches supérieures ou

inférieures. Seules les interfaces nécessitent d'être définies de sorte que les services fournis par cette couche puissent être accessibles par les autres couches.

- La modularité fournie par les couches permet une combinaison arbitraire des protocoles. D'un point de vue architectural, cette modularité améliore également la maintenance lorsqu'une nouvelle version d'un protocole est à insérer sans avoir à changer le reste de la pile réseau.

Cependant, dans les environnements à technologie sans fil, mobiles et satellite, les propriétés des différentes couches ont des interdépendances substantielles. Une conception modularisée peut donc être sous-optimale quant à l'exécution et à la disponibilité dans les réseaux sans fil. En effet, les modes de transmission et les modèles de trafic pour ces technologies sans fil sont très différents de ceux des réseaux filaires.

Un exemple typique aux environnements sans fil est la perte de paquets due aux mauvaises conditions des canaux de transmission sans fil qui est interprétée par la couche de transport, plus particulièrement par le protocole TCP, comme étant une situation de congestion au niveau réseau. Ceci entraine la diminution du débit de transmission de la communication en cours. Appliqué à un réseau d'accès par satellite, cette communication mettra alors un délai important avant de retrouver un débit nominal.

Ainsi, afin de répondre aux exigences strictes de Qualité de Service des utilisateurs et des applications multimédia interactives, le système de communication doit pouvoir s'adapter dynamiquement aux situations du trafic et aux conditions réseau. Ce besoin ne peut être adressé par l'architecture réseau protocolaire traditionnelle.

Comment satisfaire les besoins des applications multimédia (point de vue utilisateur) sur des réseaux à technologie « sans fil » tel que le satellite, et ceci, sans gaspiller les ressources disponibles (point de vue opérateur réseau) ? Telle est la problématique de cette thèse. Elle peut être présentée sous forme de deux objectifs à atteindre:

- Fournir un service de qualité aux utilisateurs d'applications multimédia interactives (VoIP, visio-conférence) : ces utilisateurs ne se soucient guère de la gestion des ressources du réseau, mais attendent plutôt un service correct;
- Optimiser l'utilisation des ressources radio du système de communication, en particulier dans un environnement réseau satellite: ces ressources étant à la fois limitées et onéreuses.

Solution proposée et plan du mémoire

Afin de répondre à ces besoins, nous proposons d'utiliser les caractéristiques du flux multimédia telles que le débit de transmission, afin de paramétrer au mieux le système de communication. Ces caractéristiques sont obtenues à partir de la signalisation de session SIP utilisée en début de communication lorsque les applications multimédia négocient les paramètres de la session multimédia à venir. Puis une approche « Cross-Layer » est alors adoptée afin de transmettre ces informations au système de communication sous-jacent. Une conception Cross-Layer, est, d'un point de vue, une approche coopérative où l'adaptation est coordonnée entre les couches multiples du système de communication. Des interactions peuvent être alors établies entre des couches adjacentes, ou même non adjacentes dans la pile. Cette approche Cross-Layer viole la hiérarchie en

couche du modèle de référence Internet. Cependant, la flexibilité résultant de cette approche cross-layer permet de mettre à disposition des informations pertinentes à toutes les couches du système de communication. Ces informations sont alors utilisées pour configurer de manière optimale l'ensemble du système de communication afin d'optimiser l'utilisation des ressources réseau, tout en fournissant un service de qualité aux utilisateurs d'applications multimédia.

La première partie du mémoire présente un état de l'art descendant de la Qualité de Service des systèmes actuels. En partant du haut de la pile, la première couche, application, repose sur le système de communication sous-jacent. Ses besoins en QdS sont donc définis par les applications elles mêmes. Puis pour les couches sous-jacentes (session, transport, réseau, liaison), les services fournis et plus particulièrement les mécanismes de QdS mis en place pour améliorer la communication sont présentés.

Nous proposons dans la seconde partie du mémoire, de présenter l'approche dite « Cross-Layer ». Les caractéristiques et architectures existantes de cette approche sont détaillées.

La troisième partie présente trois contributions basées sur l'approche Cross-Layer. Elles utilisent les informations de sessions et ont pour but à la fois de : (i) fournir un service de qualité aux applications multimédia, (ii) optimiser l'utilisation des ressources du système de communication dans un environnement à technologie sans fil. Elles sont appliquées à un réseau satellite DVB-S2/RCS et plus particulièrement pour les mécanismes d'allocation dynamique de ressources. La première contribution tend à améliorer l'allocation dynamique de ressources lors du démarrage d'une session multimédia dans le réseau satellite. La seconde intervient durant la session multimédia. La troisième contribution améliore le compromis entre optimisation des ressources réseau et qualité de service garantie aux applications multimédia : elle adapte le contrôle d'admission de nouvelles sessions multimédia et le contrôle de débit des sessions en cours, en fonction des conditions de charge du réseau.

En quatrième partie du mémoire, une architecture cross-layer basée sur une approche avec entité intermédiaire est proposée. Elle permet de formaliser les interactions cross-layer précédemment évoquées. Sa spécification et sa mise en œuvre sont présentés.

Une connaissance précise des besoins en QdS des applications multimédia interactives est nécessaire afin de déclencher les différents services QdS au sein du réseau sous-jacent. Nous utilisons alors dans la cinquième partie de ce mémoire, une architecture orientée Web Services qui répond à ce besoin. Mis à part les informations de besoins en QdS des applications interactives multimédia, elle permet la découverte et l'invocation de l'ensemble des services disponibles dans le domaine satellite. Ce service fourni par cette architecture sera alors utilisé par les autres contributions de ce travail de thèse.

La sixième partie de ce mémoire présente l'évaluation des différentes contributions et architectures proposées durant ces travaux. Enfin la septième partie permet de conclure les travaux de cette thèse et d'en présenter les perspectives.

1. Garantie de Qualité de Service dans les systèmes de communication

La garantie de **Qualité de Service** (QdS) est la capacité à fournir et garantir différentes priorités à différents utilisateurs, différentes applications, ou encore à différents flux de données dans un système de communication. Ces priorités, appliquées aux flux de données, sont caractérisées par des garanties sur le débit, le délai, la gigue, la probabilité de perte de paquets de données et le taux d'erreur sur bit. Ces garanties de Qualité de Service prennent tout leur sens lorsque les capacités de transmission du système sont insuffisantes par rapport à l'ensemble des flux de données à transmettre.

Nous proposons dans ce chapitre de faire un état de l'art non exhaustif sur les mécanismes de QdS dans les systèmes de communication actuellement déployés, qui sont pour la plus part basés sur l'architecture TCP/IP [2]. Les couches de cette architecture sont parcourues afin d'identifier les fonctionnalités QdS proposées.

1.1. Couche Application

La couche application est au sommet du modèle. Toutes les autres couches sous-jacentes ont pour objectif de proposer des services afin de répondre aux besoins de la couche la plus haute. Ces besoins, dans le cadre de la gestion de la Qualité de Service, sont caractérisés par des métriques de performance telles que le taux de perte, le délai de bout en bout, etc.

L'ITU-T (International Telecommunications Union) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. Elle émet des recommandations en vue de la normalisation des télécommunications à l'échelle mondiale. À partir de la perception de l'utilisateur, l'ITU-T fournit un certain nombre de recommandations en termes de performance pour les applications [3]. Cette perception est caractérisée par différents paramètres qui permettent de la quantifier. En voici les principaux :

- Le délai de bout en bout, pour une communication point à point, représente le temps nécessaire pour transmettre les informations utiles une fois que le service est établi. Il est l'un des paramètres qui agit directement sur la satisfaction de l'utilisateur selon l'application.
- La variation du délai, communément appelée gigue, représente la variabilité inhérente des dates d'arrivée des paquets d'informations. Tout comme le délai de bout en bout, une gigue importante a une conséquence directe sur la satisfaction de l'utilisateur selon l'application.
- La perte d'informations représente à la fois les erreurs sur les bits d'informations, les pertes de paquets en cours de transmission, ou encore les pertes d'informations introduite par le codage des médias afin d'en faciliter la transmission. Elle a un effet direct sur la qualité des informations finalement présentées à l'utilisateur, quelque soit leur type (voix, vidéo, données, etc.).

Ces paramètres de performances perceptibles par l'utilisateur sont utilisés pour caractériser quelques principales applications susceptibles d'être utilisées par l'utilisateur final sur les réseaux. Ces applications sont avant tout des exemples, et non une liste exhaustive. Elles sont classées en trois types de médias : audio, vidéo et donnée.

1.1.1. Applications à média audio

Dans la catégorie d'applications à média audio, on retrouve trois classes d'applications :

1.1.1.1. Audio Conférence, Voix sur IP

Les applications d'audio conférence véhiculent des données audio, plus particulièrement des données de voix, et ceci de manière bilatérale. Selon le délai de bout en bout unilatéral, deux effets sont remarqués sur les applications d'audio conférence :

- Des délais de l'ordre de dizaines de millisecondes créent un écho durant la conversation. Des mesures de réduction d'écho doivent être prises à ce stade [4].
- Lorsque le délai augmente (plusieurs centaines de millisecondes), il crée une désynchronisation dans la conversation et devient très perceptible par le correspondant utilisateur [5].

En ce qui concerne la gigue, l'oreille humaine y est sensible. La suppression de cette gigue est nécessaire et possible au moyen de tampons de compensation de gigue : à l'arrivée des paquets à destination, ils sont mis en tampon. Cependant, comme tout tampon, il introduit un délai supplémentaire. Ce dernier doit rester faible pour ne pas retomber sur le problème précédemment évoqué.

Pour les pertes d'informations, l'oreille humaine peut tolérer un certain degré de distorsion d'un signal vocal due à des pertes survenues durant le processus d'encodage.

En dehors de ce type de perte d'informations, il est nécessaire de considérer les erreurs isolées sur les bits. Ces dernières peuvent être introduites durant la transmission par un lien de communication de mauvaise qualité. Certains mécanismes de codage audio tels que AMR (Adaptive Multi-Rate), ILBRC (Internet Low Bit Rate Codec), tolèrent les erreurs sur bits, en utilisant des mécanismes de recouvrement (c.f. section 1.3.3). Cependant, d'autres codages y sont beaucoup plus sensibles, et ces erreurs introduisent des perturbations perceptibles par l'utilisateur. Dans ce dernier cas, un taux de pertes nul est requis.

1.1.1.2. Messagerie vocale

La messagerie vocale permet de laisser un message à données audio (ex. voix) à un destinataire. Comme toute messagerie classique, le service de délivrance du message s'effectue de manière asynchrone : sa lecture par le destinataire est donc différée. La tolérance au délai et à la gigue est donc complètement différente de celle de l'application audio conférence. Le délai à considérer ici est celui entre l'instant où l'utilisateur destinataire lance l'ordre de lecture du message vocal, et la production effective du signal sonore de ce message vocal. La tolérance de l'utilisateur face à ce délai reste difficile à quantifier, cependant quelques secondes devraient suffire à ne pas impatienter l'utilisateur.

Par contre, la tolérance aux pertes d'informations sont pratiquement les mêmes que celles des applications audio conférence car le média est du même type.

1.1.1.3. Streaming Audio

Les applications de streaming audio offre un service d'écoute musical à la demande. Tout comme la messagerie vocale, elle implique une conversation unilatérale. Cependant, en termes de perte de

paquets, les besoins sont beaucoup plus sévères, car les données audio sont censées être de bonne, voire de haute qualité.

1.1.2. Applications à média vidéo

Les applications à média vidéo regroupent essentiellement deux grandes classes d'applications :

1.1.2.1. Vidéo Conférence

Les applications de vidéo conférence véhiculent à la fois les médias audio et vidéo, et ceci de manière bilatérale. Par conséquent les prescriptions sont les mêmes que pour l'audio conférence. Par contre les deux medias (audio et vidéo) nécessitent d'être synchronisés mutuellement afin d'assurer la « synchronisation avec les lèvres ».

Tout comme l'oreille humaine, l'œil humain peut tolérer un certain degré de pertes d'informations visuelles dues au codage vidéo.

1.1.2.2. Streaming Vidéo

Les applications de streaming vidéo, communément appelées Vidéo à la Demande (Video on Demand : VoD) impliquent une communication unilatérale. Ainsi le délai et la gigue ne présentent pas nécessairement de contrainte. De même pour la synchronisation des medias audio et vidéo, elle peut être réalisée à la réception, par mise en tampon.

La Table 1 [3] récapitule les besoins en QdS des applications à média audio et vidéo en termes de métriques quantifiables :

Applications	Symétrie	Débits	Délai unilatéral	Gigue	Taux de Perte d'informations
Audio Conférence	Bilatéral	4 - 64 kbit/s	< 150 ms préféré < 400 ms limite	< 1 ms	< 3 %
Vidéo Conférence	Bilatéral	16 – 384 kbit/s	< 150 ms préféré < 400 ms limite	< 1 ms	< 1 %
Messagerie Vocale	Unilatéral	4 – 32 kbit/s	< 1 s (reproduction) < 2 s (enregistrement)	< 1 ms	< 3 %
Streaming Audio	Unilatéral	16 – 128 kbit/s	< 10 s	<< 1 ms	< 1 %
Streaming Vidéo	Unilatéral	16 – 384 kbit/s	< 10 s	< 1 ms	< 1 %

Table 1: Objectif de qualité pour les applications audio et vidéo

1.1.3. Applications de données

Les recommandations pour les applications de données ont un critère commun, à savoir un taux de perte nulle. En ce qui concerne la gigue, elle n'est généralement pas perceptible par l'utilisateur. La tolérance au délai cependant varie selon les applications de données.

1.1.3.1. Navigation Web

Le rôle de ces applications est d'extraire et de consulter un composant HTML d'une page web. Lorsque d'autres composants, telles que les images et les séquences audio/vidéo entrent en jeu, une

synchronisation de ces flux médias est nécessaire (par exemple, une donnée audio combinée a une présentation sur tableau blanc).

Du point de vue de l'utilisateur, ce dernier souhaite que les pages demandées apparaissent suffisamment rapidement. Des délais inférieurs à environ 10 secondes restent acceptables.

1.1.3.2. Transfert de fichiers

Les recommandations pour le transfert de fichiers sont intimement liées à la taille du fichier à transmettre. Du point de vue de l'utilisateur, il est requis d'avoir une indication de l'état d'avancement.

1.1.3.3. Services de transaction à priorité élevée

Dans, le cadre des services de transaction à priorité élevée, l'utilisateur a besoin d'être rassuré sur le bon déroulement de la transaction (ex. opération bancaire). Pour cela, des délais plutôt faibles (maximum quelques secondes) sont nécessaires afin de rassurer l'utilisateur sur le bon déroulement de l'opération.

1.1.3.4. Images fixes

Les images fixes, tout comme les medias audio et vidéo, sont généralement codées et par conséquent, un certain degré de perte due au codage est toléré. Cependant, certains codages sont moins tolérants aux erreurs isolées sur les bits, car ces pertes peuvent provoquer d'importantes perturbations dans l'image. Dans ce cas, un taux de perte d'informations quasiment nul est requis.

Les recommandations pour le délai de transfert des images fixes s'apparentent à celles pour le transfert de données. Un affichage progressif durant la réception de l'image offrirait a l'utilisateur un état d'avancement du transfert.

1.1.3.5. Jeux interactifs

L'interactivité requise est très dépendante du type de jeu : en effet, des jeux temps réel en ligne nécessitent de très courts délais (inferieur à la seconde), compatibles avec l'interactivité requise.

1.1.3.6. Telnet

Afin d'assurer un retour en écho de caractères quasiment instantané, il est nécessaire que, pour une application telle que Telnet, le délai soit court (inferieur à la seconde).

1.1.3.7. Courier électronique

Le courrier électronique, possède, en matière de délai, les mêmes propriétés que la messagerie vocale : c'est un service différé, il peut donc tolérer des délais de l'ordre de quelques secondes et plus.

1.1.3.8. Messagerie instantanée

La messagerie instantanée, généralement appelée « chat » transmet essentiellement des données textuelles. Mais elle peut également inclure des données audio, vidéo et graphiques. Bien que ces applications soient interactives, les délais de transmission peuvent être noyés dans le délai nécessaire à la saisie de ce message (ex. par clavier). Ainsi des retards de plusieurs secondes sont acceptables.

Compte tenu de l'analyse précédente, les objectifs de qualité pour les applications de données sont résumés dans la Table 2 :

Applications	Symétrie	Quantité de données	Délai unilatéral	Taux de Perte d'informations
Navigation Web	Unilatéral	~ 10 Ko	< 2 s par page préféré < 4 s par page acceptable	0 %
Transfert de fichiers	Unilatéral	~ 10 Ko	< 15 s préféré < 60 s acceptable	0 %
Transaction prioritaire	Bilatéral	< 10 Ko	< 2 s préféré < 4 s acceptable	0 %
Images fixes	Unilatéral	< 100 Ko	< 15 s préféré < 60 s acceptable	0 %
Jeux interactifs	Bilatéral	< 1 Ko	< 200 ms	0 %
Telnet	Bilatéral asymétrique	< 1 Ko	< 200 ms	0 %
Courrier électronique	Unilatéral	< 10 Ko	< 2 s préféré < 4 s acceptable	0 %

Table 2 : Objectif de qualité pour les applications de données

1.1.4. Plate-forme multimédia distribuée de collaboration

Les plates-formes multimédia distribuées sont utilisées comme solution d'apprentissage collaboratif à distance pour un groupe distribué d'utilisateurs, où chacun peut joindre indépendamment une session.

Deux types d'interaction peuvent être mises en place : la collaboration asynchrone qui ne requiert pas la présence des participants au même instant, et la collaboration synchrone où les utilisateurs travaillent en coprésence.

Ces plates-formes peuvent être utilisées dans différents contextes : « e-learing », « co-design », travail coopératif, etc.

Nous pouvons citer l'environnement Platine développée au LAAS-CNRS [6]. Cette plate-forme est principalement dédiée aux collaborations synchrones. Elle intègre un certain nombre de composants : un module de préparation de sessions off-line, un affichage d'états de session, un gestionnaire de session, une vidéoconférence multiutilisateur, un chat multipoint, un module de partage d'application, un tableau blanc partagé.

Comme nous pouvons le constater, ces plates-formes multimédia sont composées de plusieurs types d'applications, dont les différents flux de communication requièrent une qualité de service particulière. De plus, certains flux nécessitent une synchronisation entre eux afin d'assurer une présentation correcte des différents médias à l'utilisateur.

1.1.5. Classification des applications en catégories d'exigences QdS

La Figure 1 résume le classement des applications précédemment évoquées sur un référentiel ayant pour abscisse le délai unilatéral recommandé et en ordonnée le taux de perte d'informations.

Chaque type d'application est représenté sous forme d'un rectangle qui représente les limites de délai et de pertes d'informations tolérables par l'utilisateur.

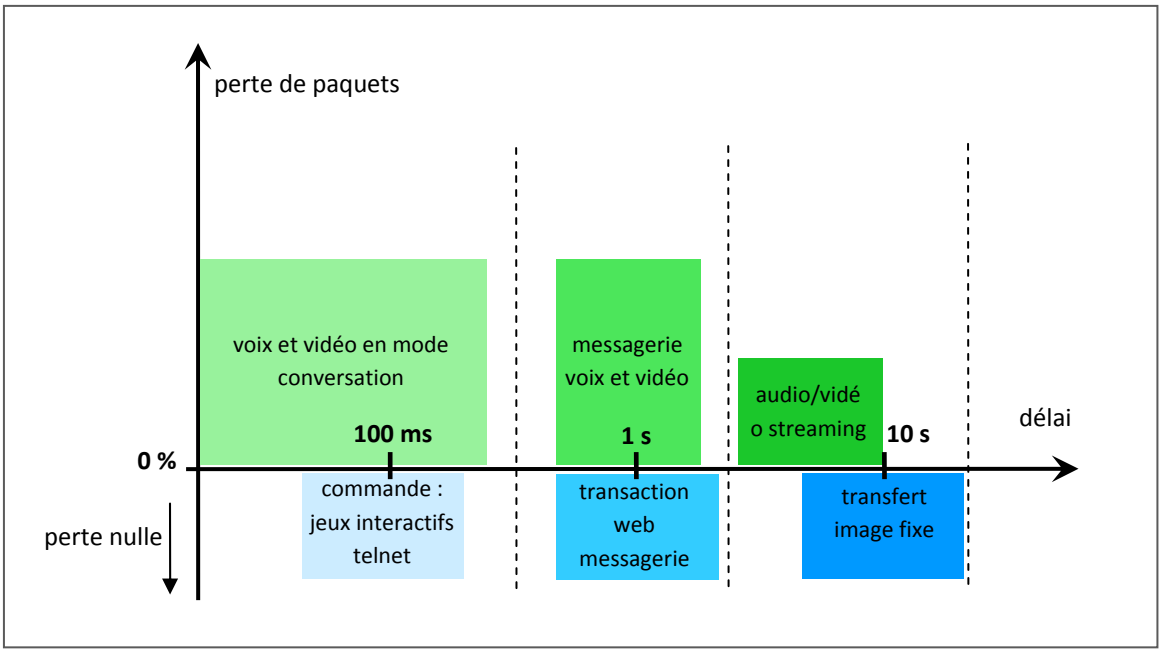


Figure 1 : Classes d'applications en fonction du délai et des pertes

À partir de cette figure, il apparait six groupements distincts, qui couvrent l'étendue des applications identifiées. Ces derniers peuvent être formalisés et ainsi assimilés à des catégories de QdS pour l'utilisateur final.

Cependant, quelques points méritent d'être éclaircis :

- Ces catégories de QdS sont fondées sur la perception des dégradations de bout en bout par l'utilisateur. Elles ne sont dépendantes d'aucune technique ou technologie particulière. Par conséquent elles peuvent être appliquées à n'importe quelles techniques de transport sous-jacente (IP, ATM, ligne filaire, liaison hertzienne, etc.) ;
- Ces catégories de QdS sont caractérisées par des paramètres avec des bornes. Ces limites supérieures et inférieures permettent à l'utilisateur final de définir un service comme étant acceptable ou pas. Un service qui dépasse une limite supérieure (de perte ou de délai) est considéré comme insatisfaisant. Un service en dessous de la limite inférieure est considéré satisfaisant. Cependant, cela sous-entend qu'il gaspille les ressources du système de communication ;
- Ces catégories de QdS ne représentent que des regroupements d'applications par métriques de performance. Par conséquent elles ne forment qu'une base de classes de QdS de réseau afin de différencier la qualité de service.

1.1.6. Conclusion

À travers ce début d'état de l'art, nous avons pu spécifier les attentes de l'utilisateur concernant différentes catégories d'applications. Les besoins en QdS ont été définis sur la base de la tolérance aux pertes d'informations et aux délais. Parmi ces applications, la voix et la vidéo en mode

conversation bilatérale possèdent des caractéristiques de performance élevées : faible délai de bout en bout, gigue quasi nulle, taux de perte minimal.

Nous proposons maintenant de présenter les services offerts par les couches sous-jacentes des systèmes de communication. Nous nous intéresserons plus particulièrement aux mécanismes et fonctionnalités capables de répondre aux exigences des applications multimédia interactives.

Dans la mesure où une couche protocolaire implémentée fournit un service à la couche juste supérieure, nous commencerons par présenter les services des couches basses en remontant aux couches hautes, ceci dans un souci de lisibilité du mémoire.

1.2. Sous-couche MAC

La couche Liaison se charge de délivrer des données entre deux nœuds d'un même segment réseau. Elle est composée généralement de deux sous couches, la sous couche basse de contrôle d'accès au medium (Media Access Control : MAC), et la sous couche haute de contrôle de lien logique (Logical Link Control : LLC). La sous couche haute LLC s'occupe principalement de la gestion des erreurs. La sous couche basse MAC fournit l'accès et le partage du support physique entre les nœuds.

Dans le contexte de la qualité de service, nous allons nous intéresser plus particulièrement à la sous couche MAC et aux mécanismes de qualité de service des technologies sans fil à l'échelle LAN, MAN et WAN. Pour les réseaux LAN, nous évoquerons la technologie réseau 802.11 (WiFi) et 802.11e (WMM : WiFi MultiMedia) sachant qu'elle est la communément répandue pour cette taille de réseau. Pour les réseaux à l'échelle MAN, nous nous intéresserons à la technologie 802.16, encore appelée WiMax qui dans sa catégorie de réseau intègre plusieurs mécanismes de QdS. Et en ce qui concerne les réseaux à l'échelle WAN, nous nous focaliserons sur la technologie Satellite de type DVB-S2/RCS utilisée pour le déploiement des services IP dans les zones géographiques reculées.

1.2.1. Technologie 802.11(e) (WiFi)

Le réseau local sans fil (WLAN ou Wireless Local Area Network) 802.11 [7] est un système de transmission des données standardisé par l'IEEE (Institute of Electrical and Electronics Engineers). Il fonctionne dans les bandes de fréquences 2.4, 3.6 et 5Ghz avec des débits variants entre 11 et 108Mbit/s. Il permet d'assurer une liaison indépendante de l'emplacement des périphériques informatiques qui composent le réseau et utilisant les liens sans fil plutôt qu'une infrastructure câblée. Dans l'entreprise, les WLAN sont généralement mis en œuvre comme le lien final entre le réseau câblé existant et un groupe d'ordinateurs clients, offrant ainsi aux utilisateurs de ces machines un accès sans fil à l'ensemble des ressources et des services du réseau de l'entreprise, sur un ou plusieurs bâtiments.

1.2.1.1. Architecture

Dans le standard 802.11, une station sans fil (STA) représente l'interface air de l'utilisateur. Deux modes d'utilisation des réseaux WLAN sont définis :

- le mode infrastructure, où une station de base, appelée AP (Access Point), joue le rôle de pont entre le WLAN et le réseau filaire. Avant toute communication, les STAs doivent exécuter une procédure d'association avec l'AP. Ce dernier est donc l'élément central d'un WLAN.
- Le mode ad hoc où les STAs communiquent directement entre elles sans avoir recours à un AP ou une connexion à un réseau filaire. Ce mode permet de créer rapidement et simplement un WLAN là où il n'existe pas d'infrastructure filaire. Ainsi, il est doté de capacités qui lui permettent de s'organiser et de se configurer de manière autonome.

1.2.1.2. Mécanismes d'accès

Le canal est divisé en intervalles temporels appelés supertrame. La supertrame est divisée en deux périodes :

- une période d'accès sans contention (Contention Free Period : CFP) durant laquelle les STAs accèdent en utilisant le mécanisme d'accès centralisé PCF (Point Coordination Function).

- une période d'accès avec contention (Contention Period : CP). Les STAs y accèdent en utilisant le mécanisme d'accès distribué DCF (Distributed Coordination Function) ;

Procédure d'accès PCF

Le mécanisme d'accès au canal PCF (Point Coordination Function) définit une entité appelée le point de coordination PC (généralement l'AP). Cette entité PC gère l'accès au canal. Il interroge à tour de rôle (polling) les STA afin de leur offrir l'opportunité de transmettre. Le PC accède au canal plus rapidement que les autres stations classiques car il possède des délais d'attente plus faibles. Ce mode d'accès est cependant optionnel, par conséquent très peu déployé.

Procédure d'accès DCF

Le mécanisme d'accès DCF (Distributed Coordination Function) est basé sur CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). Les stations gèrent entre elles l'accès distribué et équitable du canal.

Chaque station souhaitant émettre, surveille l'inactivité du canal durant un délai appelé *backoff.* Ce délai est choisi aléatoirement dans un intervalle temporel CW (Contention Window). Le décompte du délai backoff est interrompu temps que la STA détecte une activité sur le canal. La première station libérée du délai backoff, gagne l'accès au canal. Cependant, avant d'émettre sur le canal, il utilise le mécanisme d'accès RTS/CTS (Request to Send / Clear to Send) afin de palier à d'éventuels problèmes de stations « cachées ». La station émet une petite trame RTS au destinataire. Si l'émission réussit sans collision, le récepteur répond avec une autre petite trame CTS. Ainsi la station peut transmettre sa trame de données sur le canal.

Limitations du standard 802.11

Les méthodes d'accès proposées par le standard 802.11 présentent un certain nombre d'inconvénients.

Dans le mode d'accès centralisé PCF, tant que le canal est occupé par une communication entre station, le PC n'a plus le contrôle du canal, ce qui engendre des délais imprévisibles.

Avec le mécanisme d'accès distribué DCF, la bande passante est partagée entre toutes les stations de manière équitable ; il n'y a pas de notion de priorité. Des garanties en termes de bande passante, de délai ou de gigue sont difficiles à obtenir.

Actuellement, la qualité de service offerte dans les équipements existants est celle du « best effort » : les performances dépendent étroitement du nombre d'utilisateurs.

1.2.1.3. Mécanismes de Qualité de Service IEEE 802.11e

Afin d'apporter un support de qualité de service, le groupe IEEE 802.11 propose des améliorations intégrées dans un nouveau standard **802.11e**. Ces modifications introduisent deux nouvelles méthodes d'accès, EDCA (Enhanced Distributed Channel Access) et HCCA (Hybrid Controlled Channel Access), chacune basée respectivement sur les mécanismes d'accès standard DCF et PCF précédemment décrits.

Un nouveau concept, l'opportunité de transmission (TXOP) est introduit dans 802.11e, à la fois pour EDCF et HCF. TXOP détermine une date de début et une durée pendant laquelle une station a le droit

d'émettre. Ainsi, une station peut transmettre plusieurs trames dans la même opportunité TXOP, tant qu'elles ne s'étendent pas au delà.

Procédure d'accès distribué améliorée : EDCA

Le mécanisme EDCA (Enhanced Distributed Channel Access) propose un service de différentiation : quatre catégories d'accès (Access Category : AC) au niveau MAC sont définies. Elles sont implémentées au sein des stations sous forme de différentes files de transmission. Chaque catégorie d'accès est considérée comme une entité de transmission MAC indépendante, car elle possède ses propres paramètres tels que la borne haute CW. Ainsi pour chaque files d'accès i, en faisant varier la taille de l'intervalle CW[i] pour l'obtention du délai backoff, on offre un accès au canal différencié entre les stations, mais également au sein des stations. La catégorie d'accès la plus prioritaire est celle qui possède les plus petites valeurs d'attente. La Figure 2 [8] représente quatre files d'émission implémentées dans une station, chaque file supportant une catégorie d'accès. Comme nous pouvons le constater, le trafic à transmettre provenant de la couche supérieure doit être étiqueté (marqué) avec une priorité adéquate afin d'être dirigée dans la file d'attente à QdS correspondante.

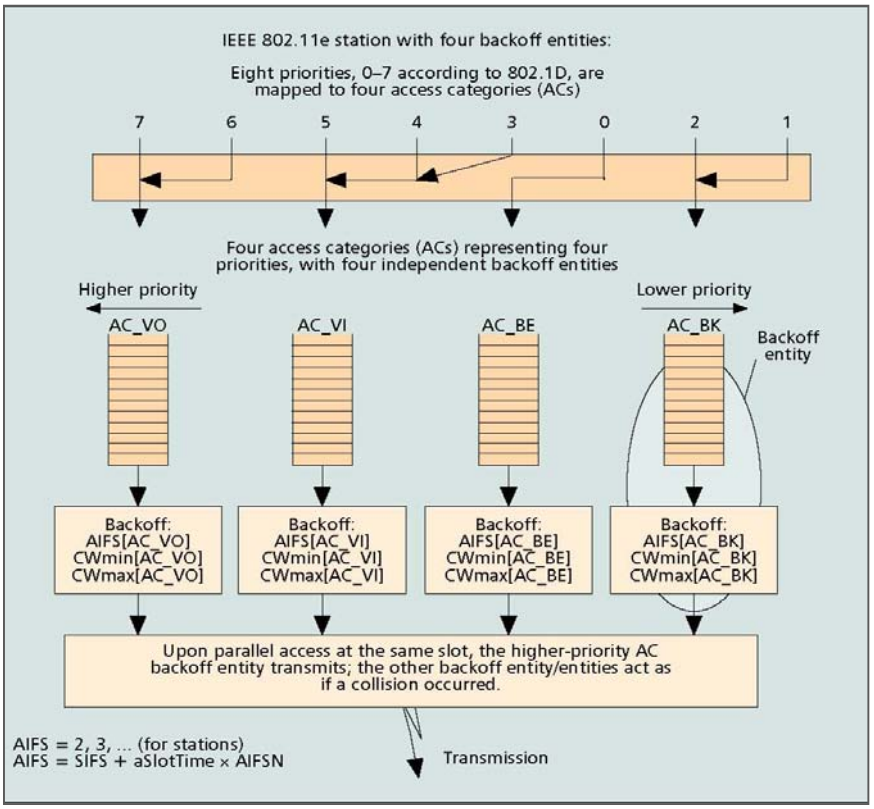


Figure 2 : station 802.11^{e} ([8])

Procédure de scrutation contrôlée HCCA

Dans le mode de fonctionnement HCCA (HCF (Hybrid Coordinator Function) Controlled Channel Access), les stations sont toujours interrogées à tour de rôle par l'AP. Ce dernier devient une entité de coordination hybride HC (Hybrid Coordinator). Les stations requièrent le paramètre TXOP auprès du HC avant d'émettre. Ce dernier, basé sur une politique d'admission de contrôle accepte ou pas la requête. Si cette dernière est acceptée, le HC programme et transmet les TXOPs aux stations.

1.2.1.4. Conclusion

Malgré une portée moyenne, la technologie réseau 802.11 offrent un certain nombre de caractéristiques, telles que : un débit moyen de 54Mbit/s, un déploiement et une configuration instantanée, un gain de mobilité, un coût réduit. La version 802.11e vient compléter le standard, avec des mécanismes de priorité et de différentiation de trafic, à la fois au sein du réseau, mais également au sein des stations, offrant ainsi un support QdS aux applications exigeantes.

1.2.2. Technologie réseau 802.16 (WiMax)

Dans le cadre des nouvelles technologies réseau sans fil, le haut débit sans fil n'avait jusqu'à présent pas réussi à devenir une plate-forme de boucle locale radio haut débit (Broadband Wireless Access : BWA). Afin de tenir ces objectifs, l'organisme IEEE a proposé un nouveau standard de transmission radio : 802.16, plus communément appelée WiMAX (Worldwide Interoperability for Microwave Access).

Cette technologie sans fil est considérée comme un complément de la technologie 802.11 car, grâce à une portée allant jusqu'à 50km, une bande de fréquence entre 10 et 66 Ghz (très réglementée) et un débit de transmission maximal théorique de 70Mbit/s, elle permet d'établir l'interconnexion de différents points d'accès des réseaux WiFi. Ainsi, elle accroît la portée et le débit de la boucle locale radio. Elle permet aussi de desservir les zones géographiquement reculées.

1.2.2.1. Architecture

Il existe deux types d'équipements : la station utilisateur SS (Subscriber Station) et la station centrale BS (Base Station) qui possède des fonctionnalités de gestion et de contrôle du réseau.

On retrouve deux topologies du WLAN, à savoir en étoile où toutes les communications se font entres les SSs et la BS, et maillé (mesh) où les communications directes entre SSs sont possibles (cf. Figure 3).

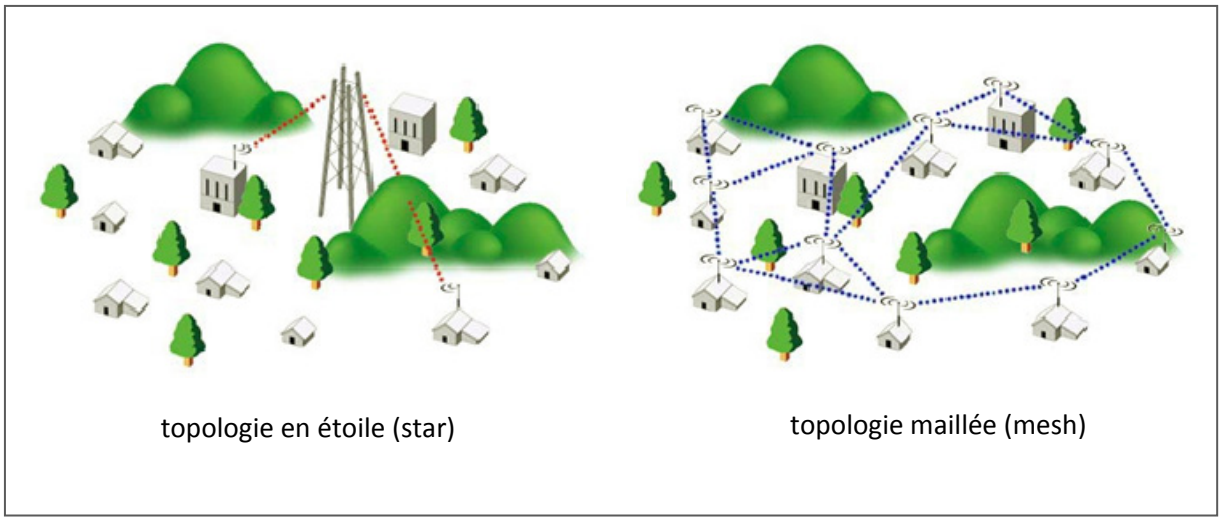


Figure 3 : Topologies réseau 802.16

Chaque topologie possède bien évidemment ses avantages et ses inconvénients mais l'on retiendra qu'une topologie en étoile acceptera d'avantage de capacité en terme de Mbps/km² qu'un réseau

maillé qui à l'inverse acceptera de son côté un déploiement plus rapide et moins contraint par les exigences de ligne de vue directe entre équipements.

1.2.2.2. Caractéristiques

La couche MAC du 802.16 fournit un service orienté connexion aux couches supérieures, ce qui suppose un protocole de mise en place de la connexion entre la SS et la station de base BS. Ceci permet la négociation et la mise à jour de paramètres de connexion tels que la bande passante, la QdS, les paramètres de trafic, le type de transport, etc. Trois paires de connexions (voie montante et descendante) sont établies entre un SS et la BS pour :

- L'échange des messages de gestion MAC avec contraintes temporelles.
- Les messages de gestion MAC, plus tolérants au délai.
- L'échange de messages standards (DHCP (Dynamic Host Configuration Protocol), SNMP (Simple Network Management Protocol), TFTP, etc.), tolérants au délai.

Ces trois paires de connexions sont inhérentes à trois niveaux de QdS pour le trafic de gestion MAC entre les SS et la BS.

La couche MAC du 802.16 est divisée en 3 sous couches distinctes :

- La couche de convergence CS (Convergence Sublayer) : la première tâche de cette sous couche est de classifier les unités de données de services SDU (Service Data Unit) vers la connexion MAC adéquate. Elle prend en charge le transport des cellules ATM mais aussi des paquets IP.
- La couche commune (Common Part Sublayer) : offre les fonctionnalités de base de contrôle d'accès, d'allocation de bande passante, de garantie de QdS, d'établissement et de maintien de connexion.
- La couche de protection de donnée (Security Sublayer) : repose sur le protocole Privacy Key Management (PKM). Elle offre des mécanismes d'authentification, d'échange de clés sécurisés et de cryptages tel que l'AES (Advanced Encryption Standard).

Nous allons donc nous intéresser plus particulièrement à cette sous couche commune MAC qui offre un support de QdS.

1.2.2.3. Méthodes d'allocation de ressources

La BS attribue dynamiquement de la bande passante pour la voie montante (SS vers BS) et la voie descendante (BS vers SS). La voie descendante utilise le mécanisme de multiplexage temporel TDD (Time Division Duplex). La voie montante utilise quant à elle l'accès multiple à temps divisé TDMA (Time Division Multiple Access). La taille des trames ainsi que la taille de chaque intervalle de temps peuvent varier, sous le contrôle de la BS.

Les stations SS, afin d'avoir accès au canal pour émettre leurs données, demandent l'autorisation auprès de la BS afin que cette dernière leur fournisse des allocations de transmission sur la voie montante. Ces allocations sont implémentées en utilisant différentes procédures qui sont :

- « Unsolicited bandwidth grants » ou allocation de bande non réclamée,
- « Polling » ou allocation de bande par système d'interrogation des SS,
- « Contention procedure » ou allocation de bande selon un mécanisme de contention.

Mécanisme d'allocation sans contention (Polling)

Le polling est le processus par lequel la BS alloue spécifiquement des slots temporels aux SS afin qu'elles puissent effectuer des requêtes de bande passante. La BS interroge à tour de rôle les SS sur leurs besoins de transmission.

Mécanisme d'allocation avec contention

D'autres slots temporels peuvent être alloués par la BS pour les SS afin d'émettre leur requête de bande passante. Ces requêtes sont alors transmises par la méthode de contention en utilisant, comme pour le 802.11, une fenêtre d'intervalle d'attente (minimum et maximum) backoff window.

Une requête de retransmission automatique ARQ (Automatic Repeat reQuest) peut être utilisée afin de solliciter la transmission de MSDU non fragmentés ou de fragments de MSDU qui ont été perdus ou endommagés.

Les requêtes de bande passante peuvent être incrémentales : la BS accumule la bande passante précédemment demandée par la SS ; ou agrégées : la BS remplace la bande passante demandée par la SS.

L'attribution de bande passante se fait suivant deux modes: GPSS (Grant Per SS), où l'attribution se fait soit de manière globale pour une SS ; GPC (Grant Per Connection), l'attribution n'est que pour la dite connexion.

1.2.2.4. Qualité de Service dans la couche MAC 802.16

Afin de répondre aux besoins des applications multimédia, la couche MAC du 802.16 propose des services QdS différenciés pour supporter les exigences de ces applications. Nous présentons leurs applications ainsi que les méthodes d'accès utilisées.

- Le service UGS (Unsolicited Grant Service) :
  - *Application* : supporte les flots temps-réels qui génèrent périodiquement des paquets de données de taille fixe, tel que la voix sur IP.
  - *Méthode d'accès* : La SS n'effectue pas de requêtes de bande passante. Elle est négociée à l'initialisation de la station et lui est allouée statiquement tout au long de ce service.
- Le service rtPS (real-time Polling Service) :
  - *Application* : offre des requêtes d'opportunités temps-réel périodiques permettant aux SS de spécifier la taille de leurs données. Ce service génère plus de requêtes que l'UGS mais supporte des demandes de données de taille variable.
  - *Méthode d'accès* : La SS reçoit les opportunités de transmission périodiques qui lui sont attribuées par polling de la BS. L'utilisation des opportunités de requêtes par contention est interdite pour les flux rtPS.
- Le service nrtPS (non-real-time Polling Service) :
  - *Application* : les flots nrtPS reçoivent le même service que les rtPS.
  - *Méthode d'accès* : est la même que pour le service rtPS. De plus, durant les congestions, les requêtes de contention sont autorisées.
- Le service best effort (BE) :
  - *Application* : le service minimal. Il est utilisé pour les flux ne nécessitant pas de qualité de service particulière.

- *Méthode d'accès* : La SS est obligée de rentrer en mode de contention pour transmettre des requêtes de bande passante.

1.2.2.5. Conclusion

Longtemps freinée par les réglementations de fréquence, la technologie réseau 802.16 peut enfin offrir une extension conséquente à la boucle locale radio : avec une portée de 50Km, elle permet de couvrir de larges zones géographiques reculées. Conçue avec des mécanismes de différentiation de trafic, elle offre également un support QdS adéquat aux applications interactives exigeantes.

1.2.3. Technologie DVB-S2 / RCS (Satellite)

1.2.3.1. Présentation

Bien que les réseaux satellite géostationnaires aient longtemps été dédiés aux services de diffusion, les réseaux par satellites suscitent aujourd'hui l'intérêt des chercheurs et des industriels. La démocratisation des terminaux satellites DVB-S (Digital Video Broadcast for Satellite) et l'évolution des transmissions et techniques de codage en ont fait une technologie complémentaire aux infrastructures terrestres.

En 1999, la norme DVB-RCS (Digital Video Broadcast Return Channel over Satellite) standardise une voie retour via satellite pour les terminaux satellites (utilisateur terminal vers satellite). La capacité de la voie retour (DVB-RCS) varie selon le nombre de transpondeurs utilisés, sachant qu'un transpondeur peut gérer jusqu'à 32Mbit/s. Couplée avec la voie aller s'appuyant sur la norme DVB-S2, les réseaux satellite DVB-S2/RCS introduisent l'interactivité nécessaire aux services IP multimédia large bande dans les zones géographiques non couvertes ou à couverture difficile.

Cette évolution a donné lieu à de nouvelles topologies réseau telles que les réseaux d'accès Internet par satellite, illustrés par la Figure 4 :

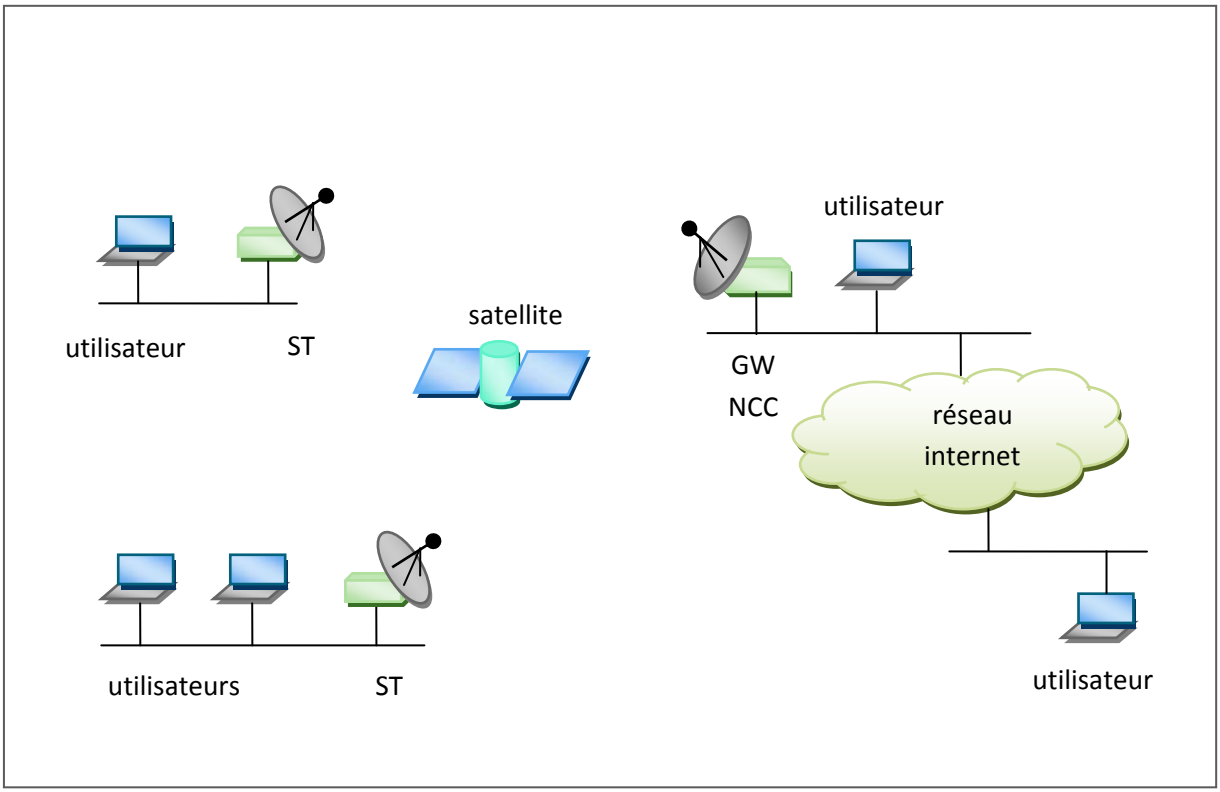


Figure 4 : système satellite DVB-S2/RCS

Les ST (Satellite Terminal) se comportent comme des routeurs d'accès à la voie retour pour le trafic utilisateur. Le GW (Gateway) centralise l'ensemble du trafic dans le réseau satellite et établit l'interconnexion avec les réseaux terrestres. Une entité, le NCC (Network Control Center), est chargée de la gestion des ressources du réseau satellite. Cette fonction est généralement effectuée au sol et est couplée au GW.

Deux types de satellites sont proposés :

- Les satellites classiques, dits « transparents », qui propagent le signal montant sur la voie descendante sans aucun traitement additionnel. Sachant que les terminaux satellites ne sont pas aptes à recevoir un signal DVB-RCS, une conversion DVB-RCS vers DVB-S2 est effectuée par une passerelle au sol. Ainsi, toute communication de ST à ST passe forcement par la passerelle ce qui donne lieu à un « double bond » et une topologie communément appelée en « étoile ».
- Les satellites dotés de capacités de traitement à bord, dits satellites « régénératifs », sont capables d'effectuer cette conversion (DVB-S2 vers DVB-RCS, et inversement) et offrent ainsi la capacité de communication de ST à ST en un seul bond. Ces topologies sont dites « maillées ». Des mécanismes de commutation multi-spots peuvent être ajoutés à ce type d'architectures.

Dans le cadre de nos travaux de thèse, nous nous plaçons dans une topologie réseau d'accès Internet. Dans cette configuration, il existe généralement un unique GW qui représente le seul point d'entrée du domaine satellitaire. De ce fait, la méthode d'accès et la gestion de la QdS sont complètement différentes entre la voie aller et la voie retour du système satellite.

Sur la voie aller, le GW centralise l'ensemble des données et la signalisation, occupant ainsi l'ensemble de la largeur de bande offerte par le transpondeur sur la voie aller. La méthode d'accès s'appuie sur un multiplexage par répartition fréquentielle ou temporelle TDM (Time Division Multiplexing). Par la singularité de la gestion de la QdS sur la voie aller, nous nous intéresserons plus particulièrement aux fonctionnalités disponibles sur la voie retour.

Sur cette voie retour, le point d'entrée est distribué entre autant de STs qui voient leur accès aux ressources contrôlé par l'algorithme DAMA (Demand Assignment Multiple Access).

Comme tout protocole de liaison de données utilisant un médium à diffusion, la norme DVB-RCS propose un mécanisme de résolution des contentions d'accès entre les différents terminaux satellite qui tentent d'accéder simultanément au lien retour du satellite. La méthode utilisée est de type MF-TDMA (Multi Frequency Time Division Multiple Access) et se base sur une division fréquentielle correspondant aux fréquences des différentes porteuses. Chacune des porteuses est divisée en trames (appelée « supertrame ») puis en « unités temporelles » de durée fixe que l'on appellera timeslots par la suite. Les STs utilisent ces timeslots afin de transmettre leurs données au format MPEG2-TS ou ATM.

1.2.3.2. Mécanisme de bande passante à la demande

Afin d'allouer la bande passante de manière dynamique et donc plus efficacement, des techniques d'allocation de bande passante à la demande (BoD : Bandwidth on Demand) sont introduites par l'institut ETSI-BSM (European Telecommunications Standards Institute, Broadband Satellite Multimedia) dans le standard DVB-RCS. Le protocole DAMA, sous un mode centralisé (client/serveur) permet aux STs de requérir régulièrement auprès du NCC de la « capacité d'émission », c'est à dire des réservations de timeslots pendant lesquels ils pourront émettre sur la voie retour sans contention possible. Ces demandes de réservation de capacité sont calculées par le client DAMA installé au sein du ST selon l'état de ses files d'émission et transmises par des requêtes de capacité (CR) vers le serveur DAMA du NCC (c.f. Figure 5).

Le standard définit quatre types de requêtes de capacité pour satisfaire les besoins des applications :

- des requêtes de bande passante fixe CRA (Continuous Rate Assignement) négociées à l'initialisation du ST et allouées statiquement pour toute la durée de connexion du ST;
- des requêtes de bande passante variable RBDC (Rate Based Dynamic Capacity) négociées pour une supertrame ;
- des requêtes de transmission d'un volume de données VBDC (Volume Based Dynamic Capacity) cumulables sur plusieurs supertrames ;
- des requêtes de transmission d'un volume de données AVBDC (Absolute Volume Based Dynamic Capacity et Absolute Volume Based Capacity) : idem que le VBDC sauf qu'une nouvelle requête annule la précédente.

A noter qu'une allocation « bonus » FCA (Free Capacity Assignement) offerte par le NCC est redistribuée équitablement entre les ST lorsqu'il reste de la bande passante non utilisée.

Afin de transporter les requêtes de capacité vers le serveur DAMA du NCC, les ST disposent de deux types de signalisation :

1. une signalisation dans la bande : les requêtes sont encapsulées et émises dans des paquets de données classiques en utilisant la méthode DULM (Data Unit Labeling Method). Cette méthode permet également d'envoyer des informations de contrôle et administratives au NCC ;
2. une signalisation hors bande : des timeslots avec contention sont spécialement dédiés en début de chaque supertrame pour transmettre périodiquement les requêtes.

Par le biais d'une table de signalisation (TBTP) émise périodiquement sur la voie aller, le NCC communique aux ST l'affectation des ressources de la voie retour pour la prochaine supertrame.

Comme le montre la Figure 5, une latence minimum incompressible appelée MSL (Minimum Scheduling Latency) est imposée à l'allocation dynamique. Cette latence représente au moins un temps aller-retour dans le réseau satellite, plus les temps de calcul au sein du ST (CR) et du NCC (TBTP), soit au minimum 500ms.

Le standard définit également des classes de trafic au niveau MAC qui peuvent être implémentées au sein du ST par des files d'émission MAC distinctes. Ces classes sont au nombre de quatre :

- la classe RT (Real Time) pour les applications à fortes contraintes temporelles, utilise les requêtes de capacité CRA pour les besoins en ressources ;
- la classe VR-RT (Variable Rate) dédiée au trafic sensible au débit variable, utilise les requêtes RBDC ;
- la classe VR-JT dédiée au trafic tolérant à la gigue, utilise les requêtes VBDC ou AVBDC ;
- et JT (Jitter Tolerant) pour le reste du trafic, utilise l'allocation FCA.

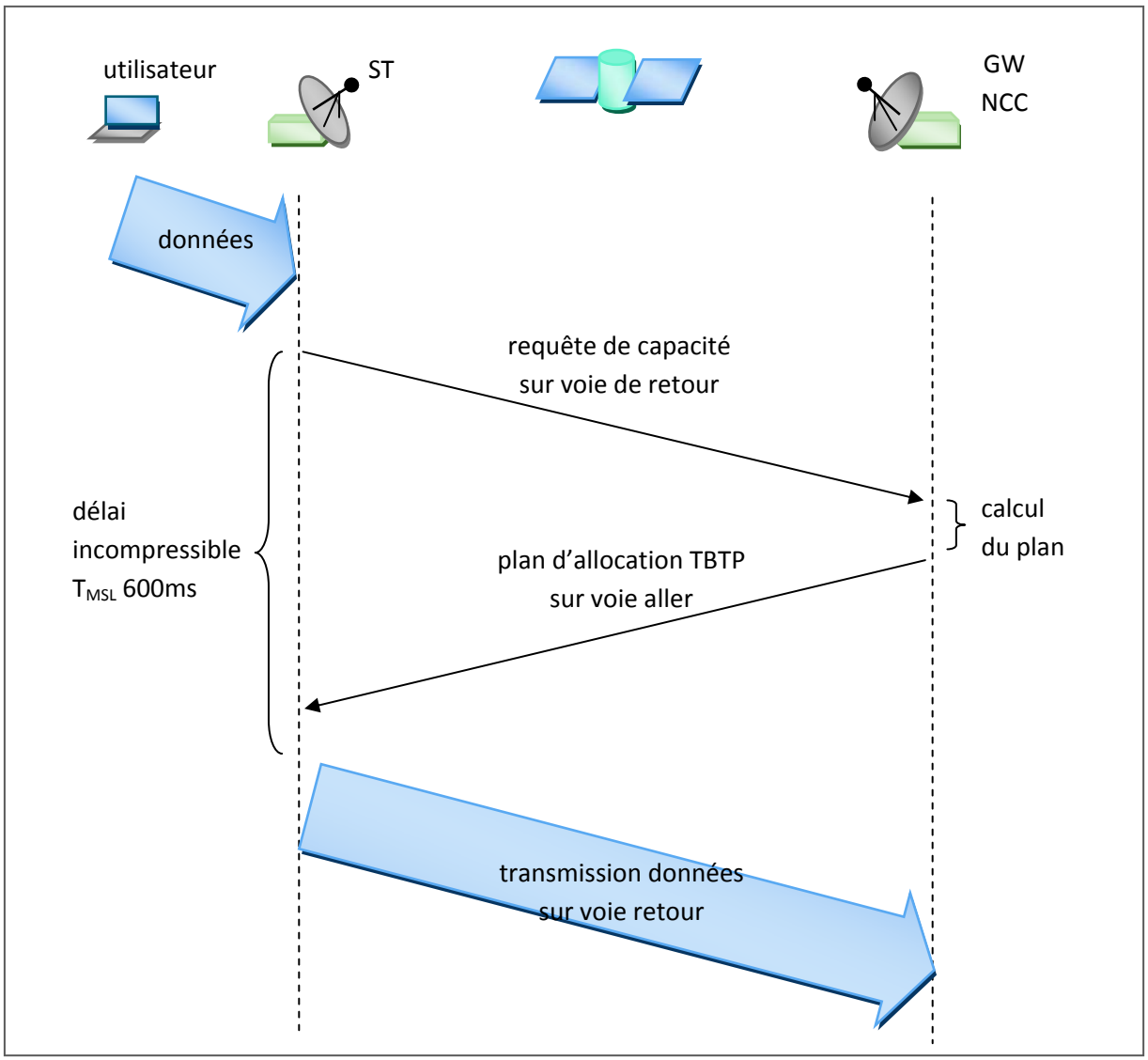


Figure 5 : cycle requête/allocation du DAMA

1.2.3.3. Conclusion

La grande portée de la technologie satellite DVB-S2/RCS offre un tremplin considérable pour l'accès Internet dans les zones géographiques inaccessibles à la technologie xDSL. En effet, trois satellites géostationnaires permettent une couverture totale de la planète. Avec des débits avoisinant les 50Mbit/s pour la voie retour par satellite, elle reste, tout comme le 802.11 et 802.16, une technologie à ressources limitée. Des mécanismes de priorités sont proposés afin d'assurer une meilleure gestion des ressources pour les applications interactives. Cependant, le délai de propagation incompressible de 500ms rend une allocation dynamique de ressources difficile.

1.3. Couche Réseau

La couche réseau est responsable de la livraison de bout-en-bout (de la machine initiale vers la destination finale) de paquets. L'implémentation la plus déployée à l'heure actuelle est le réseau Internet, avec son protocole IP. Nous proposons une brève présentation de l'évolution des services proposés par le réseau Internet.

1.3.1. Introduction : évolution des services Internet

Le réseau Internet fût initialement conçu à des fins militaires par Paul Baran, en 1964. Le besoin était de créer un réseau de communication militaire capable de résister à une attaque nucléaire.

En 1969, indépendamment de tout objectif militaire, l'ARPA (Advanced Research Projects Agency) créa le réseau expérimental ARPANET afin de répondre au besoin de connectivité. Il permit ainsi de relier un certain nombre d'instituts universitaires entre eux.

A partir de 1971, Ray Tomlinson et Lawrence G. Roberts orientent l'utilisation d'Internet principalement vers la communication par courrier électronique.

Un besoin d'acheminer des données sur le réseau ARPANET en les fragmentant en petits paquets conduit Bob Kahn en 1972 et Vinton Cerf en 1973 à élaborer un protocole de communication NCP (Network Control Program). Ce dernier devint en 1976 TCP afin d'étendre le protocole aux autres réseaux.

Dès 1980, Tim Berners-Lee répondit au besoin de consulter des documents en ligne, grâce à un système de navigation hypertexte à travers les réseaux. Ce système donna naissance, fin 1990, au protocole HTTP (Hyper Text Tranfer Protocol), ainsi qu'au langage HTML (HyperText Markup Language). Le World Wide Web est né.

Le réseau Internet, depuis sa création, n'a cessé de croitre et d'évoluer, en nombre de machines, en nombre et en variété d'applications, et dans la capacité de l'infrastructure réseau. Cette croissance est toujours aussi importante et d'actualité.

En matière de qualité de service, le réseau Internet actuel fournit un seul type de service, celui du « best effort » : le réseau tente de véhiculer les paquets de l'émetteur jusqu'au destinataire. Cependant tous les paquets transitant sur le réseau sont traités de la même manière : aucune ressource spéciale ni priorité particulière de bout en bout n'est accordée au paquet.

La simplicité de ce service lui permet d'être déployé à grande échelle sur le réseau. Cependant, le service best effort dépend directement de la stabilité des routes, de la charge et de la disponibilité du réseau. Des paquets peuvent être perdus, dupliqués, ou corrompus durant la transmission sur le réseau. Il n'y a donc aucune garantie sur la livraison à destination des paquets.

Afin d'avoir un service d'acheminement des paquets avec des performances satisfaisantes, les opérateurs réseau sur-dimensionnent les cœurs de réseau.

Cependant, certaines technologies réseau évoluent beaucoup plus lentement. Il s'agit plus particulièrement des technologies récentes de réseau sans fil telles que les réseaux par satellite, présentés dans la section 1.2.3, où le surdimensionnement est impossible dû au spectre limité. Pour faire face aux limitations des technologies réseau sans fil, la notion de Qualité de Service est

introduite à différents niveaux. Au niveau IP, de nombreuses propositions ont été faites, offrant de nouveaux services afin de prioriser les flux critiques. Nous allons donc présenter deux architectures majeures de QdS IP : DiffServ et IntServ.

1.3.2. Integrated Services : IntServ

Les besoins et les mécanismes pour les services intégrés ont été le sujet de nombreuses discussions et travaux de recherche durant les années passées, dont voici quelques références qui sont bien loin d'être une liste exhaustive [9] [10] [11] [12] [13] [14]. Ces différents travaux ont mené à une approche unifiée du support Integrated Services, RFC 1633 [15]. L'objectif étant de fournir un minimum de garantie sur le délai de bout en bout et le débit de transmission des paquets de flux applicatifs dans le réseau Internet. Pour cela, il propose de conserver l'état des flux concernés dans les routeurs, ce qui représente un important et fondamental changement dans l'architecture Internet. En effet, dans la conception originale de l'architecture Internet, tous les états relatifs aux flux doivent être conservés dans les systèmes terminaux [16].

1.3.2.1. Classes de Service IntServ

Le groupe IntServ propose d'offrir des garanties de QdS par flux. Pour cela il définit deux nouveaux services en plus du Best Effort : Guaranteed et Controlled Load services.

- Le service Guaranteed est exprimable de façon quantitative en termes de bande passante et de délai de transit maximal : il garantit que tous les paquets d'un même flux arriveront en un temps borné défini par l'application.
- Le service Controlled Load est un service de bout en bout, exprimable de façon qualitative : il assure que la transmission se fera comme sur un réseau peu chargé (pas de congestion).

1.3.2.2. Composants et mécanismes QdS IntServ

Afin de formuler ses besoins en ressources, l'application spécifie la QdS désirée en utilisant un certain nombre de paramètres (flowspec). Cette liste de paramètres est acheminée par un protocole d'établissement et de réservation de ressources RSVP (Resource reSerVation Protocol, c.f. Figure 6). Ce protocole permet de lancer la réservation des ressources réseau nécessaires à l'obtention du service auprès des routeurs du réseau. Chaque routeur, par le mécanisme de contrôle d'admission, est en charge d'accepter ou non la demande de réservation en tenant compte des ressources disponibles localement et de la caractérisation du trafic fournie avec la réservation. En cas de réponse positive, la réservation s'effectue par la création et le maintien d'un état spécifique au flux chez le client et dans les routeurs se trouvant le long du chemin du flux. La liste de paramètres est ensuite utilisée pour paramétrer les mécanismes de classification et d'ordonnancement de paquets dans les routeurs.

Lorsque le flux arrive dans les routeurs, un mécanisme de classification permet de l'identifier par le quadruplet {adresse IP source, destination, numéro de port source, destination}, puis de le diriger vers la file implémentant le service requis. Un autre mécanisme d'ordonnancement de paquets réordonne les paquets dans les files par priorité.

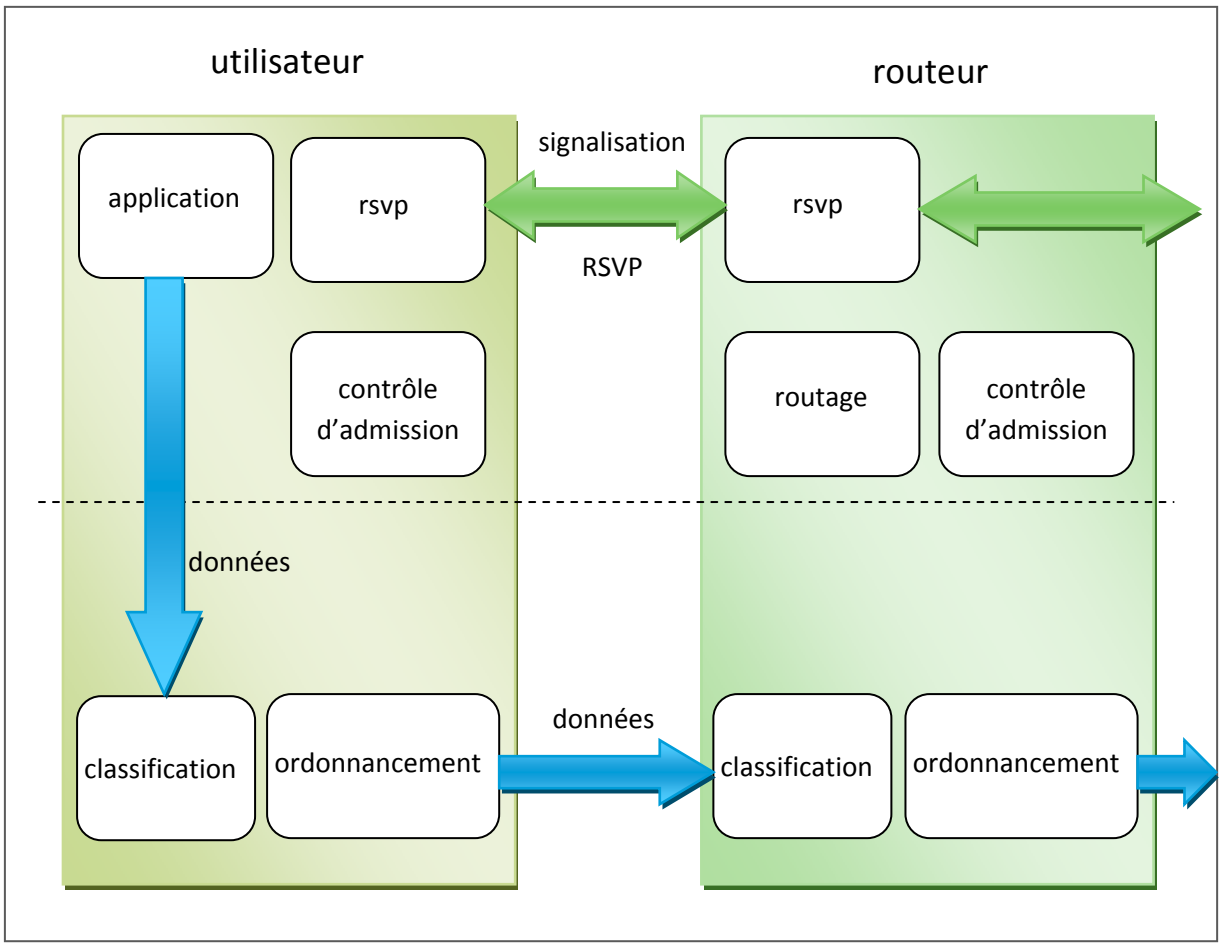


Figure 6 : architecture IntServ avec signalisation RSVP

1.3.2.3. Conclusion

L'architecture IntServ définit un cadre pour la gestion de la QdS de bout en bout la plus fine qui soit puisqu'elle garantit des niveaux de QdS non prédéfinis et contrôlés qui correspondent exactement aux besoins d'un flux applicatif. L'utilisation d'un protocole unique de signalisation offre un cadre de référence homogène d'un domaine à un autre. Cependant, l'inconvénient majeur du modèle IntServ reste sa difficulté de passage à l'échelle. En effet, l'architecture se révèle totalement inadéquate en cœur de réseau où la concentration de trafic est telle que la charge due à la signalisation induite et le nombre d'états dans le plan de contrôle et de donnée à gérer au sein des routeurs explose.

1.3.3. Differentiated Services : DiffServ

Dans le paragraphe précédent, nous avons vu qu'une gestion de QdS flux par flux telle que proposée par IntServ est difficilement applicable dans le cas de grands réseaux. Le groupe IETF Diffserv propose donc d'abandonner globalement sur l'Internet le traitement du trafic par flux (IntServ) pour le caractériser sous forme de classes en proposant le modèle Differentiated Services [17]. La différenciation de services consiste à agréger par classe le trafic arrivant sur le réseau selon les besoins en QdS. La granularité est donc moins fine. La complexité des routeurs ne dépend plus du nombre de flux mais du nombre de classes de service implémentées. Il n'est plus nécessaire de maintenir des états dans les routeurs pour chacun des flux. De plus, la complexité est reportée sur

les routeurs aux frontières du réseau, allégeant ainsi les tâches des routeurs de cœur. Cette agrégation de flux par classe facilite grandement le passage à l'échelle de l'architecture.

1.3.3.1. Architecture DiffServ

Le modèle DiffServ propose une division du réseau Internet en domaines DiffServ. La Figure 7 représente un ensemble de routeurs de bordure (edge router) et de routeurs de cœur du réseau (core router) DiffServ.

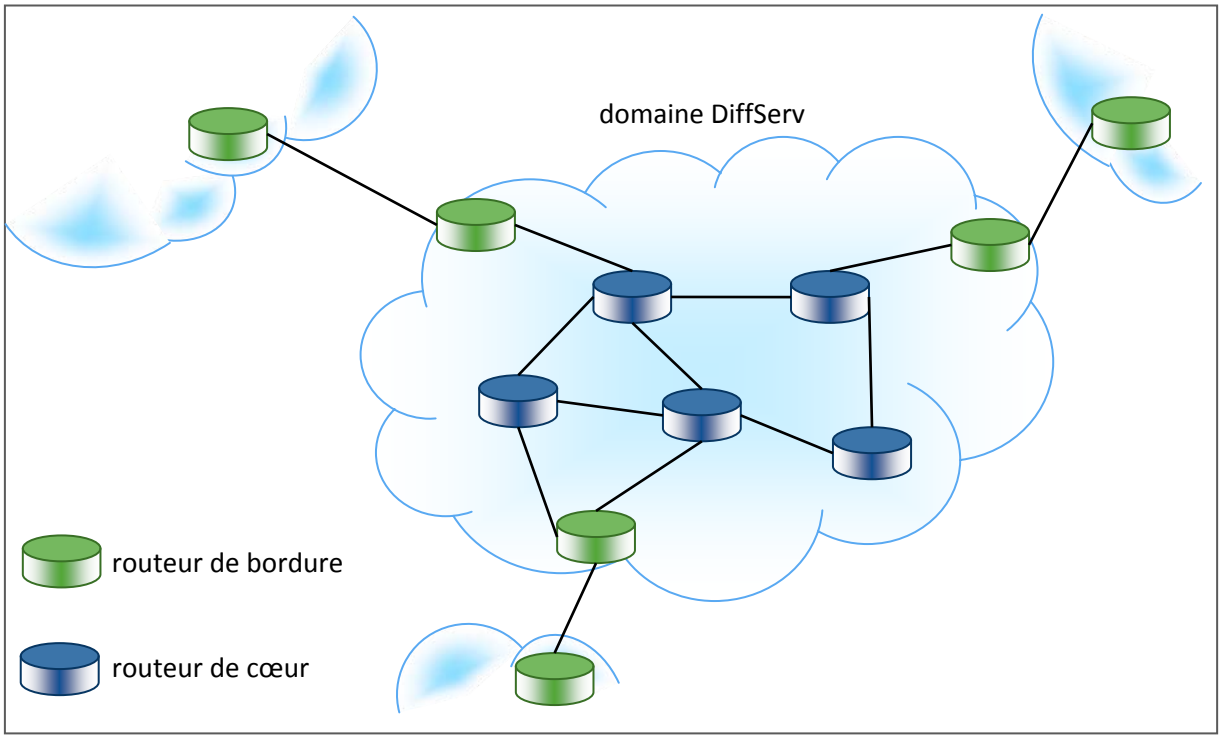


Figure 7 : un domaine DiffServ

Les routeurs de bordure représentent les premiers éléments actifs d'un domaine DiffServ face à l'arrivée du trafic. Ils permettent d'interconnecter le domaine DiffServ à d'autres domaines DiffServ ou non-DiffServ. Ils effectuent deux opérations fondamentales sur les paquets : la classification des flux en classes de services et leur conditionnement. Tout comme dans le modèle IntServ, la classification s'effectue à partir du quadruplet. Des mécanismes de profilage et de mesure du trafic permettent la mise en forme des flux afin qu'ils correspondent au profil de trafic contracté (SLA : Service Level Agreement). Les paquets ne respectant pas le profil, sont gérés par des mécanismes qui soit les mettent en tampon, soit leur affectent une priorité inférieure, soit les jettent. Puis les paquets sont dirigés vers la fonction de marquage qui fixe la valeur du champ DSCP (Differentiated Service Code Point) indiquant la classe de trafic auquel appartient le flux. Les champs de précédences des en-têtes IP sont utilisés à cet effet : en IPv4, c'est le champ ToS (Type of Service) et en IPv6 c'est le champ CoS (Class of Service).

Puis les paquets sont envoyés à leur destination. À chaque routeur de cœur sur le chemin, les flux sont classifiés par leur marquage, dirigés vers une file de transmission implémentant le service afin d'y recevoir le traitement PHB (Per-Hop-Behavior) associé. Diffserv définit trois PHB :

- Expedited Forwarding (EF) [18] : correspond à la priorité maximale. Il assure le transfert du flux en garantissant une bande passante avec des taux de perte, de délai et de gigue faible ;

- Assured Forwarding (AF) [19] qui comprend plusieurs sous classes : garantit l'acheminement de certains paquets en cas de congestion.
- Class Selector (CS) : est utilisée pour maintenir la compatibilité avec les réseaux utilisant les champs de précédences IP (ToS).

Les priorités sont assurées par les algorithmes d'ordonnancement tels que PQ (Priority Queueing) ou encore WRR (Weighted Round Robin) servant à contrôler la distribution de ressources entre les classes de trafic.

1.3.3.2. Conclusion

L'architecture DiffServ définit un cadre pour le support de Qualité de Service dans les réseaux IP. Le traitement des flux par classes facilite son passage à l'échelle. Des contrats de service SLA (Service Level Agreement) sont établis entre client et fournisseur de service. Ils spécifient le service que le client recevra pour ses flux. Ce principe simplifie la tarification du service, cependant la négociation et la mise à jour de ces contrats restent des procédures relativement statiques et lourdes qui s'effectuent manuellement.

1.3.4. Conclusion

Bon nombre de solutions IP proposent des mécanismes de QdS, que cela soit de bout en bout ou encore de bordure en bordure de réseau. Nous avons abordé deux des principales solutions IP actuelles, IntServ et DiffServ, qui sont complémentaires de part leurs différentes fonctionnalités. L'architecture IntServ propose aux utilisateurs des services personnalisables à l'aide du protocole de signalisation et réservation de QdS RSVP. Il est utilisable en bordure de réseau. Alors que l'architecture DiffServ utilise des classes prédéfinies de service et peut par conséquent être plus facile à déployer à large échelle. Cependant le passage à l'échelle complet du réseau Internet reste toujours problématique. Un tel déploiement nécessiterait des modifications majeures pour lesquelles les opérateurs réseau ne sont pas encore prêts.

1.4. Couche Transport

La couche transport est chargée de l'encapsulation des données applications en paquets (datagrammes) adaptés pour la transmission jusqu'au destinataire sur les infrastructures réseau. De nombreux protocoles de transport sont proposés dans l'architecture Internet. Dans ce chapitre, nous allons donc aborder quatre des principaux protocoles de transport actuellement utilisés sur le réseau Internet. L'objectif est, à partir de leurs principales caractéristiques, de faire ressortir la Qualité de Service qui est offerte par la couche Transport à travers ces protocoles.

1.4.1. TCP

Le protocole TCP (Transmission Control Protocol) [20] est un protocole au service fiable, orienté connexion. Ses fonctionnalités principales sont les suivantes :

Orienté connexion : une poignée de main à trois voies est utilisée pour l'établissement et la terminaison de la connexion. La connexion est alors identifiée par le quadruplet : adresse IP source, destination, numéro de port source, destination.

Fiabilité : il utilise des mécanismes d'acquittement, de retransmission de paquets ainsi que des timers afin délivrer de bout en bout les données avec la plus haute fiabilité.

Ordre : il assure que les paquets arrivent dans le même ordre que celui d'émission en utilisant à la fois des numéros de séquence de paquets et des tampons de réception.

Contrôle de congestion : des mécanismes de contrôle de congestion (Slow-Start, Congestion Avoidance, Fast Retransmit, Fast Recovery [21]) permettent d'éviter l'engorgement du réseau lorsque ce dernier est saturé et d'augmenter le débit de transmission lorsque le réseau est faiblement chargé.

Contrôle de flux : TCP fournit un mécanisme de fenêtre d'émission glissante dynamique pour contrôler le débit de transmission de l'émetteur. Ce dernier ne peut émettre plus rapidement que le récepteur l'autorise. Ce mécanisme est également utilisé pour le contrôle de congestion.

Transfert de flux d'octets de données : TCP découpe les flux d'octets de données de l'application en segments dont la taille est déterminée par le mécanisme de fenêtre glissante.

Contrôle d'erreur sur bit : TCP intègre un mécanisme de checksum sur 16 bits pour couvrir les erreurs sur bit de l'entête et des données.

1.4.2. UDP

Le protocole UDP (User Datagram Protocol) [22] est un protocole de transport de datagrammes, au service non fiable, en mode non connecté. Ses fonctionnalités sont les suivantes :

Non fiable : UDP est un protocole transactionnel, aucun mécanisme n'est implémenté afin de garantir la délivrance du message au destinataire, ni son éventuelle duplication.

Non orienté connexion : Il n'y a pas de notion de connexion (ni établissement, ni terminaison). Des paquets peuvent être envoyés et reçus à tout instant par les applications. Aucun état n'est conservé sur les flux.

Non ordonné : il ne garantie pas l'ordre d'arrivée des paquets à la destination.

Sans contrôle de congestion : aucun mécanisme de contrôle de congestion n'est implémenté.

Transfert de flux de datagrammes : exactement un datagramme UDP est généré pour chaque opération de transmission effectuée par l'application, entraînant ainsi l'émission d'un datagramme IP.

Contrôle d'erreur sur bit : UDP intègre un mécanisme de checksum sur 16 bits pour couvrir les erreurs sur bit de l'entête et des données.

1.3.3. UDP-Lite

Le protocole UDP-Lite [23] est un protocole de transport de datagrammes, au service non fiable, en mode déconnecté. Il est donc très similaire à UDP. Cependant, il offre une fonctionnalité de plus : la couverture partielle du « ***checksum*** ». TCP comme UDP utilise un mécanisme de checksum leur permettant de détecter les erreurs sur bit à la fois sur l'en-tête et les données des paquets qui pourraient survenir durant la transmission. En cas de présence d'erreur sur bit, le paquet est automatiquement supprimé.

Dans le cadre des communications d'applications multimédia, certaines gammes de codage audio et vidéo tolèrent les erreurs de bit. Il s'agit par exemple du codec de voix AMR [24] [25], le codec à faible débit (Internet Low Bit Rate Codec), et le codec résistant aux erreurs H.263+ [26], H.264 [27], et les codecs vidéo MPEG-4 [28].

UDP-Lite offre la possibilité à l'utilisateur de spécifier la partie de données (en nombre d'octets) qui sera soumise à la vérification du checksum. La présence d'erreur sur bit dans la partie des données non couverte par le checksum ne déclenchera donc pas la suppression du paquet par le protocole de transport.

Ces erreurs, du point de vue de l'utilisateur final, sont perçues comme une légère perturbation dans le media (audio/video) qui lui est présenté. Alors qu'un datagramme entièrement manquant (perdu ou jeté pour cause d'erreur sur bit) introduit une interruption ou une distorsion flagrante dans le flux multimédia présenté à l'utilisateur final.

Ce service de **fiabilité partielle** offert par le protocole UDP-Lite convient parfaitement à ce type d'application.

1.3.4. DCCP

Le protocole DCCP (Datagram Congestion Control Protocol) [29] est un protocole de transport de datagrammes, et fournit un service de transmission non fiable, orienté connexion. Il se trouve à mi chemin entre TCP et UDP, car il possède des fonctionnalités communes à ces deux protocoles :

Orienté connexion : une poignée de main à trois voies fiable est utilisée pour l'établissement et la terminaison de la connexion, afin de configurer certains paramètres de la communication.

Non fiable : DCCP ne fournit aucune garantie sur la livraison des données au destinataire.

Non ordonné : DCCP ne garantit pas l'ordre d'arrivée des paquets à la destination.

Contrôle de congestion : divers algorithmes de contrôle de congestion, configurables et adaptés à différents types d'applications (TCP-Like, TFRC, TFRC-SP [30]) sont proposés. Ils permettent ainsi d'éviter l'engorgement du réseau sous-jacent lorsque ce dernier est saturé.

Notification de délivrance : des mécanismes d'acquittement permettent de rendre compte de l'état des paquets, et ceci de façon précise (paquet supprimé sur le chemin, paquet corrompu, débordement de tampon à la réception, etc.).

Transfert de datagrammes de données : exactement un datagramme DCCP est généré pour chaque opération de transmission effectuée par l'application, entraînant ainsi l'émission d'un datagramme IP.

Plusieurs paramètres d'une connexion DCCP peuvent être configurés, tels que le choix de l'algorithme de contrôle de congestion, l'utilisation de la notification explicite de congestion (Explicit Congestion Notification, ECN [31]), la fréquence d'acquittement ou encore la couverture du checksum.

Un contrôle de congestion G-TFRC [32] adapté à l'architecture DiffServ, est en cours de spécification. Pour un flux multimédia admis dans la classe AF avec un débit minimum garanti, le contrôle de congestion G-TFRC permet au flux d'atteindre rapidement le débit minimum autorisé, et ceci quelque soit la valeur du RTT, tout en garantissant le partage équitable de la bande passante disponible avec les autres flux.

1.4.3. Conclusion

La couche transport aujourd'hui possède un grand nombre de protocoles, chacun proposant des services divers et variés.

- Le protocole de transport TCP est capable de répondre aux exigences de fiabilité et d'ordre requises par les applications de données, telles que la navigation web, le transfert de fichiers, le courrier électronique, etc.
- UDP est un protocole de transport minimaliste. Sa simplicité est requise par deux classes d'applications :
    - celles qui effectuent de nombreux échanges requête/réponse de petite taille. Ces applications privilégient la simplicité de délivrance des paquets. On peut citer les applications DNS (Domain Name System), SNMP (Simple Network Management Protocol), DHCP (Dynamic Host Configuration Protocol), RIP (Routing Information Protocol). Leur très faible volume en taille de paquet aura peu d'impact sur le réseau.
    - celles qui ont des contraintes temporelles et d'interactivité. Il s'agit des applications de streaming média tel que IPTV (Internet Protocol Television), VoD (Video on Demand), Voix sur IP (VoIP). Des pertes occasionnelles de paquets sont encore tolérables.

  Cependant, dans tous les cas, le protocole UDP n'intègre pas de contrôle de congestion. C'est à l'application d'implémenter son propre mécanisme afin de respecter les autres flux présents sur le réseau.

- Le protocole DCCP cherche à minimiser les délais de bout en bout des applications à forte contrainte temporelle grâce à l'absence de contrôle de fiabilité. Ses divers contrôles de

congestion s’adaptent aux besoins des applications tout en garantissant le respect des autres flux concurrents sur le réseau [33].

Nous pouvons également citer un protocole de transport de nouvelle génération orienté QdS: QoSTP (Quality of Service oriented Transport Protocol) [34]. Ce protocole fournit un ensemble de mécanismes de transport répondant précisément aux différentes exigences en QdS des applications multimédia. Ces contraintes concernent le délai, la gigue, le débit, l’ordre et fiabilité partielle. Ces exigences sont donc remplies tout en utilisant les ressources et services réseaux disponibles. En outre, QoSTP a été spécifié dans un contexte de QdS capable de fournir un espace sémantique extensible et une architecture composable. Ce principe de modélisation facilite l'extension et la spécialisation du protocole pour répondre à un grand nombre de besoins applicatifs tout en tenant compte de l'ensemble des services offerts par le système de communication. Une API (Application Programming Interface) est fournie aux applications afin qu’elles puissent communiquer leurs besoins en QdS à la couche transport, minimisant ainsi les efforts d’adaptation de la couche application.

1.5. Couche Session

1.5.1. Présentation

La couche session fournit des mécanismes pour établir, gérer et terminer des sessions entre les utilisateurs. Elle n'apparait pas comme une couche protocolaire à part entière dans le modèle Internet, cependant ses fonctionnalités sont reparties entre les couches transport et application. Le protocole TCP au niveau transport (c.f. section 1.4.1) propose certaines de ces fonctionnalités. Nous proposons ici de présenter d'autres fonctionnalités fournies par la couche application.

À l'heure de la révolution de l'information par l'utilisation du réseau Internet, on assiste à la convergence de la téléphonie et de l'informatique. L'utilisation du réseau Internet permet le support simultané de plusieurs types de média de communication pour une même session. L'environnement de communication étant bien plus complexe, il est nécessaire d'avoir à disposition des protocoles d'établissement de session d'appel.

H.323 [35] a été un protocole pionnier de la téléphonie sur IP, émanant des instances du monde des télécommunications (ITU). Il a été détrôné par SIP (Session Initiation Protocol), de conception un peu plus récente, qui vient du monde de l'Internet (IETF) et s'intègre sans doute un peu mieux sur les réseaux IP. Nous proposons dans ce chapitre de présenter ce protocole SIP qui est utilisé par les applications multimédia interactives pour établir les sessions. Puis nous verrons QoS SIP qui est un protocole basé sur SIP qui permet de déclencher la mise en place de réservation de ressources dans le réseau sous-jacent.

1.5.2. SIP (Session Initiation Protocol)

1.5.2.1. Présentation

SIP est un protocole normalisé et standardisé par l'IETF, décrit dans [36]. Il a été conçu pour établir, modifier et terminer des sessions multimédia. Il se charge de l'authentification et de la localisation des différents participants. Il se charge également de la négociation des types de média utilisables par les différents participants. SIP n'est donc pas seulement destiné à la voix sur IP mais aussi à de nombreuses autres applications telles que la visiophonie, la messagerie instantanée, la réalité virtuelle ou même les jeux vidéo. Il est indépendant du transport des données utiles, ainsi tout type de protocoles de transport peut être utilisé pour la transmission.

Cependant le protocole RTP (Real-time Transport Protocol) [37] est couramment utilisé pour le transport des données de session audio et vidéo. Il gère la synchronisation des média lors de leur présentation à l'utilisateur. Les paquets RTP sont encapsulés dans des paquets UDP.

1.5.2.2. Fonctionnement

Nous proposons de présenter le fonctionnement de SIP à travers un scénario typique.

Les User Agents, plus communément appelé clients SIP, représentent les entités de communication que l'on retrouve dans les téléphones SIP, les soft-phones (logiciels de téléphonie sur IP) des ordinateurs et PDA (Personal Digital Assistant).

Enregistrement du client SIP auprès du réseau

Lorsque le client SIP se connecte au réseau Internet, la première étape consiste à s'enregistrer auprès d'une entité appelée Registrar (c.f.Figure 8).

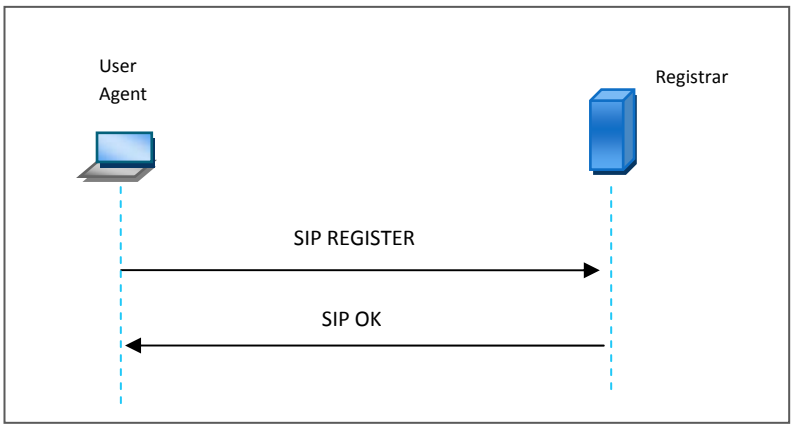


Figure 8 : enregistrement d'un Client SIP

Le client SIP signale son emplacement courant sur le réseau Internet. Il lui fournit, à travers le message REGISTER, l'association URI - Adresse IP de l'utilisateur. Les URI SIP sont très similaires dans leur forme à des adresses email (ex. sip:utilisateur@domaine.com). Ainsi, quelque soit l'adresse IP de la machine sur laquelle se trouve l'utilisateur pour communiquer, il sera en mesure d'être authentifié et localisé grâce à son adresse email qui elle, est bien plus pérenne. Le Registrar stocke ces informations dans une base de données. Puis il répond au client SIP (message SIP OK) avec les informations nécessaires du domaine SIP auquel il appartient (ex. adresses IP des autres entités SIP).

Établissement de la session de communication

Un client SIP qui souhaite communiquer avec un correspondant client, requiert juste son URI (adresse email). Il n'est donc pas nécessaire qu'il ait connaissance de son emplacement actuel (adresse IP). Afin de le localiser, le client SIP entre en contact avec une autre entité SIP appelée Proxy. Le proxy SIP est également responsable de l'établissement des sessions des clients SIP du domaine (ex. laas.fr). Le scénario est décrit en Figure 9.

Le client SIP appelant envoie un message INVITE au Proxy responsable de son domaine. Ce message INVITE contient les informations nécessaires pour négocier les paramètres de la communication. Il s'agit entre autre de son URI, adresse IP, numéro de port pour la communication, l'URI du correspondant SIP, le type de communication (ex. audio, vidéo), une liste de codecs utilisables (ex. GSM, G711, H263).

Le Proxy local, à partir de l'URI du correspondant, se charge alors de retrouver l'adresse IP du Proxy responsable du domaine distant. Pour cela, il utilise les services de nom de domaine (DNS). Puis il lui transmet le message INVITE.

Le Proxy distant reçoit le message INVITE, puis il recherche dans la base de données, la localisation (adresse IP) de son client SIP appelé, pour lui transmettre le message INVITE.

Le client SIP appelé reçoit le message INVITE et négocie au moins un codec commun qui sera susceptible d'être utilisé pour la communication.

Puis il alerte l'utilisateur appelé par une sonnerie d'appel entrant.

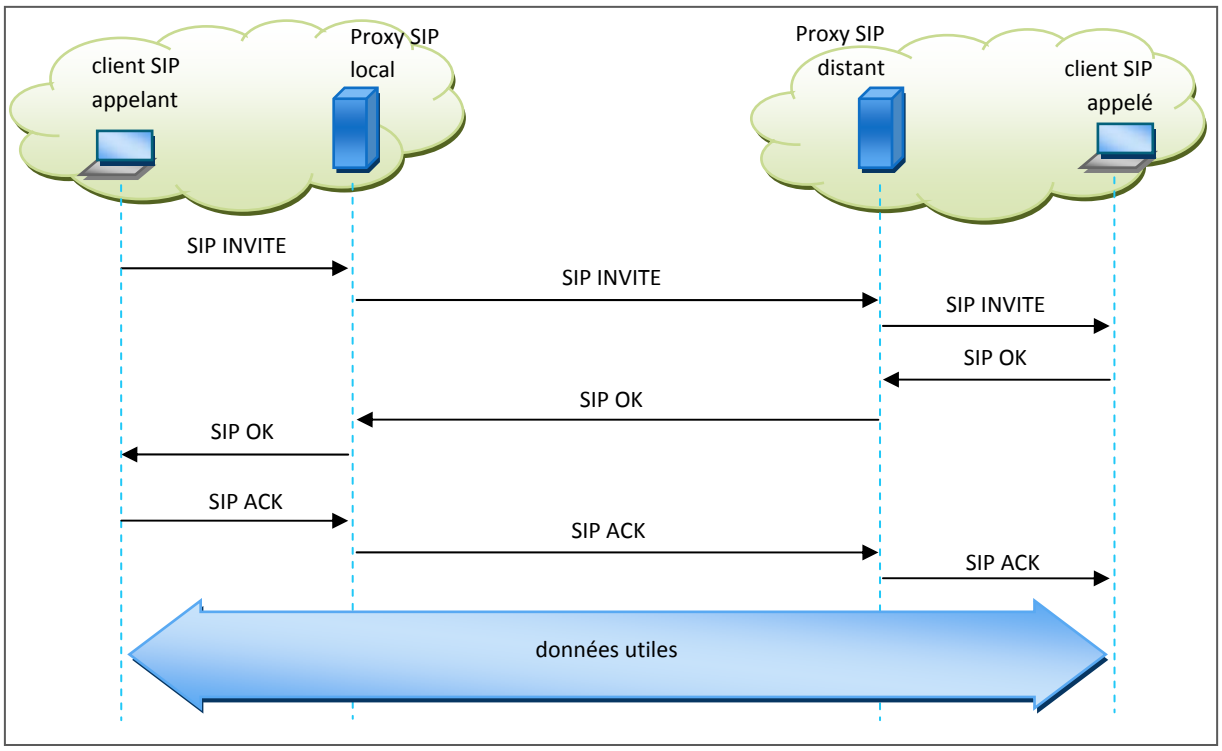


Figure 9 : établissement de session SIP

Lorsque l'utilisateur appelé accepte la communication entrante, le client SIP envoie un message OK en retour. À ce stade, le client SIP appelé a connaissance de l'adresse IP du client appelant grâce aux informations contenues dans le message INVITE reçu. Il a donc la possibilité de continuer l'établissement de la session directement avec le client appelant, sans avoir à passer par les proxys. Deux options « loose-route » et « record-route » peuvent être utilisées par les proxys afin de forcer les clients SIP à faire passer les messages d'établissement et de terminaison de session par les proxys SIP. On considère que ces options sont activées pour les travaux de ce mémoire.

Le message OK est envoyé au proxy distant. Il contient l'adresse IP du client SIP appelé, le numéro de port, et la liste de codecs qui sera finalement utilisée pour le ou les médias de la communication.

Le Proxy distant transmet le message OK au Proxy local. Le Proxy local transmet le message OK au client SIP appelant. Le client SIP appelant alerte l'utilisateur de l'établissement de la communication.

Un message ACK est transmis par le client SIP appelant au client SIP appelé, afin de confirmer l'établissement de la communication. La communication est alors établie et les données utiles peuvent transiter entre les deux clients SIP.

Modification de la session

Les clients SIP ont la possibilité de modifier certains paramètres d'une communication en cours, par exemple la proposition d'un nouveau codec pour la communication (c.f. Figure 10).

Pour cela, le client SIP désirant mettre à jour la session existante, envoie un message Re-INVITE au client SIP destinataire avec la description du nouveau media.

Le client SIP destinataire répond avec un message OK pour accepter les modifications de la session en cours.

Le client SIP initiateur confirme avec le message ACK.

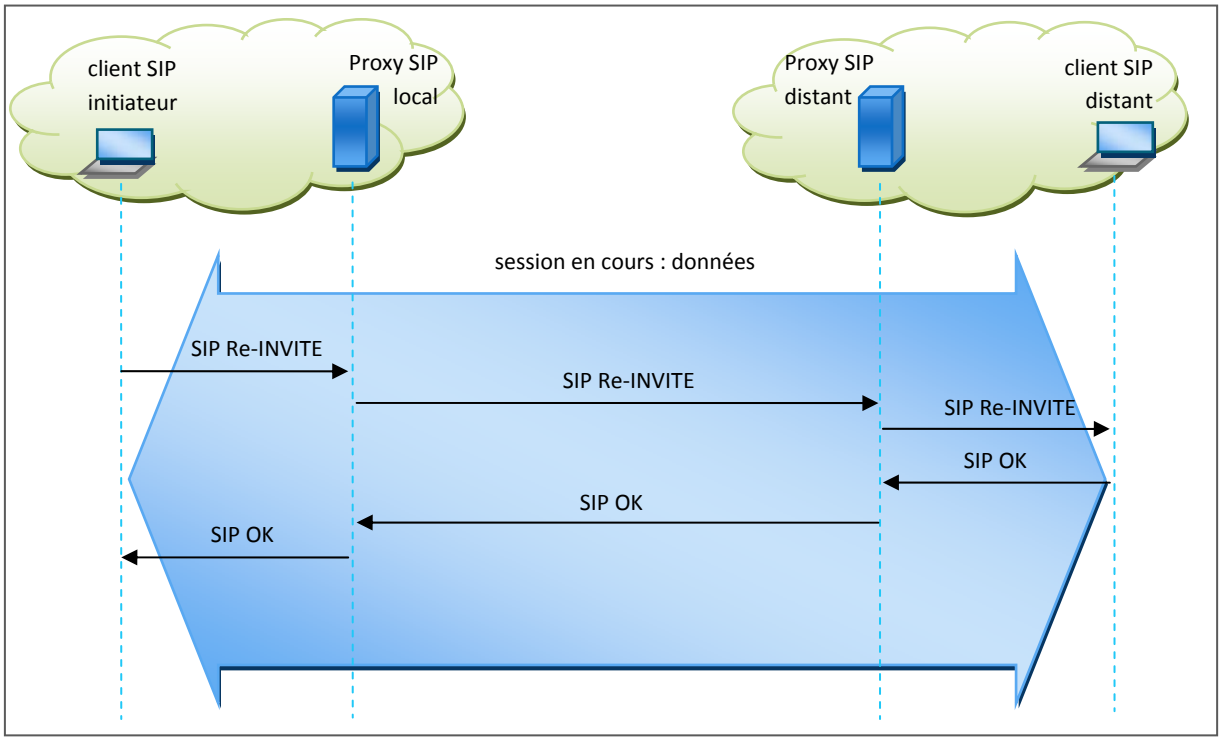


Figure 10 : modification de session SIP en cours

Terminaison de session

Lorsque l'un des clients SIP souhaite mettre fin à la session (c.f. Figure 11), il envoie un message BYE au client SIP distant. Ce dernier répond avec le message OK et la communication est ainsi terminée.

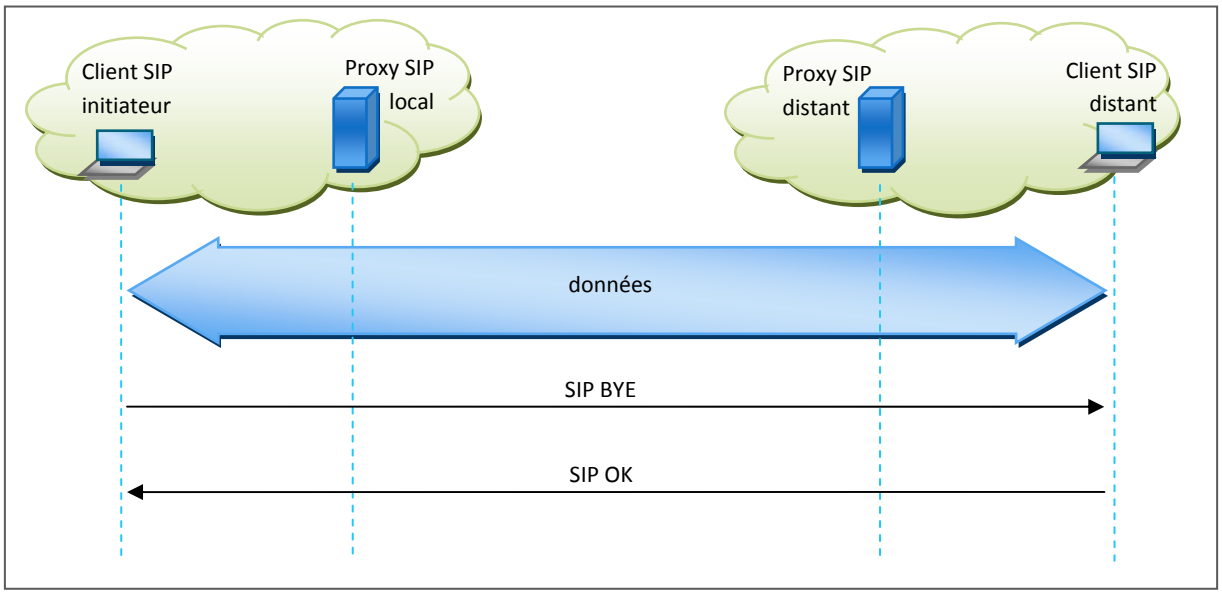


Figure 11 : terminaison de session SIP

1.5.2.3. Conclusion

Le protocole SIP offre un cadre pour la gestion des sessions multimédia (audio, vidéo, etc.) de bout en bout. Il permet la localisation des participants, la création, la modification et la terminaison des sessions. Il est également choisi pour l'établissement d'appel à travers la téléphonie sur IP (VoIP).

Cependant, aucun mécanisme n'est proposé afin de communiquer avec le réseau sous-jacent pour l'établissement d'un support QdS.

1.5.3. QoS-SIP

1.5.3.1. Présentation

Afin de proposer un service de transmission satisfaisant aux applications multimédia sur certains types de réseaux (ex. réseaux sans fil), il est nécessaire de déclencher une réservation de QdS auprès du réseau sous-jacent. Plusieurs propositions ont été faites afin d'attacher la signalisation SIP aux mécanismes IP de QdS [38], [36]. Cependant, la plupart de ces propositions considèrent que le terminal utilisateur est conscient du modèle de QdS du réseau sous-jacent. Par conséquent, c'est au terminal utilisateur d'initier la requête de réservation de QdS auprès du système de gestion de QdS du réseau. Cette approche pose un certain nombre de problèmes :

- Chaque terminal utilisateur souhaitant un support de QdS, doit être modifié afin d'intégrer les mécanismes de QdS (ex RSVP, COPS [39], etc.) utilisés dans le réseau sous-jacent. Les clients utilisant les applications sans extensions QdS sont dans l'incapacité de profiter de la Qualité de Service fournie par le réseau ;
- Les modifications à apporter au terminal utilisateur concernent à la fois les piles complètes de SIP et de réservation de QdS. Ces modifications ajoutent un degré de complexité élevé dans les terminaux utilisateur, dont la plupart possèdent des capacités mémoire et de calcul limitées.

L'objectif est donc (i) de supprimer le besoin de supporter un protocole et une implémentation QdS spécifique au sein des terminaux utilisateur, et (ii) de continuer à utiliser le protocole SIP pour l'établissement de session avec et sans support de QdS.

Afin de répondre à ce besoin, une solution a été proposée par [40] afin d'étendre le protocole SIP pour supporter la Qualité de Service. Cette solution simple d'établissement de session avec support de QdS se base sur l'extension du protocole SIP pour véhiculer les informations relatives à la QdS de bout en bout. Sa simplicité facilite son passage à l'échelle. Elle est décrite en Figure 12.

Les acteurs impliqués sont les deux clients SIP, les deux proxys SIP, et le réseau sous-jacent capable de supporter la QdS. Le choix des entités du réseau sous-jacent chargées de traiter les requêtes de réservation de QdS est laissé à l'opérateur.

Les clients SIP utilisent la signalisation SIP standard. Des extensions sont apportées à la signalisation SIP afin d'échanger les informations relatives à la QdS entre les proxys SIP. Ces extensions sont transparentes aux clients SIP. Cela permet de préserver la compatibilité avec le protocole standard SIP. Ainsi il est possible d'utiliser des clients SIP existants (legacy applications), sans avoir à leur apporter des modifications ou améliorations.

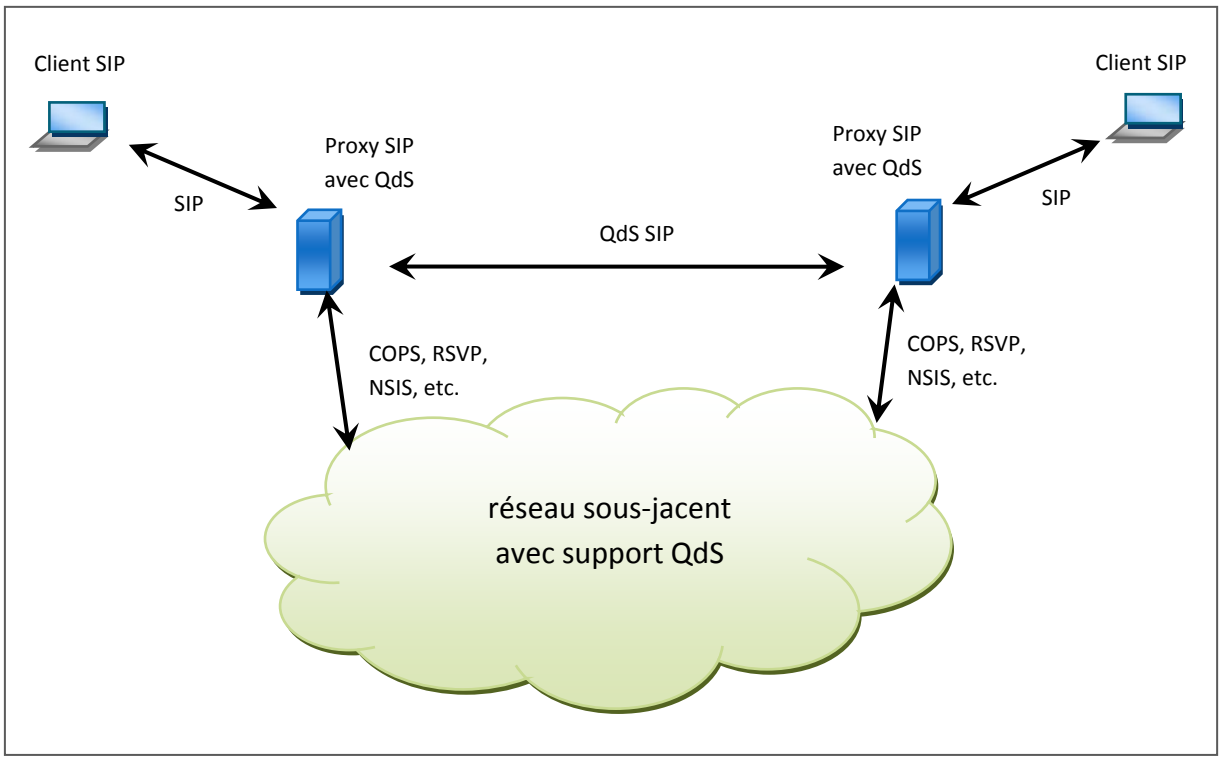


Figure 12 : architecture SIP avec support de QdS

Les fonctions et mécanismes relatifs à la Qualité de Service sont repoussés dans les proxys SIP. Les proxys SIP sont modifiés pour supporter la QdS sont appelés proxys QSIP (QoS enabled SIP). Ainsi ils contrôlent à la fois l'établissement de la session multimédia et la réservation de ressources QdS auprès du réseau sous-jacent à l'aide de signalisation QdS telle que RSVP, COPS ou encore NSIS [41]. Cette approche décharge les terminaux SIP d'une complexité inutile.

L'établissement de session QdS dans un tel scénario est composé de deux aspects : le mécanisme de signalisation de bout en bout pour l'échange des informations QdS (QdS SIP), et la négociation QdS entre les proxys QSIP et le réseau capable de QdS. L'architecture proposée découple ces deux aspects le plus possible. Ainsi, les mécanismes du protocole SIP pour échanger les informations QdS sont à la fois génériques et indépendants des mécanismes QdS du réseau sous-jacent.

1.5.3.2. Fonctionnement

Afin d'assurer le support de QdS pour les sessions multimédia, les messages SIP d'initialisation, modification et terminaison de session SIP doivent passer par les proxys QSIP des domaines concernés. Ainsi ces proxys QSIP peuvent ajouter (et lire) les informations relatives à la QdS dans les messages SIP. Si un proxy SIP basique (sans extension QdS) gère l'un des domaines des utilisateurs impliqués dans la communication, il ne comprendra pas les informations QdS insérées dans les en-têtes de messages SIP et les ignorera silencieusement. La session SIP sera établie, mais sans support de QdS.

Le principe de fonctionnement de l'architecture SIP avec support de QdS est décrit en Figure 13.

Lorsque le client SIP souhaite démarrer une session SIP, il entame la procédure standard d'établissement : il envoie le message INVITE à son proxy QSIP. Ce dernier en extrait l'URI de l'appelant ainsi que la spécification de la session (media, codecs, ports source, etc.). Le proxy QSIP se

base sur ces informations précédentes, et décide localement, s'il doit établir ou non une session QdS pour cette session. Cependant, aucune réservation de QdS ne peut être faite car, du point de vue de la signalisation SIP, la négociation des paramètres de la communication multimédia entre les clients SIP n'est pas terminée.

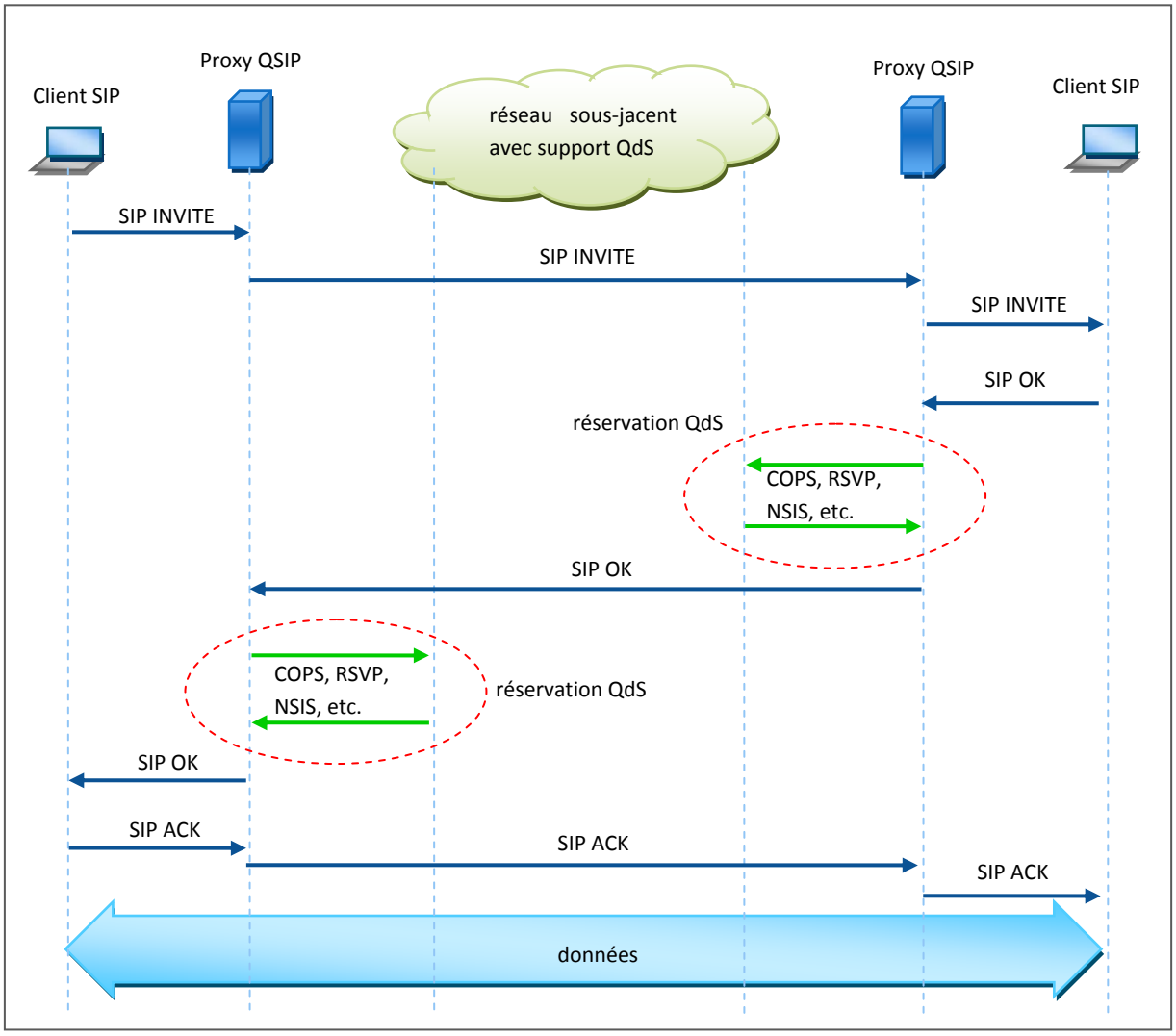


Figure 13 : fonctionnement de l'architecture SIP avec support de QdS

En cas de décision positive d'admission, un nouvel en-tête SIP (QoS-Info) est défini afin de permettre aux proxys QSIP d'échanger les informations QdS nécessaires (ex. adresses IP des éléments de QdS entrant et sortant). Le proxy insère cet en-tête dans le message SIP INVITE et le transmet au proxy distant.

Ce nouvel en-tête offre également la possibilité au proxy QSIP d'opérer en mode « stateless ». En insérant des informations d'états de QdS dans les messages SIP, les proxys conservent bien moins d'informations d'états. La variante opposée, le mode « stateful » permet au proxy QSIP intéressé de garder des informations d'états de QdS de session après établissement et de véhiculer très peu d'informations dans les messages SIP.

Quand le proxy QSIP du client appelé reçoit le message SIP INVITE qui contient les extensions de QdS, il comprend qu'une session SIP avec QdS est en cours d'établissement. Dans le cas stateless, il ajoute des informations additionnelles dans l'entête QoS-Info. Dans le cas stateful, il initialise et conserve les informations d'état de QdS. Puis il transmet le message INVITE au client SIP appelé.

Le client SIP appelé répond avec un message SIP OK et l'envoie à son proxy QSIP. À ce stade, le proxy QSIP possède toutes les informations pour effectuer une requête de réservation de ressources auprès du réseau d'accès sous-jacent (ex. routeur de bordure) pour le flot de données dans le sens appelé-vers-appelant.

En mode « stateless », lorsque le proxy QSIP reçoit une réponse positive de réservation de QdS du point d'accès QdS, il insère les informations d'état de QdS concernant cette session dans le message de réponse SIP OK, puis transmet ce message au proxy QSIP initiateur. En mode « stateful », il met à jour ses informations d'état de QdS concernant la session.

Si la réponse de réservation de QdS est négative, le proxy QSIP enregistre le status d'échec du support de QdS pour cette session. Ceci empêche que d'éventuelles retransmissions de messages SIP OK de la part du client SIP pour cette session ne déclenchent d'autres tentatives de réservation de QdS. Le proxy QSIP insère tout de même les champs additionnels nécessaires dans l'en-tête du message réponse SIP OK, afin de laisser au proxy QSIP du client appelant la possibilité d'effectuer une réservation de QdS pour la communication multimédia dans l'autre sens appelant-vers-appelé.

Quand le Proxy QSIP coté appelant reçoit le message SIP OK, il extrait les informations de session QdS, et émet une requête de réservation de ressources auprès du point d'accès QdS responsable de son domaine.

Si le réseau d'accès sous-jacent n'est pas en mesure d'assurer un support QdS pour la session, le proxy QSIP transmet le message SIP OK au terminal utilisateur, puis la session multimédia est alors établie. Cependant cette session n'aura pas de support QdS. Les états de QdS précédemment installés dans le proxy QSIP sont traités de manière que le proxy Q-SIP coté appelé gère correctement les éventuelles retransmissions de messages SIP.

Si le point d'accès QdS donne une réponse positive, les informations d'état de QdS sont mises à jour à l'intérieur du proxy Q-SIP, ce dernier transmet le message SIP OK au client SIP, et la session multimédia est établie avec le support QdS dans le sens appelant-vers-appelé.

Quand la session multimédia se termine, toutes les ressources qui ont été réservées pour la session, doivent être désallouées. Le client SIP envoie le message BYE au proxy QSIP. Si ce message concorde avec les informations d'états QdS pour une session, le proxy QSIP envoie une requête de libération de ressources auprès du réseau d'accès sous-jacent. Puis le proxy QSIP transmet ce message BYE jusqu'au client SIP distant, en passant bien entendu par l'autre proxy QSIP qui effectuera les mêmes opérations dans le domaine opposé.

1.5.3.3. Modèle de QdS : QoS-Assured vs QoS-Enabled

Comme nous l'avons vu précédemment, les ressources requises pour une session à QdS peuvent ne pas être disponibles. Face à ce scénario, il existe deux modes de fonctionnement basés sur deux modèles de QdS :

- « QoS-Assured », la session ne devrait pas être établie si les ressources nécessaires ne sont pas disponibles : dans ce cas, la QdS devrait être installée avant d'alerter l'utilisateur (sonnerie), évitant ainsi que l'utilisateur réponde à un appel quand les ressources ne sont pas disponibles.
- « QoS-Enabled », la session est établie quelque soit la disponibilité des ressources QdS ; éventuellement, l'utilisateur peut être signalé de la présence ou pas de QdS. Dans tous les cas, la réservation de ressources n'est pas une pré condition obligatoire et peut être exécutée en parallèle avec l'établissement de la session multimédia.

1.5.3.4. Conclusion

L'extension QdS apportée au protocole SIP offre des fonctionnalités QdS de réservation de ressources auprès du réseau sous-jacent. Ces fonctionnalités peuvent être implémentées sans avoir à modifier les clients SIP standards. Ceci permet de conserver la compatibilité avec le protocole SIP initial.

1.6. Architecture à Qualité de Service pour les réseaux nouvelle génération

Afin de répondre à la demande croissante de QdS sur le réseau Internet, de nombreuses architectures à Qualité de Service ont vu le jour. Nous proposons dans cette section de présenter deux de ces architectures à QdS. La première, SatSix, est orientée réseau d'accès sans fil (satellite et boucle locale radio). La seconde, EuQoS, est plus générale et propose des fonctionnalités à QdS de bout en bout.

1.6.1. Architecture SatSix

Le but du projet SatSix [42] est d'implémenter et valider des concepts innovants et solutions QdS à moindre coût pour les systèmes satellite large-bande de type DVB-S2/RCS.

Parmi ces solutions, nous pouvons citer l'introduction de la pile IPv6, ou encore le développement de réseaux hybrides combinant l'utilisation de la technologie satellite avec celle de la boucle locale radio (WiFi et WiMax).

La Figure 14 représente un exemple de topologie réseau du système SatSix. Nous retrouvons les principaux éléments d'un réseau satellite décrits en section 1.2.3, à savoir le satellite, les STs, le Gateway (RSGW) et le NCC. Des routeurs/switch interconnectent les STs avec les points d'accès WiFi et WiMax, permettant aux utilisateurs d'utiliser des interfaces sans fil pour se connecter au réseau satellite.

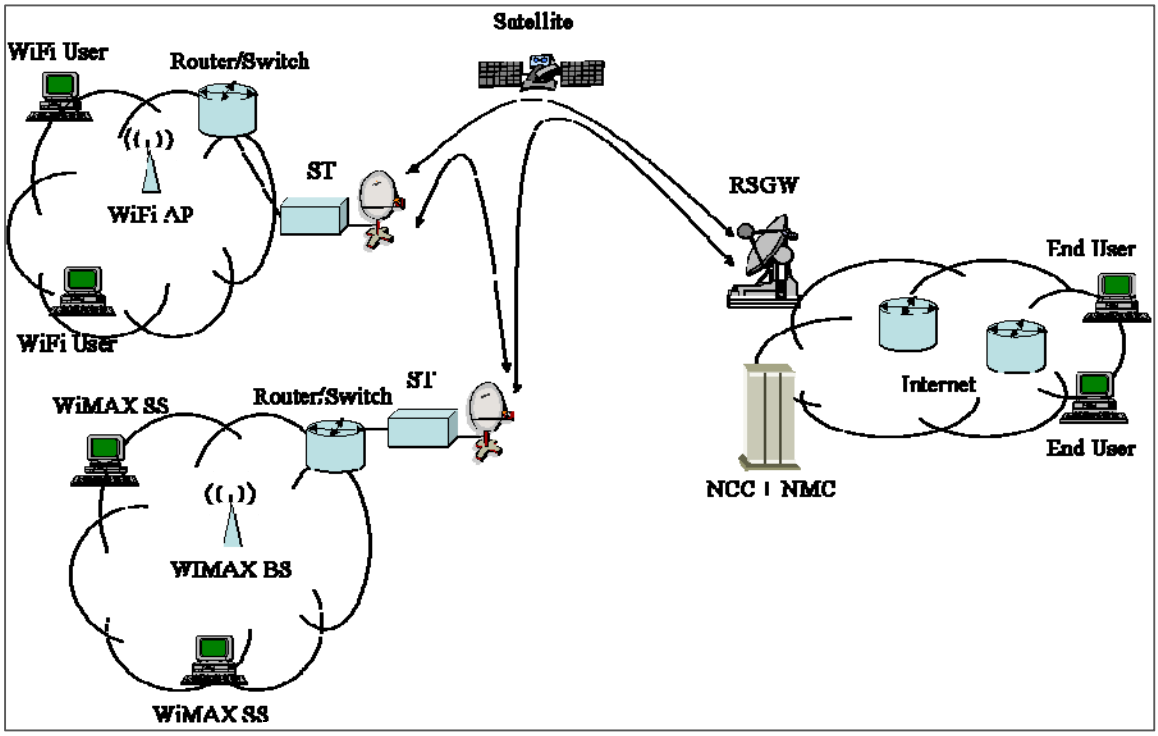


Figure 14 : Architecture réseau pour topologie mesh regénératif

Mis à part le support QdS de bout en bout, cette architecture propose diverses fonctionnalités :

- une gestion dynamique de la QdS selon les applications et les besoins des utilisateurs. Le segment satellite peut servir d'élément de bordure d'un domaine DiffServ afin de fournir une QdS de bout en bout au niveau réseau. Plusieurs entités permettent l'utilisation de signalisation et mapping de paramètres QdS : proxies SIP, agent et serveur QoS, PEP, compression IP, QdS niveau IP, QdS niveau MAC, RRM, agent et serveur C2P.
- une gestion du multicast à la fois pour IPv4 et IPv6 : les ST agissent comme des routeurs multicast MLDv2 (Multicast Listener Discovery) [43].
- des fonctionnalités de sécurité au niveau application telles que TLS (Transport Layer Security) [44], SSL (Secure Sockets Layer) et DTLS (Datagram Transport Layer Security) [45], des protocoles de gestion de clés tels que SATIPSec [46] et GSAKMP (Group Secure Association Key Management Protocol) [47], un système de distribution de clés tel que LKH (Logical Key Distribution) [48].
- une amélioration du standard de mobilité IPv6 dans le système satellite, avec l'utilisation de Mobile IPv6 [49]. Une entité MAP (Mobility Anchor Point) est localisée côté ST et le HA (Home Agent) du côté Gateway. Cette topologie réduit la signalisation durant les mouvements intra-domaine.

L'une des principales caractéristiques des différentes architectures QdS est la séparation entre les services et le réseau sous-jacent. Ainsi de nouveaux services sont mis en place indépendamment du réseau et de la technologie d'accès.

Il apparaît alors deux plans distincts (c.f. Figure 15) : le plan application (service) qui fournit les services aux utilisateurs. Les services sont invoqués par les protocoles de signalisation de session utilisateur tels que SIP. Cette signalisation permet la description des caractéristiques QdS de la session. Le plan de transport fournit un service de transport orienté paquet ainsi que des mécanismes de QdS permettant de garantir le niveau de QdS requis.

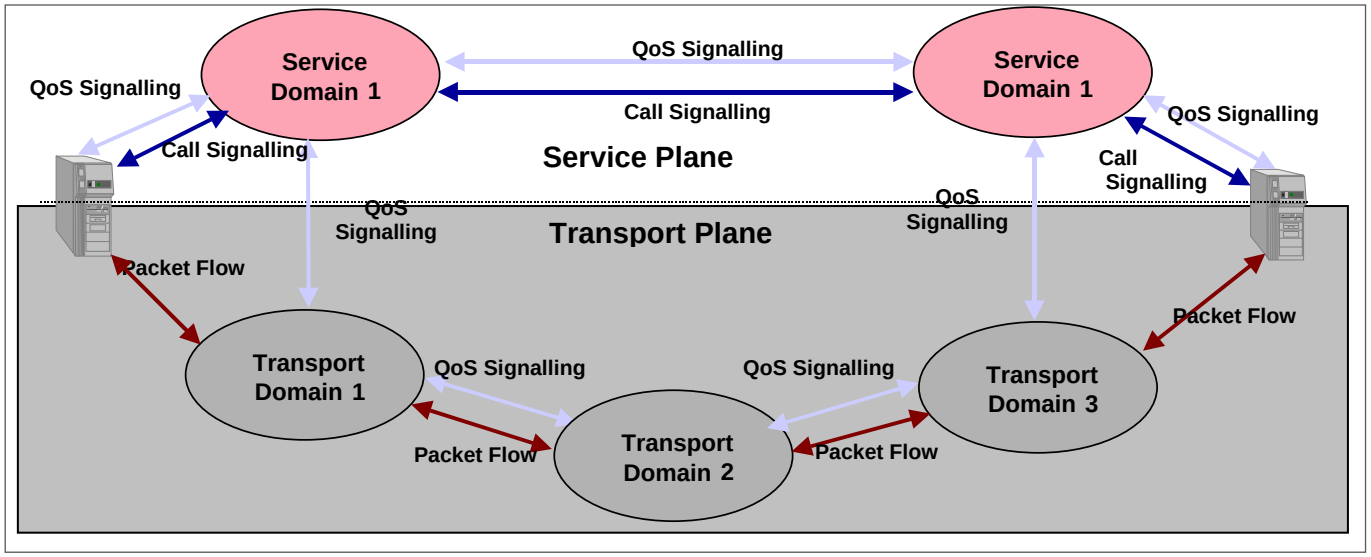


Figure 15 : Vue globale des plans de service et transport

L'architecture QdS SatSix est basée sur DiffServ et utilise deux approches :

- « orienté IP », où une signalisation spécifique est utilisée entre le ST et la Gateway uniquement dans la topologie étoilée ;
- « orienté MAC », où une communication entre le ST et la Gateway est basée sur le protocole C2P, qui ajoute une complexité et un délai dans la communication, mais a l'avantage de pouvoir adresser à la fois les topologies étoilée et maillée.

La Figure 16 présente l'architecture fonctionnelle du système SatSix. Plusieurs éléments participent à la gestion de la QdS :

Le proxy SIP utilise la signalisation d'établissement de session afin d'informer le QoS Server des caractéristiques des sessions à venir.

Pour les applications n'utilisant pas de protocole de signalisation, le QoS Agent permet à l'utilisateur de sélectionner le niveau de QdS pour chaque application en cours.

Le QoS Server est en charge de collecter les informations de QdS provenant des proxies SIP et QoS Agent concernant les flux utilisateurs, et lancer la configuration des mécanismes QdS présents aux couches IP et MAC.

L'entité PEP permet d'accélérer les connexions TCP et HTTP en se basant sur des informations de niveau physique.

Une compression au niveau IP permet de diminuer « l'overhead » des entêtes des protocoles, et ainsi limiter la charge et le délai dans le réseau.

Une couche IP QoS est implémentée au sein du ST afin de fournir diverses fonctions de routeurs de bordure DiffServ : classification, marquage de paquets, « policing/shaping/dropping/scheduling ».

Au niveau MAC, plusieurs files de transmission sont implémentées avec des mécanismes de dropping/scheduling selon la quantité de ressources fournie par le RRC Agent.

Le RRC Agent, situé côté ST, est équivalent au client DAMA. Il est chargé de générer les requêtes de ressources selon les informations provenant des couches IP et MAC.

À partir des requêtes de ressources des STs, et les modifications de SLA faites par le serveur C2P ou l'ARC, le RRC Server (serveur DAMA) construit le plan d'allocation de la voie retour satellite.

Dans l'approche orientée IP, l'ARC (Access Resources Controler) effectue le contrôle d'admission (CAC) et informe le RRC server du niveau de QdS requis.

Dans l'approche orientée MAC, le C2P Agent et C2P server gèrent respectivement les connexions MAC dans le ST et dans le NCC.

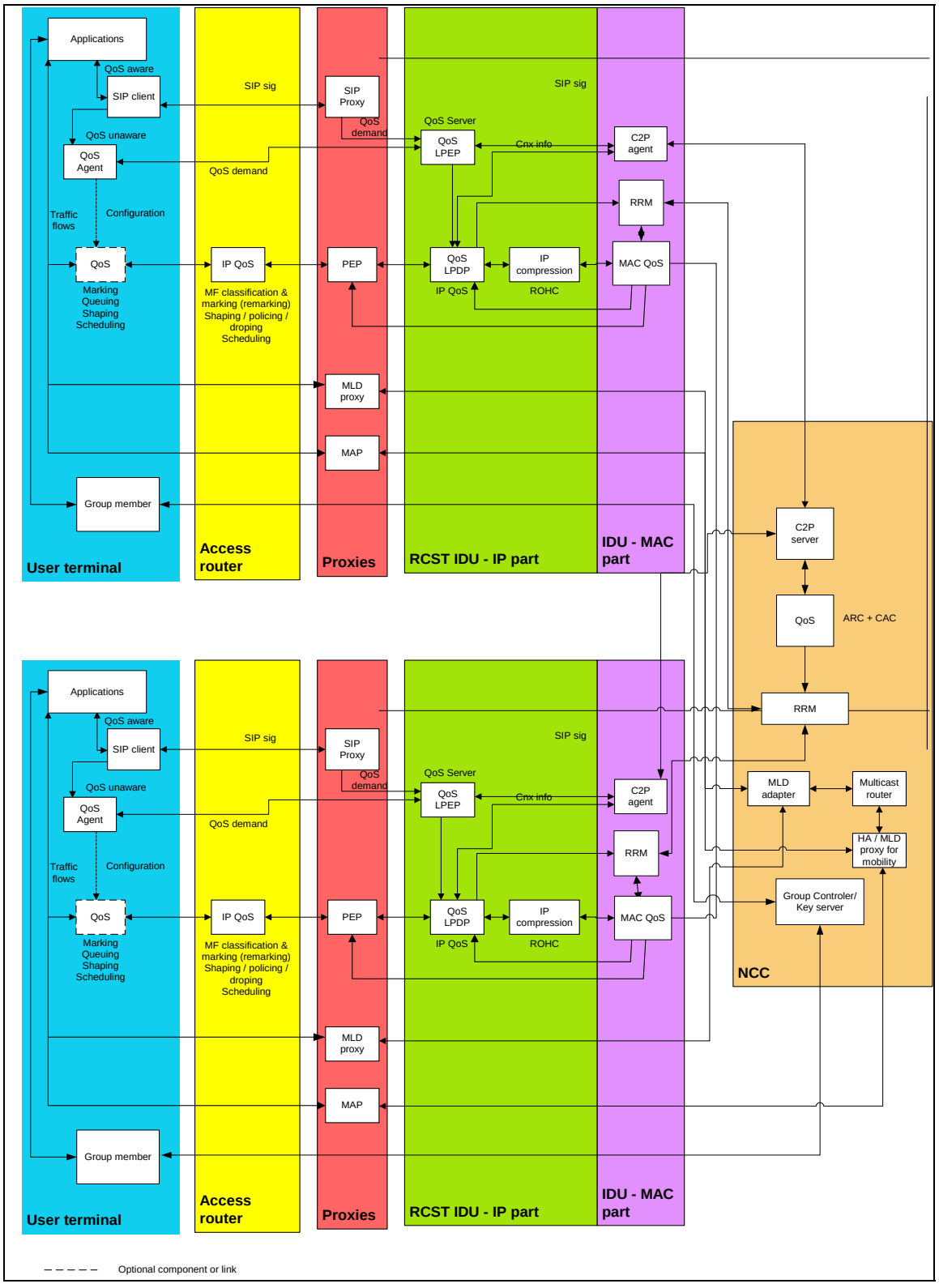


Figure 16 : Architecture fonctionnelle de référence pour topologie mesh regénératif

1.6.2. Architecture EuQoS

Introduction

EuQoS (« End-to-end Quality of Service support over heterogeneous networks ») [50] [51] est un projet européen IST6 dont l'objectif principal est de fournir une architecture (système EuQoS) permettant de garantir la qualité de service de bout en bout dans un environnement le plus général possible, multi-opérateur (multi-domaine), multi-service et multi-technologie (xDSL, LAN, Wi-Fi, UMTS, Satellite ou réseau de coeur). Cette architecture vise essentiellement les utilisateurs d'applications de type : voix sur IP, vidéoconférence, vidéo-streaming, télé-enseignement, télé-engineering et télémédecine.

Nous allons ici présenter les principales fonctionnalités de cette architecture. De plus amples détails sont fournis dans [52].

L'architecture EuQoS intègre un large ensemble de mécanismes divers : autorisation, authentification, négociation de service, contrôle d'admission, signalisation, surveillance et métrologie, ingénierie de trafic et optimisation des ressources.

Afin de garantir les propriétés de QdS, cette architecture définit des classes de service (appelées e2e CoS) avec une portée de bout en bout pour les réseaux et les domaines.

L'architecture EuQoS offre également la création des chemins à QdS garantie, de bout-en-bout (appelé EQ-Path), à travers plusieurs domaines (AS) et technologies. Chaque EQ-Path est associé à un ensemble de paramètres de QdS, en particulier à une e2e CoS.

Architecture générale du système EuQoS

Afin de réduire sa complexité et coordonner efficacement ses sous-systèmes, le système EuQoS est divisée en plusieurs plans (c.f. Figure 17) : Plan de Service, Plan de Contrôle et Plan de Transport.

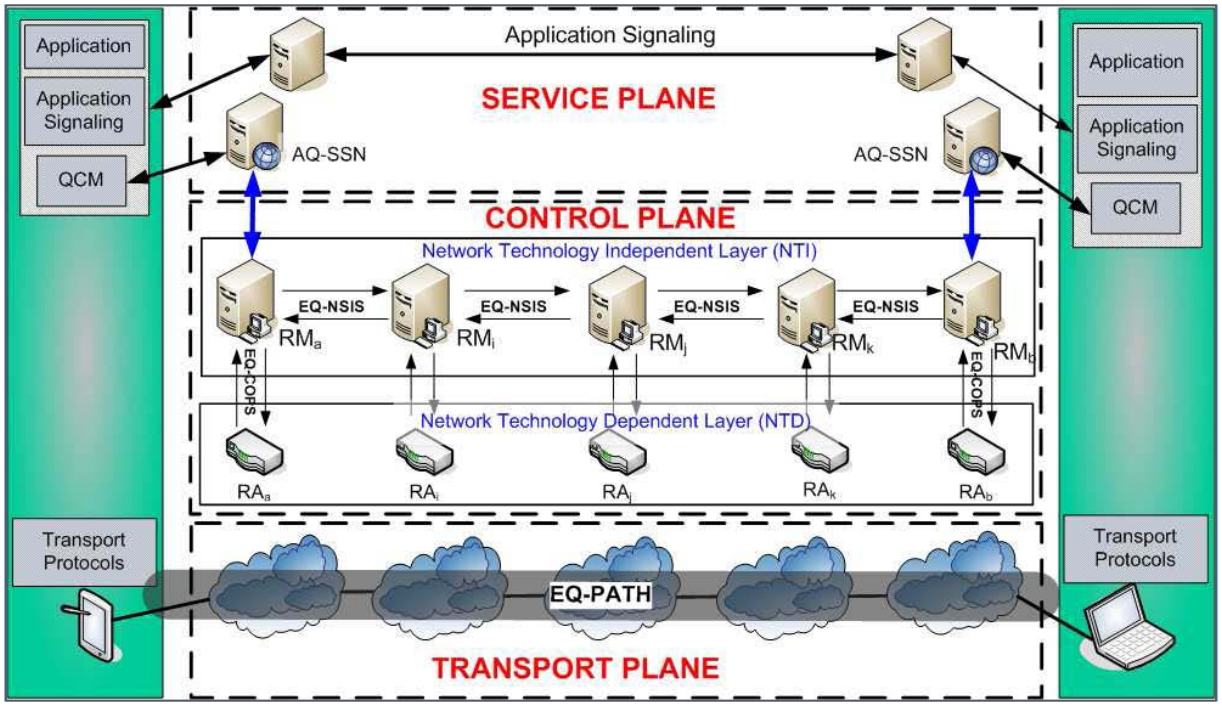


Figure 17 : architecture du système EuQoS

Le plan de service permet de déclencher les processus de choix et de réservation des ressources applicatives et réseau. Il est aussi responsable de la sécurité, de l'authentification, de l'autorisation et de la tarification (SAAA).

Le plan de contrôle assure la gestion de la QdS tout au long du chemin des données. Il traduit les paramètres de QdS fournis par les niveaux supérieurs en paramètres spécifiques à chaque domaine et gère les processus de réservation de ressources sur ce chemin. Le plan de contrôle est divisé en deux niveaux :

- Un niveau générique, indépendant de la technologie sous jacente (NTI-Network Technology Independent). Il fournit les interfaces nécessaires pour les interconnexions avec les domaines adjacents ;
- Un niveau dépendant de la technologie sous jacente (NTD-Network Technology Dependent), donc spécifique à chaque type de réseau, concepteur ou administrateur de domaine.

Le plan de transport permet de « construire » le chemin EQ-Path pour le transport des données des clients.

Composants logiciels

La Figure 18 présente les principaux composants de l'architecture EuQoS, leur localisation (client ou serveur), ainsi que les méthodes de communication entre ces différents composants, notamment les protocoles utilisés pour transporter l'information.

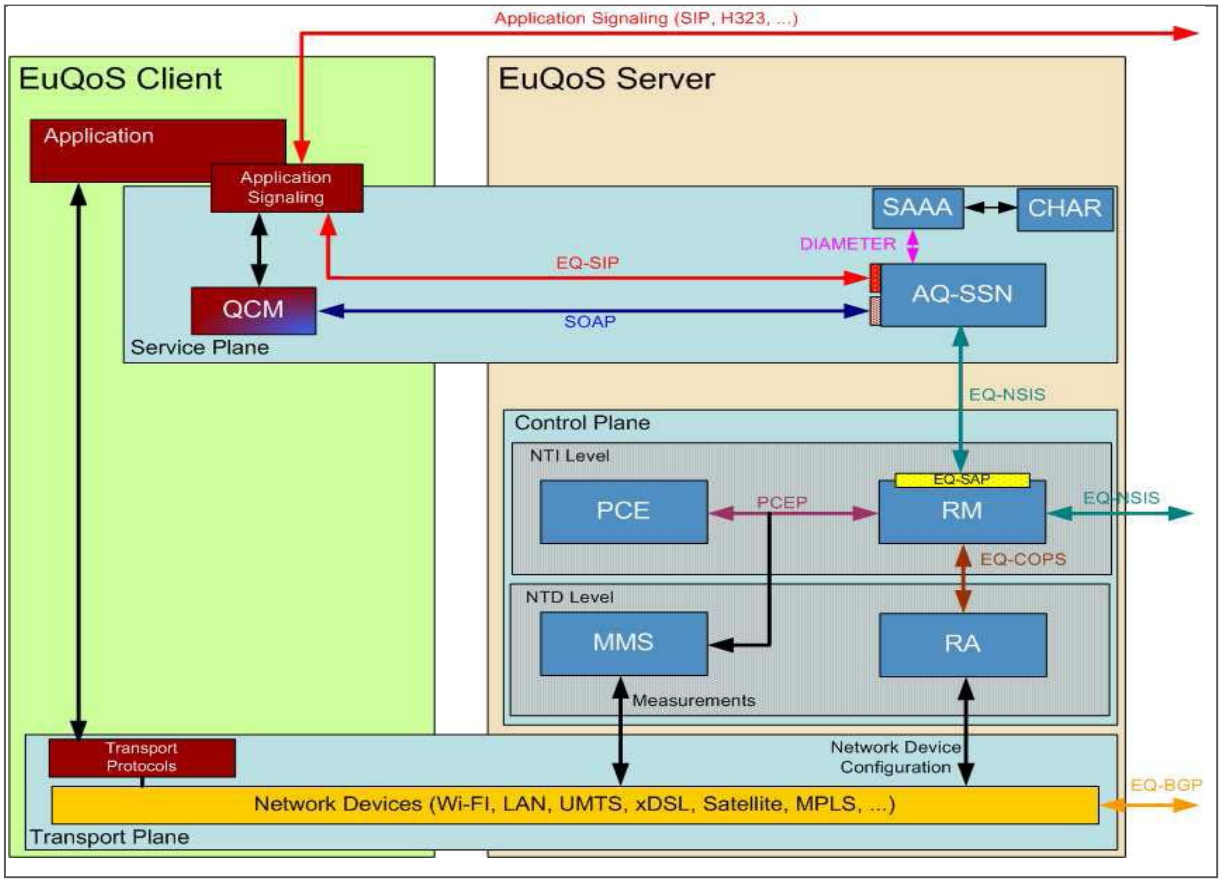


Figure 18 : composants du système EuQoS

Le **client EuQoS** est situé sur l'équipement utilisateur. Il permet à l'application d'exprimer ses besoins en QdS, d'échanger des informations de signalisation entre applications clientes (ex. SIP), de traduire les paramètres QdS au système EuQoS afin d'invoquer les services (module QCM (Quality Control Module)). Au niveau transport, un module « Transport Protocols » offre des mécanismes, des protocoles et des services enrichis (ex. ETP) pour le transfert de données.

Le **serveur EuQoS** implante les plans précédemment évoqués :

- le plan de service qui est composé de l'AQ-SSN (Application Quality Service Signaling Negotiation). Il offre l'accès au service EuQoS pour ses utilisateurs. Le SAAA est en charge de l'authentification et l'autorisation des utilisateurs. Le CHAR (charging) prend en compte la tarification du service.
- Au niveau du plan de contrôle :
    - Le niveau NTI intègre deux composants : le Resource Manager (RM) en charge de la gestion du domaine et le Path Computation Element (PCE) [53] en charge du provisionnement réseau.
    - Le niveau NTD est constitué de deux modules, le Resource Allocator (RA), en charge du traitement des requêtes de QoS et le Monitoring et Measurement System (MMS) pour assurer la surveillance et la métrologie du réseau.

Un des objectifs principaux d'EuQoS étant la création des chemins à QoS garantie, de bout-en-bout, à travers plusieurs domaines (AS), le projet a défini la notion de chemins, les « EQ-Path ». Un EQ-Path est donc un chemin entre le domaine source et le domaine destination qui offre des garanties de QoS. Chaque EQ-Path est associé à un ensemble de paramètres de QoS, en particulier à une e2e CoS.

Afin de définir ces chemins e2eCoS, le système EuQoS utilise un protocole de routage inter-domaine à QoS, EQ-BGP (Enhanced QoS Broder Gateway Protocol) [54], [55]. EQ-BGP est une variante enrichie du protocole de routage inter-domaine BGP utilisé actuellement dans l'Internet. Il permet d'établir et d'annoncer des routes qui assurent la continuité du service et répondent aux besoins des différentes e2e CoS.

En partant du chemin fourni par le protocole de routage inter-domaine EQ-BGP, l'EQ-Path est construit. Les ressources sont fournies indépendamment à l'intérieur de chaque domaine, et une signalisation est mise en place au moment de la requête de QdS afin d'associer des ressources à l'EQ-Path. Cette approche, dans le système EuQoS est appelée « loose model ».

Une deuxième approche proposée dans EuQoS, appelée « hard model », est similaire à celle utilisée dans le monde des télécommunications. Elle est basée sur le concept d'EQ-Link. Un EQ-Link est un lien virtuel entre deux routeurs de bordure de domaine. Ce lien présente des caractéristiques de QoS bien définies entre ces deux nœuds et est établi en tant que tunnels, par exemple DiffServ MPLS-TE, et peut traverser plusieurs AS.

La grande originalité d'EuQoS, en liaison avec le sous-niveau NTI, est de pouvoir intégrer différentes solutions très hétérogènes, telles que BGP (EQ-BGP [52]) et MPLS [56] (EQ-PCE), dans son architecture générale.

1.7. Conclusion

Nous avons, dans ce premier chapitre, spécifié les besoins en QdS des utilisateurs d'applications à travers des paramètres de performance tels que le délai de bout en bout, la gigue, le taux de perte, le débit. Puis, en parcourant les protocoles implémentés du modèle Internet, nous constatons qu'il existe, à chaque couche, des fonctionnalités et mécanismes QdS capables de répondre aux divers besoins de ces applications les plus exigeantes. Cependant, toutes ces solutions sont basées et conçues selon le paradigme de « l'architecture en couche » : chaque couche fonctionne indépendamment par rapport aux autres : par conséquent elles sont conçues et optimisées séparément. Il en va de même pour les mécanismes QdS proposés.

Il y a plusieurs avantages à cette approche. La modularité, la robustesse et la conception sont facilement réalisées. Seules les interfaces avec les couches adjacentes nécessitent d'être définies afin que le service fourni par cette couche puisse être accessible par les autres couches. La modularité fournie par les couches permet une combinaison arbitraire des protocoles. D'un point de vue architectural, cette modularité améliore également la maintenance lorsqu'une nouvelle version d'un protocole est à insérer sans avoir à changer le reste de la pile réseau.

Cependant, dans les environnements sans fil mobiles et satellite, les propriétés des différentes couches ont des interdépendances substantielles. Les canaux de transmission et les modèles de trafic sont bien plus imprévisibles que dans des réseaux filaires. Une conception modulaire peut donc être sous-optimale quant à l'exécution et à la disponibilité dans les réseaux satellite d'accès.

Afin de répondre aux exigences strictes de Qualité de Service des utilisateurs et des applications multimédia dans un système satellite DVB-S2/RCS, le système de communication ainsi que les mécanismes QdS, doivent pouvoir s'adapter **dynamiquement et globalement** aux situations de changement de profil de trafic et de conditions réseau. Ce besoin ne peut être adressé par l'architecture réseau protocolaire traditionnelle, même avec ses fonctionnalités QdS.

Des signaux prédéfinis pour informer des événements tels que l'échec de livraison de données entre les protocoles ont été utilisés intensivement comme moyen de partage d'informations utiles entre les couches dans les réseaux filaires et sans fil. Il s'agit par exemple du mécanisme de notification explicite de congestion (Explicit Congestion Notification : ECN) [57]. Les routeurs intermédiaires l'utilisent pour informer la couche transport (TCP, et DCCP) d'une éventuelle congestion. Ou encore les évènements produits en couche liaison (changements de lien en mobilité) qui sont transférés aux couches intéressées comme la couche réseau qui implémente les mécanismes de mobilité.

Une approche coopérative où l'adaptation est coordonnée entre les différentes couches d'une plate-forme, est appelée **Cross-Layer**, afin d'exploiter la pleine connaissance du statut de réseau rassemblé à différentes couches. Ces couches peuvent être adjacentes, ou même non adjacentes dans la pile. Pour concevoir des systèmes basés sur cette approche, il est nécessaire de définir correctement les nouvelles interactions entre les couches. Dans tous les cas, cette approche, par définition, viole le concept strict des modèles OSI et Internet en couches.

On constate au final que :

- **Les systèmes stricts de protocoles en couches permettent, aux concepteurs, de pouvoir facilement optimiser une couche sans avoir à faire face à la complexité et l'expertise associée en considérant les autres couches.**
- **Les solutions par l'adaptation et coopération cross-layer cherchent à augmenter les performances globales des communications de bout en bout du système pour répondre aux besoins QdS des utilisateurs.**

L'approche de conception a donc évolué du point de vue du concepteur, vers le point de vue de l'utilisateur.

Les travaux de cette thèse proposent d'utiliser cette approche cross-layer afin de répondre à la problématique double : (i) Fournir un service de qualité aux utilisateurs d'applications multimédia interactives (VoIP, visio-conférence); (ii) Optimiser l'utilisation des ressources du système de communication, en particulier dans un environnement réseau satellite.

Le chapitre suivant présente en détail l'approche cross-layer avec les différentes architectures à communication cross-layer existantes.

2. Concept Cross-Layer

Il y a plusieurs interprétations de la conception cross-layer. Ceci est dû au fait que la plupart des travaux dans ce domaine ont été menés indépendamment par des chercheurs de différents domaines, travaillant à des niveaux de couches protocolaires différents.

La conception cross-layer se réfère à la conception de protocole en exploitant activement les dépendances entre les couches afin d'obtenir des gains de performance : une couche se base sur certains détails de conception d'une autre couche. Ces couches peuvent être adjacentes, ou même non adjacentes dans la pile.

Elle peut être vue comme une approche coopérative où l'adaptation est coordonnée entre les couches multiples du système de communication.

En adoptant le concept cross-layer, on perd le luxe d'une conception simple et indépendante des protocoles.

En résumé, la flexibilité résultant de l'approche Cross-Layer aide à améliorer les performances de communication de bout en bout. Cependant, l'approche cross-layer peut augmenter de manière significative la complexité de conception. En effet, le système de protocoles en couches permet aux concepteurs de pouvoir facilement optimiser une couche sans avoir à faire face à la complexité et l'expertise associé des autres couches.

2.1. Communications cross-layer

Initialement, des conceptions cross-layer ont proposé de simplement fusionner deux couches adjacentes afin d'effectuer les mêmes fonctionnalités. Un exemple commun est la fusion des couches de PHY (physique) et MAC (liaison).

Une autre approche de conception est d'établir des transferts unidirectionnels ou bidirectionnels de l'information entre deux couches adjacentes ou non adjacentes. Cette conception crée de nouvelles interfaces aux couches choisies au delà de ceux déjà utilisés entre les couches pour les données.

Et récemment, un troisième type de conception commence à être utilisé. Au lieu d'établir des communications entre des couches spécifiques, ces architectures utilisent une nouvelle structure parallèle qui agit en tant que base de données partagée de l'état du système. Elle est ainsi accessible à n'importe quelle couche qui choisit de l'utiliser.

Certaines interactions cross-layer impliquent des conversations étendues et complètes entres couches, alors que d'autres impliquent uniquement de simples notifications, qui peuvent servir uniquement de conseils. Elles sont utilisées par la couche destinataire, seulement quand elles sont considérées appropriées.

Selon [58], la conception cross-layer peut être classée comme suit (c.f. Figure 19) :

- Flux d'informations de bas en haut (upward) : les couches supérieures requièrent des informations provenant des couches inférieures.

- Flux d'informations de haut en bas (downward) : les couches supérieures fournissent des paramètres de configuration aux couches inférieures.
- Flux d'informations bidirectionnel : deux couches différentes peuvent collaborer entre elles en échangeant des informations.
- Fusion de couches adjacentes : plusieurs couches adjacentes sont conçues ensemble pour former une « super couche ». Ainsi, le service fourni par cette super couche est la collection des services fournis par ces couches adjacentes.

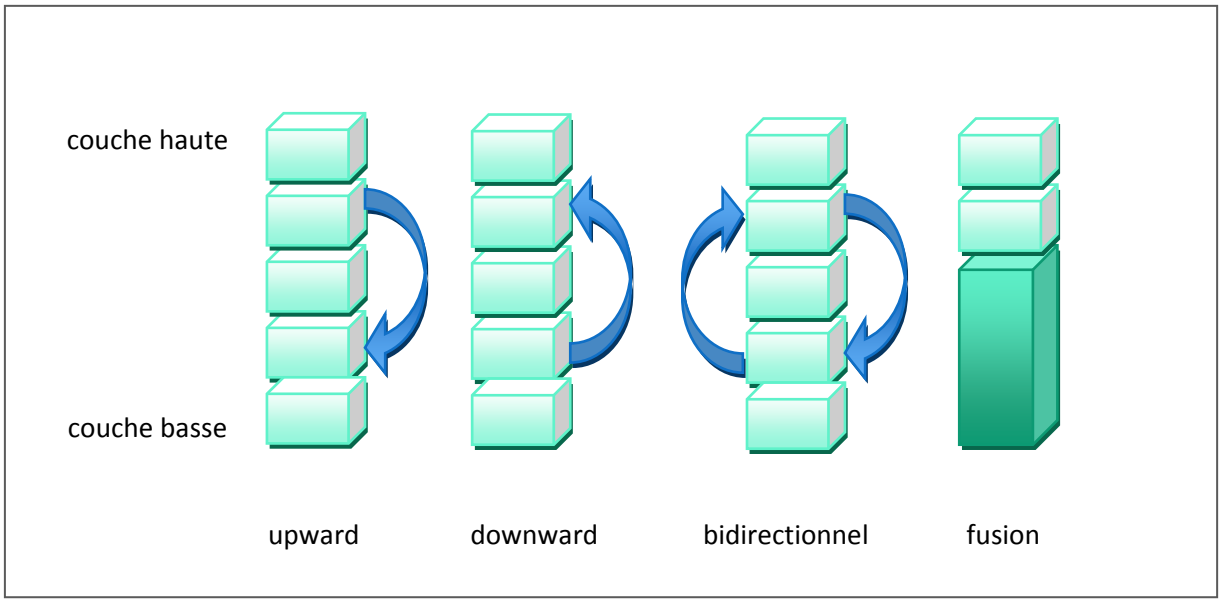


Figure 19 : types de communication cross-layer

2.2. Architectures Cross-Layer

Sachant que plusieurs architectures cross-layer ont été développées, nous proposons d'abord de les classifier, avant d'en présenter quelques exemples. Il faut tout de même préciser qu'aucune de ces architectures n'a été normalisée, ce qui risque à long terme d'impliquer des incompatibilités entre les systèmes. Il serait donc intéressant de pouvoir proposer une architecture modèle convenant à tous.

Il est possible de classifier les possibilités d'architectures Cross-Layer en trois catégories :

Une première catégorie permet aux couches, même si elles ne sont pas adjacentes, de communiquer directement entre elles (c.f. Figure 20), afin d'optimiser la QdS par exemple. Pour cela, il est nécessaire de créer de nouvelles interfaces, d'intégrer de nouvelles routines aux couches qui leur permettront la réception et le traitement des données Cross-Layer. Cependant, il faut noter que le nombre de routines à implémenter sera variable selon le nombre de protocoles à satisfaire. De plus, cette méthode ajoute un certain nombre de contraintes telles que le ralentissement de l'exécution du code puisque le code Cross-Layer a été ajouté, par conséquent une mise à jour difficile à maintenir.

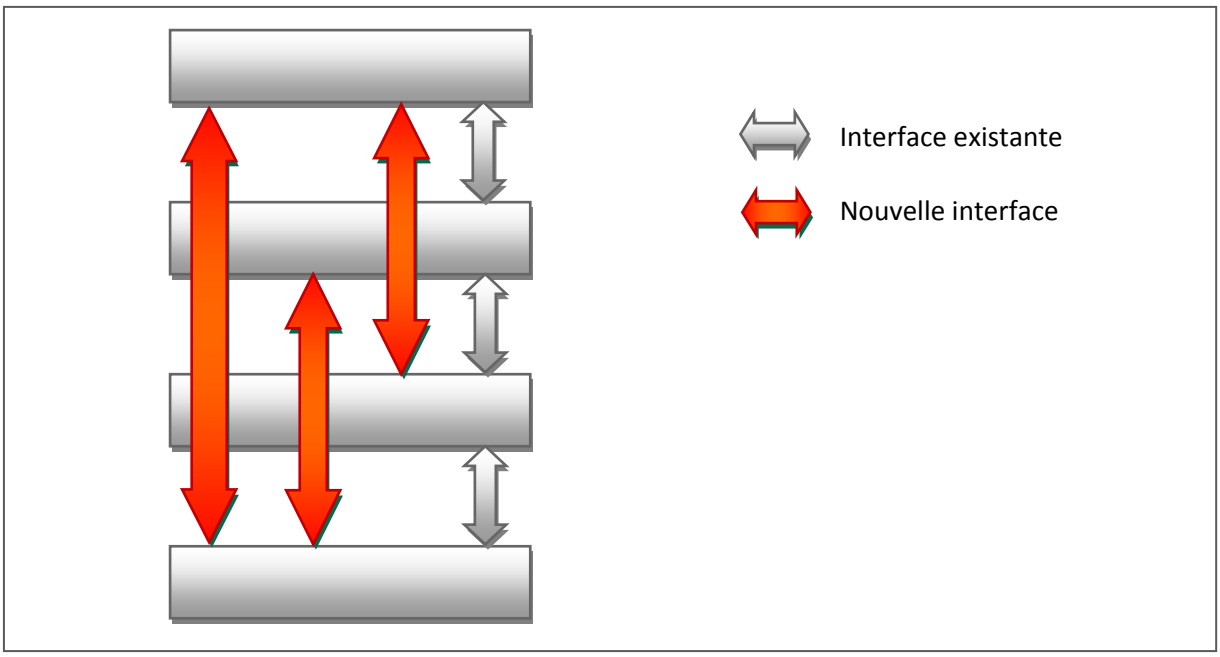


Figure 20 : Architecture cross-layer par communication directe

Une deuxième catégorie permet les interactions inter couches via une entité intermédiaire commune (c.f. Figure 21). Cette architecture permet de conserver le fonctionnement normal de la pile protocolaire, d'où une compatibilité avec l'architecture classique en couches. Cela permet donc de maintenir tous les avantages inhérents à une architecture modulaire en couches isolées, tels que la robustesse ou la facilité d'évolutivité. De plus, cette méthode permet une évolution continue de l'entité cross layer, par ajout ou suppression de protocoles.

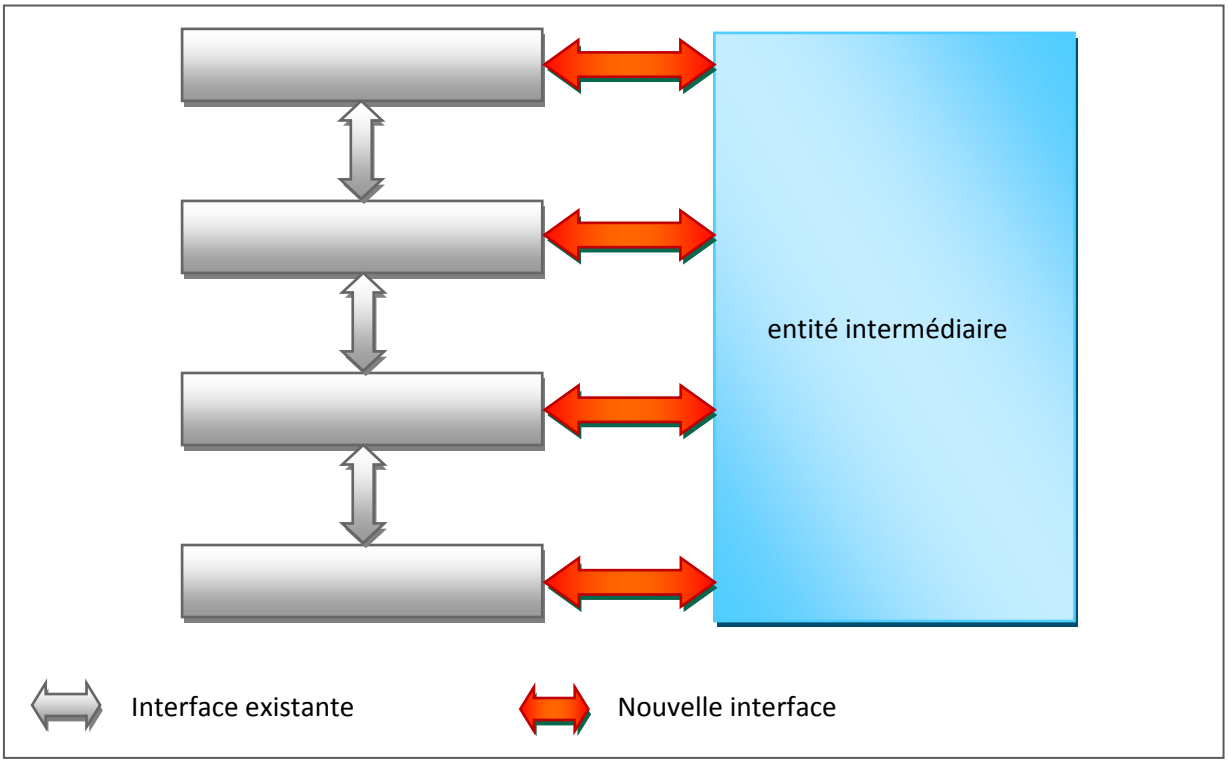


Figure 21 : Architecture cross-layer avec entité intermédiaire

Une troisième catégorie s'abstrait complètement du modèle en couche (c.f.Figure 22), elle est donc bien plus flexible mais elle viole complètement les préceptes du modèle en couches.

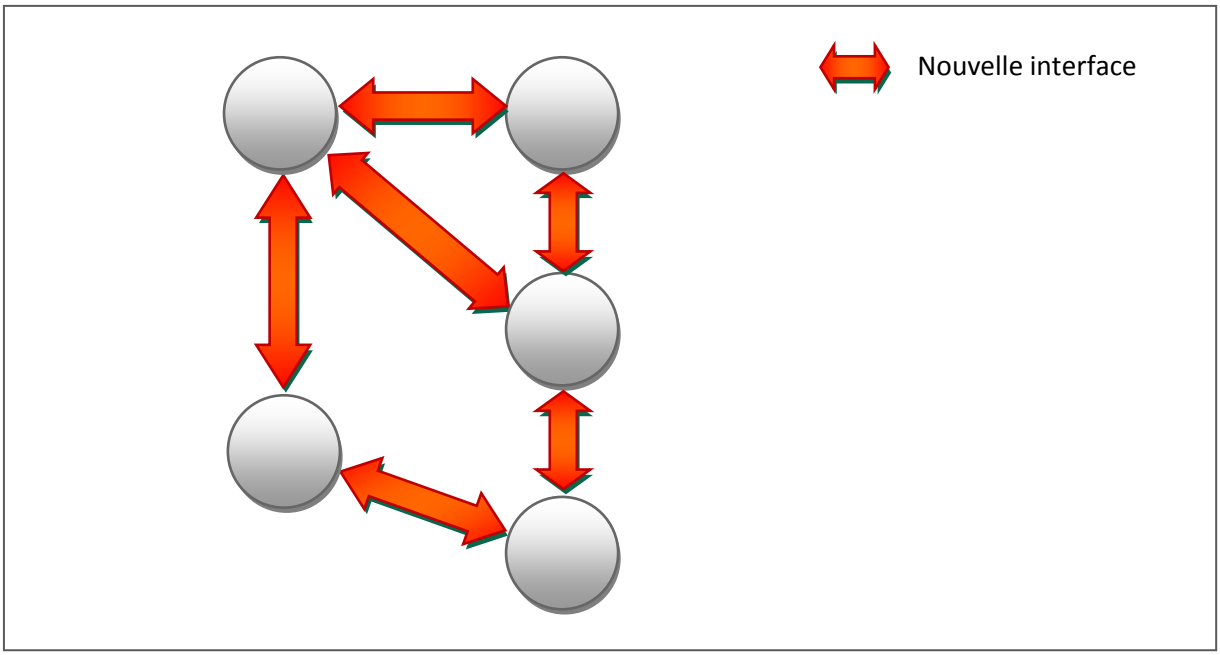


Figure 22 : Architecture cross-layer abstraite du modèle en couche

Il existe des exemples d'architectures Cross-Layer, qui se classent tous dans chacune des trois catégories ; nous allons dans la suite en présenter quelques uns.

2.3. Architectures de type « directe »

2.3.1. Packet header

Cette architecture [59] sera principalement utilisée lorsque les couches basses devront s'adapter aux couches plus hautes. Des tuyaux de signalisation inter couches stockent l'information Cross-Layer dans les en-têtes d'extension des paquets IPv6 (c.f. Figure 23).

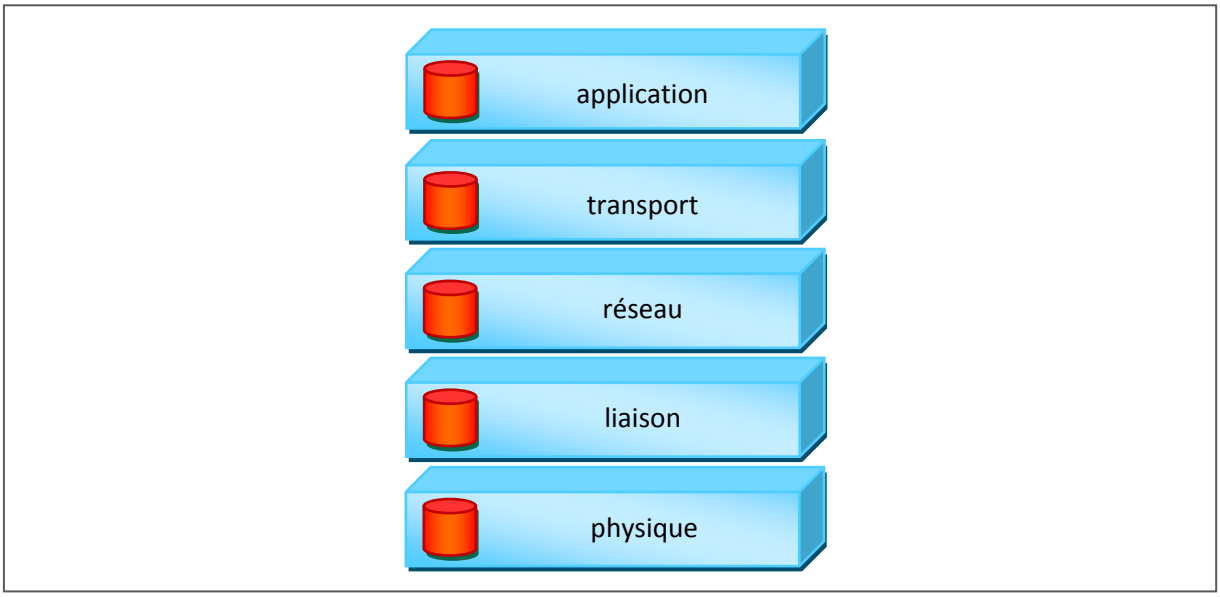


Figure 23 : Packet Header

Cette architecture se sert du champ optionnel d'information réseau de l'entête du paquet IPv6. L'information cross-layer est stockée dans le WEH (Wireless Extension Header) via les tuyaux de signalisation inter couches (c.f.Figure 24). Cette méthode utilise les paquets de données IPv6 comme porteurs de messages.

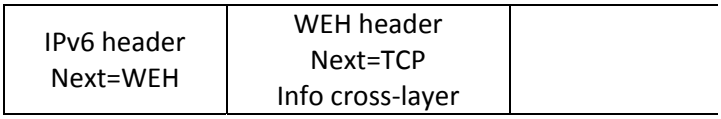


Figure 24 : en-tête IPv6 avec info cross-layer

2.3.2. Architecture basée ICMP

Cette architecture [60] utilise le protocole ICMP (Internet Control Message Protocol) largement déployé dans les réseaux utilisant IP. L'objectif est de propager les informations sur toutes les couches du dessus en utilisant des messages ICMP (c.f. Figure 25). Un nouveau message ICMP n'est généré que si un paramètre des couches basses change et dépasse un seuil convenu.

Cette méthode est plus flexible et plus efficace que la méthode des en-têtes. Cependant, les communications ne peuvent aller que du bas vers le haut. De plus, les messages ICMP sont toujours encapsulés dans des paquets IP, le message doit donc passer par la couche réseau, même si cette dernière n'est pas intéressée.

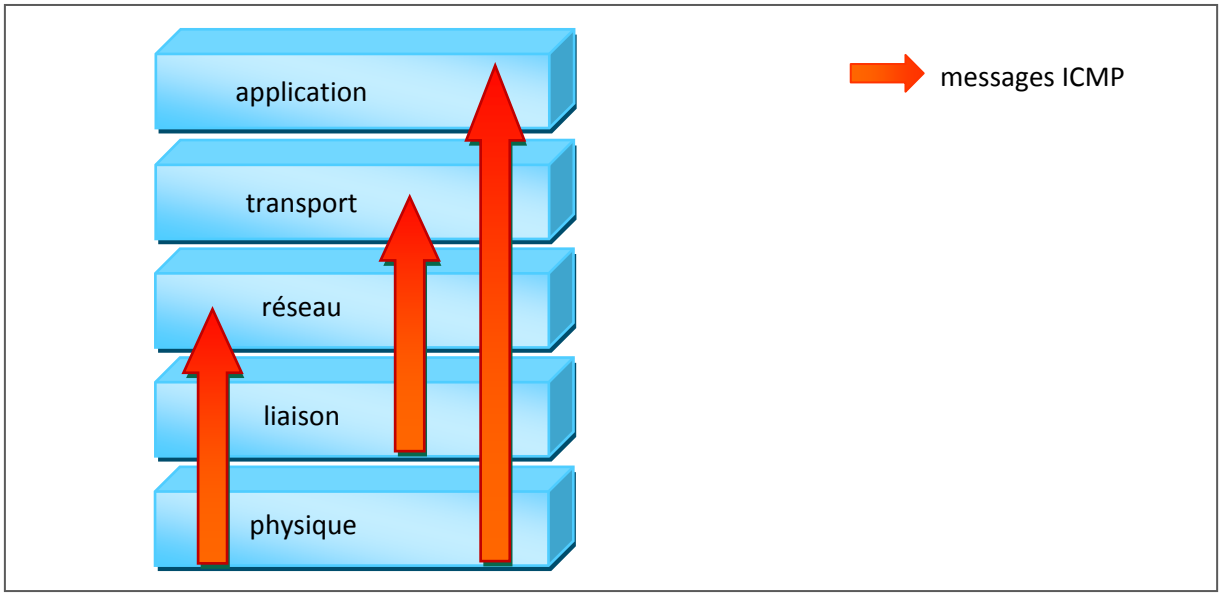


Figure 25 : ICMP

2.3.3. CLASS : (Cross-Layer Signalling Shortcuts)

CLASS [61] est une architecture flexible et facilement adaptable, qui tente de minimiser les délais et l'overhead, d'une part en permettant aux messages de passer directement d'une couche à l'autre sans intermédiaire, et d'autre part en simplifiant le format des messages internes (c.f. Figure 26).

En effet, les autres méthodes inter couches ont deux inconvénients majeurs :

- le signal se propage à travers chaque couche, même quand elles ne sont pas concernées, ce qui implique de l'overhead et des latences inutiles.
- Les formats de messages ne sont ni optimisés ni flexibles, ni prévus pour un passage à l'échelle.

L'architecture CLASS a donc été proposée pour remédier à ces défauts, avec les caractéristiques suivantes, « Direct Signalling between Non-Neighbouring Layers » :

L'idée est de passer outre le concept d'ordre des couches, mais de garder le modèle de l'architecture en couche. La communication directe entre des couches non adjacentes est possible, sans avoir à passer par les couches intermédiaires, comme avec la méthode « Packet Header » ou la méthode « ICMP ». Par exemple, la couche réseau pourrait communiquer avec la couche physique, sans avoir à passer par la couche liaison. Cette méthode permet de réduire la latence. En effet, on peut montrer que dans le cas de CLASS, la latence est (n-1) fois moins importante (n le nombre de couches traversées) que dans le cas d'une méthode où le message passe par chaque couche l'une après l'autre.

Les protocoles standards sont généralement assez lourds, avec des en-têtes importantes (ex : ICMP). CLASS tente donc de minimiser les en-têtes additionnelles et le nombre de champs afin de simplifier le format de message interne.

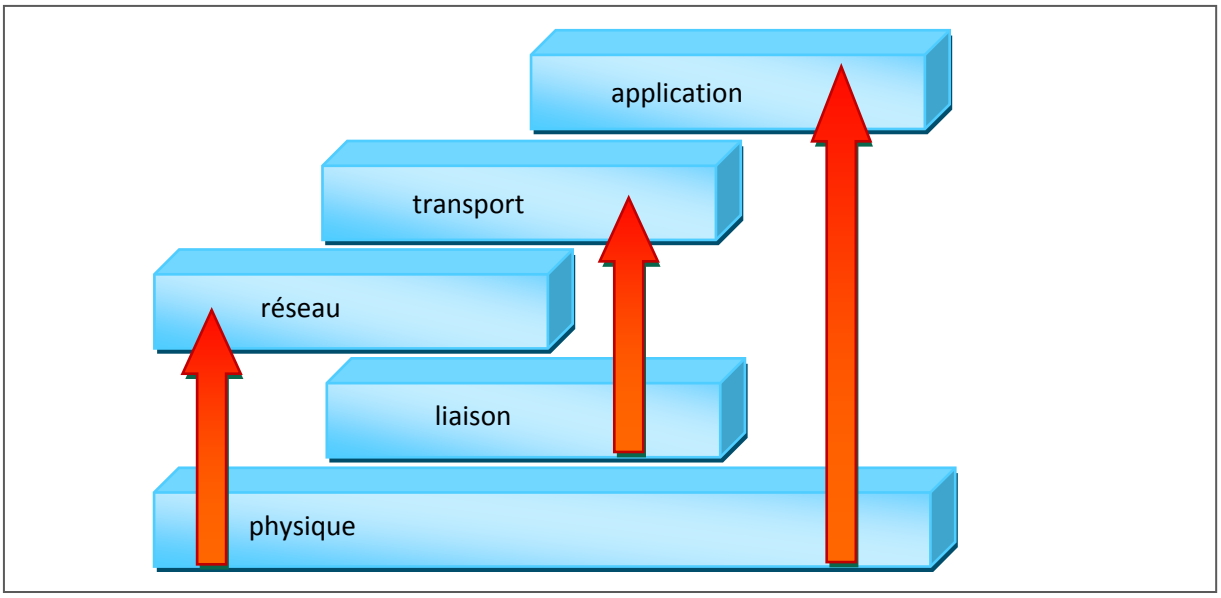


Figure 26 : CLASS

Ainsi, CLASS requiert seulement 3 champs :

- l'adresse de destination sur 1 octet : indique la couche, le protocole et l'application destination.
- l'event type (type de l'événement) sur 1 octet : indique le paramètre.
- l'event contents (contenu de l'événement) sur 2 octets : indique la valeur du paramètre.

Ces trois champs utilisent donc 4 octets, alors que la taille d'un message ICMP encapsulé dans IPv4 est de 30 octets. Il est possible de propager les informations de manière agrégée, en utilisant un champ optionnel Next Event.

2.4. Architectures de type « Entité Intermédiaire »

Nous allons maintenant nous intéresser plus particulièrement aux architectures utilisant une entité intermédiaire pour communiquer entre couches, ce qui permet de continuer à bénéficier des avantages de l'architecture en couche original.

2.4.1. XL Engine

L'architecture de XL Engine [62] gère des fonctions cross-layer locales et globales au réseau (c.f. Figure 27). L'entité principale, le CLMC (Cross Layer Management Component), est chargée des interactions inter-couches tout en gardant les avantages du modèle en couches. Elle est constituée de deux éléments :

- le CLIC (Cross Layer Interface Component) : sert d'interface entre les couches et l'entité CLMC. De plus, il collecte, met à jour les données Cross-Layer, et les présentent de telle manière qu'elles soient rapidement accessibles par tous les protocoles des couches.
- le CLE (Cross Layer Engine) : c'est le moteur Cross-Layer du CLMC, constitué de deux parties :
    - Local_CLE : utilise les données du CLIC pour générer des données plus complexes, métriques requises par les protocoles, comme le STR (Successful Transmission Rate) qui fournit une mesure de la QdS incluant les collisions et excluant les congestions, ou encore le CCR (Clear Channel Rate) qui mesure le taux de congestion pour les systèmes CSMA (Carrier Sense Multiple Access).
    - Network_CLE : permet d'avoir une vision globale sur le réseau, ce qui permet à un nœud local d'évaluer son propre état par rapport au reste du réseau, et donc de pouvoir ajuster son comportement à l'état du réseau global. Pour cela, il collecte les autres vues locales du réseau et en évalue la vue globale du nœud sur le réseau. Par exemple, quand un protocole a besoin d'une information sur le réseau, il le notifie au Network_CLE qui la lui communique dès qu'elle est disponible.

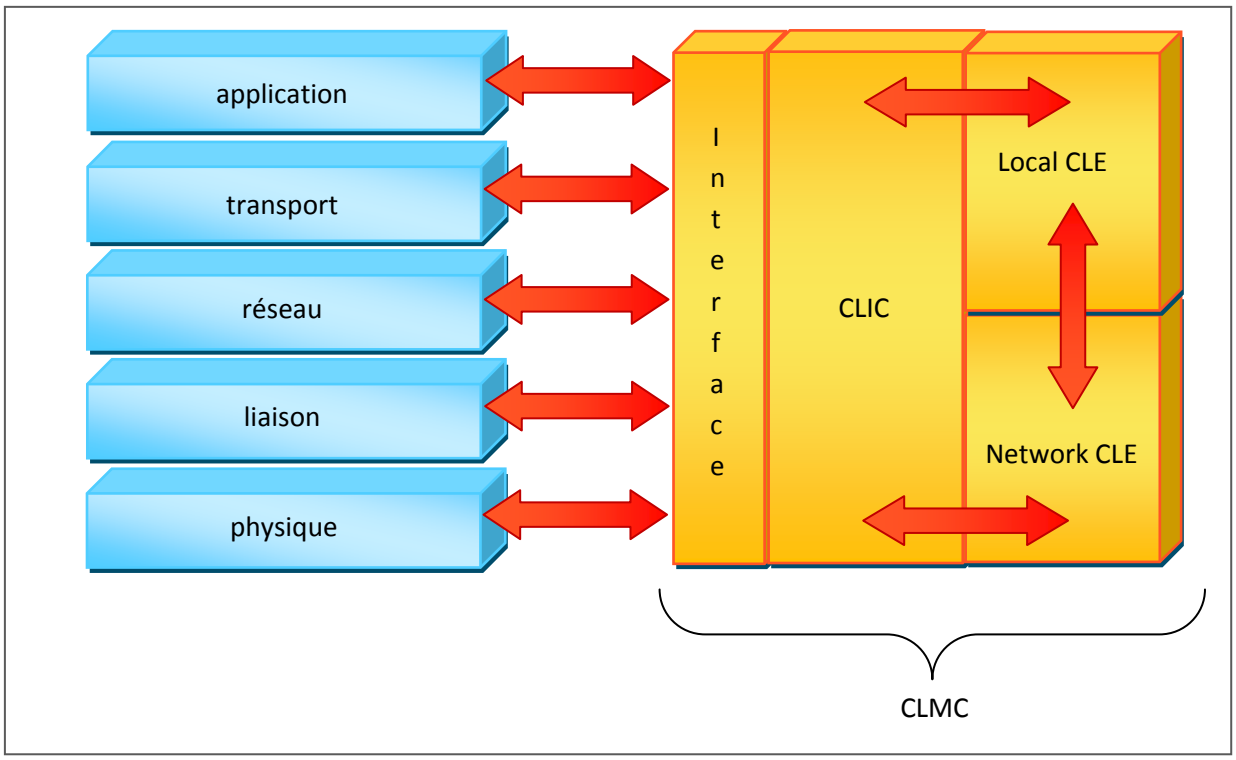


Figure 27 : XL_Engine

2.4.2. Cross Talk

CrossTalk [63] est une architecture Cross-Layer basée sur une connaissance à la fois locale et globale du réseau (c.f. Figure 28). L'objectif de cette architecture consiste à adapter le comportement d'un nœud à l'état global du réseau.

L'architecture CrossTalk n'utilise pas simplement les informations locales pour influencer le comportement des protocoles, mais aussi une vue globale du réseau. Chaque nœud disposant de cette vue globale, il peut utiliser ces informations pour comparer son propre statut local au statut global du réseau. Il évalue donc son statut relatif, ce qui lui permet de prendre les décisions adéquates concernant son propre comportement. Par exemple, si le nœud sait qu'il peut localement utiliser 60% de sa capacité, mais qu'il apprend que le réseau global ne peut en utiliser que 10%, cela lui permet d'éviter de surcharger ce dernier.

Disposer d'une vue globale du réseau est très optimiste car cela alourdit la charge de travail pour chaque nœud. Cependant, cela a l'avantage de fournir des informations optimisées et bien plus précises. Travailler en local est plus léger mais moins précis. Le système CrossTalk permet de combiner les avantages de ces deux méthodes, afin d'atteindre des objectifs globalement satisfaisants sans trop de coûts.

L'architecture CrossTalk consiste donc en deux entités intermédiaires :

- Local_View : apporte une connaissance locale du réseau et des opérations cross-layer. Il organise les informations locales, venant de chaque couche de la pile protocolaire, comme par exemple le statut d'une batterie, le rapport Eb/N0, ou encore la puissance transmise.

- Global_View : apporte une connaissance globale du réseau. CrossTalk utilise une procédure de dissémination de données. A chaque fois qu'un paquet est envoyé, il y joint une information locale. Chaque nœud recevant ce paquets extrait cette information et l'ajoute à sa vue globale.

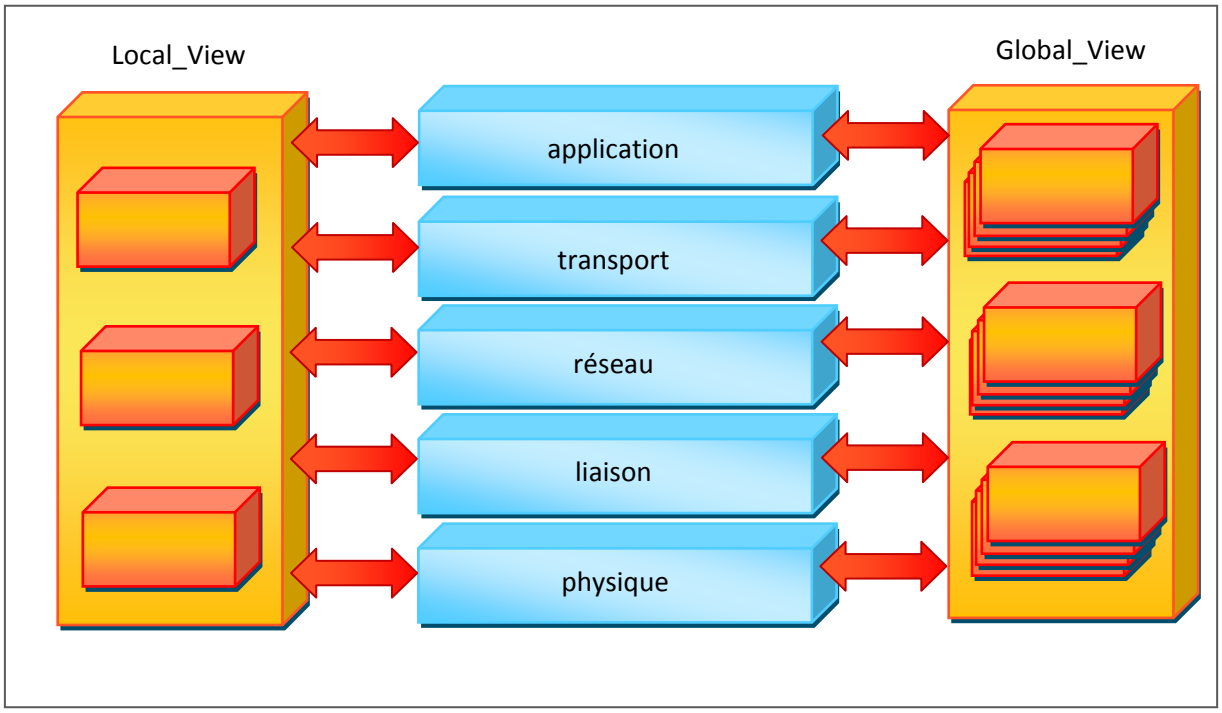


Figure 28 : Cross Talk

Cette architecture, bien que plus complexe que les autres, a l'avantage de présenter de meilleures performances.

2.4.3. ECLAIR

ECLAIR [64] est une architecture capable de supporter plusieurs types d'applications, mais elle est aussi très complexe puisque chaque interaction ou optimisation Cross-Layer passe par plusieurs entités. Cet environnement se veut générique et indépendant de l'implémentation en utilisant un module d'optimisation global et des traducteurs dépendants du système pour s'y connecter (c.f. Figure 29). Les buts de l'architecture ECLAIR sont :

- un prototypage rapide.
- la portabilité.
- l'efficacité
- le minimum d'intrusion dans les protocoles existants.

L'architecture ECLAIR est constituée de deux composants :

- le sous système d'optimisation (OSS)
- les tuning layers (TL).

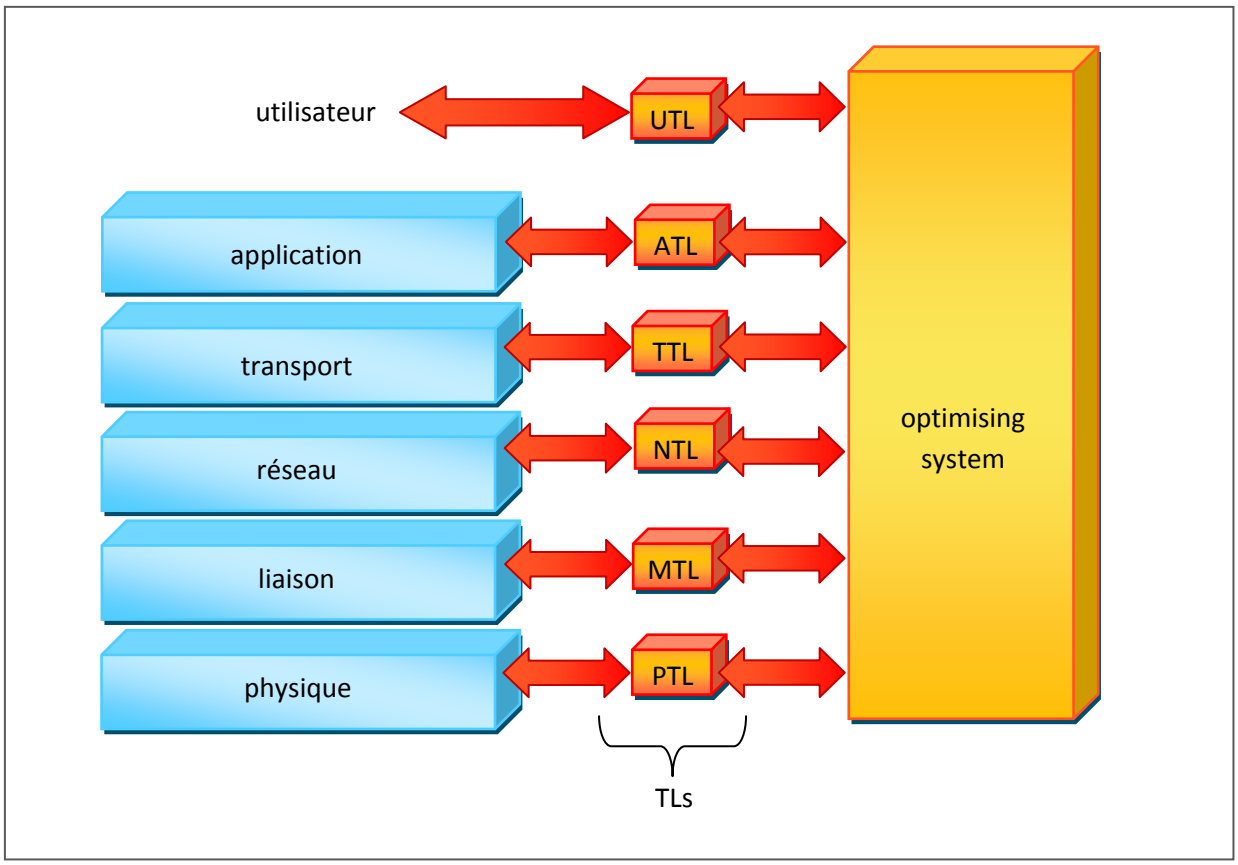


Figure 29 : ECLAIR

Le composant OSS optimise les protocoles (PO), qui maintient les algorithmes Cross-Layer. Il interagit avec la pile existante via les TLs (Tunning Layer). Chaque protocole du modèle en couche dispose de son propre TL.

Les TLs proposent une interface entre l'OSS et les protocoles du modèle en couche. Il y a une TL pour chaque couche protocolaire. Dans un souci de portabilité, une TL est constituée :

- du « generic tuning sub layer » : fournit une interface indépendante d'implémentation pour un protocole spécifique ;
- et de l' « implementation dependant access layer » : fournit des interfaces spécifiques d'implémentation pour un protocole (les implémentations du protocole TCP sous Linux, Unix, BSD et Windows sont différents)

2.4.4. GRACE

GRACE (Global Resource Adaptation through Co-opEration) [65] est une architecture combinant des adaptations globales par application, et des adaptations internes aux couches (c.f. Figure 30). Elle peut être mise en place pour des applications multimédia, qui ont des changements dynamiques de besoins devant être traités en temps réel.

Elle permet de s'adapter afin d'économiser l'énergie au mieux dans un système mobile multimédia. En effet, dans cette architecture, toutes les ressources systèmes et toutes les couches sont adaptables.

L'architecture GRACE considère 4 couches différentes, toutes connectées par un gestionnaire de ressources :

- la couche réseau
- la couche application
- la couche hardware
- la couche operating system

GRACE considère le temps CPU, la largeur de bande réseau, les caractéristiques énergétiques, la couche réseau, l'ordonnanceur CPU, l'ordonnanceur réseau et les applications multimédia.

GRACE effectue une adaptation globale continue avec un overhead minimal, tout en prédisant l'usage des ressources, et en choisissant des configurations optimisées parmi un large panel.

Les adaptations globales sont prises en charge par le gestionnaire de ressources qui choisit la configuration optimale pour chaque couche. Les adaptations locales prennent placent sur les couches. Le gestionnaire de ressources tente de choisir les combinaisons de configurations qui permettront de répondre aux besoins de l'application.

En effet, on ne peut pas utiliser les adaptations globales trop fréquemment car elles induisent trop d'overhead. Pourtant, ne pas les utiliser assez, fait perdre en précision, et de mauvaises configurations ou sous-optimisées seront choisies. C'est pourquoi GRACE utilise une approche hiérarchique, afin de bénéficier des avantages d'une approche globale tout en gardant ceux d'une approche locale.

GRACE utilise donc 3 types d'adaptations (c.f. Figure 30) :

- (schéma a) L'adaptation globale inter-couche: toutes les couches et applications sont considérées. L'objectif est d'allouer les ressources disponibles de manière à optimiser l'utilisation du système. C'est un coordinateur global qui alloue les ressources en tenant compte de toutes les configurations possibles, en calculant la performance de chaque combinaison et en sélectionnant la meilleure combinaison.
- (schéma b) L'adaptation par application inter-couche : seule une application est considérée. Cette adaptation est utilisée au début de chaque travail. Le but est de trouver la configuration attendue par le coordinateur global.
- (schéma c) L'adaptation interne : seule une couche ou application est considérée.

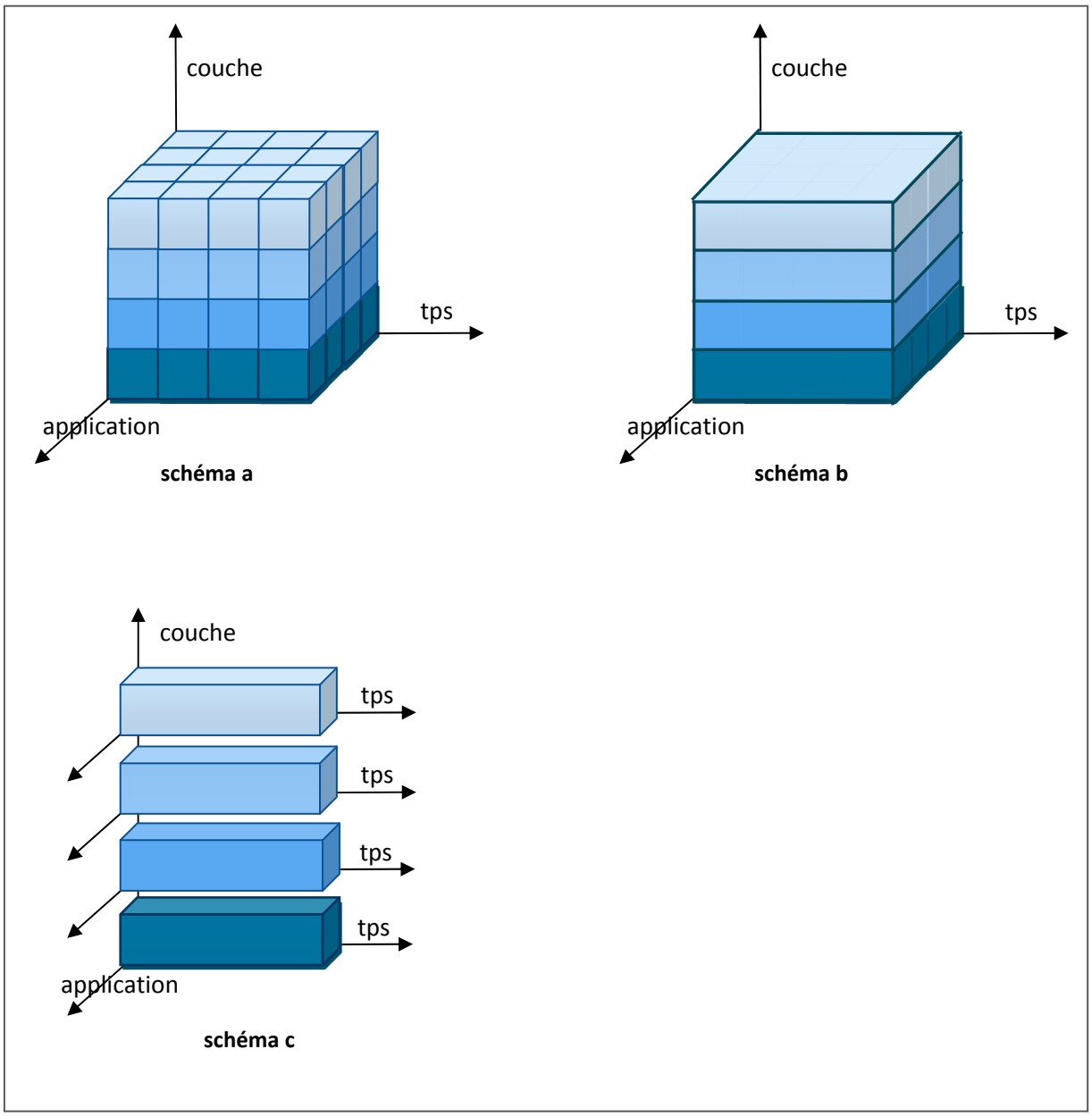


Figure 30 : GRACE

2.4.5. Local Profiles

L'architecture Local Profiles [66] est utilisée pour stocker périodiquement des informations mises à jour pour un hôte mobile dans un réseau ad hoc. Elle permet de stocker des informations cross-layer qui peuvent être utilisées par toutes les couches ultérieurement, sans avoir à modifier la pile protocolaire du modèle classique en couche. Les informations cross-layer sont extraites des couches et gardées dans des profils séparés. Les autres couches intéressées par l'information doivent alors sélectionner le profil adéquat. Cependant, il est à noter que cette solution n'est pas vraiment adaptée aux applications avec de fortes contraintes temporelles, à cause d'un temps de réponse relativement élevé.

2.4.6. Network Service

Network Service est une architecture utilisée quand les couches ont besoin d'informations de la couche physique ou de la couche liaison. En effet, elle est composée d'une entité appelée WCI (Wireless Channel Information), qui collecte et rassemble les informations sur le canal et l'état de la liaison, des couches 1 et 2 (c.f. Figure 31). Ensuite, les applications qui ont besoin de ces informations demandent au serveur applicatif de récupérer ces données sur le serveur WCI.

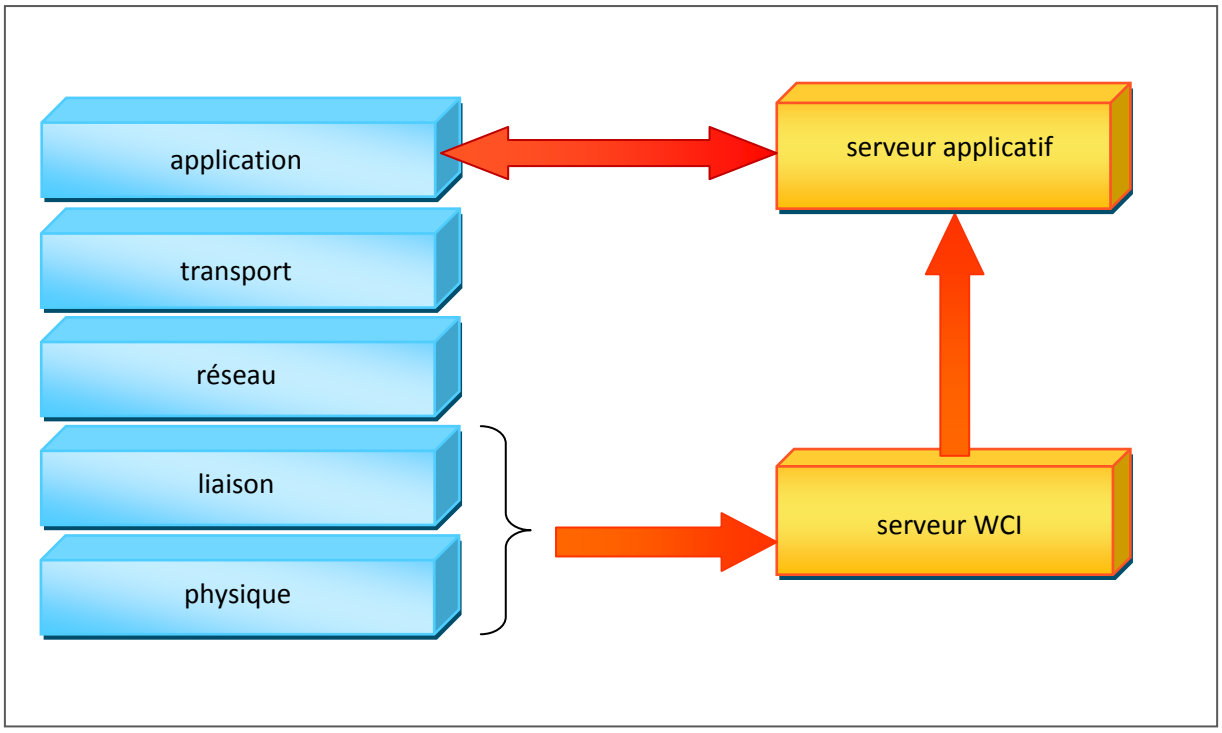


Figure 31 : Network Services

2.4.7. CLIM

Le modèle CLIM (Cross Layer Interaction Model) [67] a pour but l'interopérabilité, le prototypage rapide, le maintien, la portabilité et l'efficacité. Il fournit un cadre générique pour construire et organiser les interactions cross-layer. Il introduit le concept de Network Feature (NF) qui est un service fonctionnel fourni par le réseau. Il peut être le gestionnaire de QdS, le RRM (Radio Resource Management), le CAC (Call Admission Control), etc. C'est une architecture utilisant une entité intermédiaire, le CLME (Cross Layer Manager Entity), pour permettre les interactions inter-couches (c.f. Figure 32).

Ce modèle permet d'effectuer plusieurs opérations à partir des services, telles que leur enregistrement, découverte, demande de valeur de paramètres, ou encore modification de ces paramètres.

Elles fonctionnent selon un concept de client/serveur, avec des requêtes et des réponses. Plusieurs NF communiquent avec le CLME, qui sera leur premier point de contact. Tout d'abord vient l'identification (le nom, l'identifiant réseau et le numéro de port), puis l'enregistrement. C'est la première opération, qui alerte les autres de son existence et qui indique comment les atteindre. Ensuite vient la phase d'information en deux étapes, la requête et la réponse. Ces informations peuvent arriver soit instantanément, soit périodiquement. Le service « commande » permet à un NF

de demander à un autre de modifier ses paramètres. Le service « événement » fournit la notification d'un évènement. Enfin, le service « découverte » se fait en deux étapes. Une entité CLME est désignée dans chaque domaine réseau et reçoit la liste des NF des autres CLME présent sur le chemin choisi. Le modèle CLIM peut se déployer selon 2 modèles, 1 premier 'local' qui permet déjà la communication inter-couche au sein d'un même nœud (cad intégration verticale, avec comme ex d'utilisation les PEPS TCP) , et un second 'global' (intégration horizontale/verticale), dans lequel plusieurs CLME peuvent s 'échanger les données de toutes les couches.

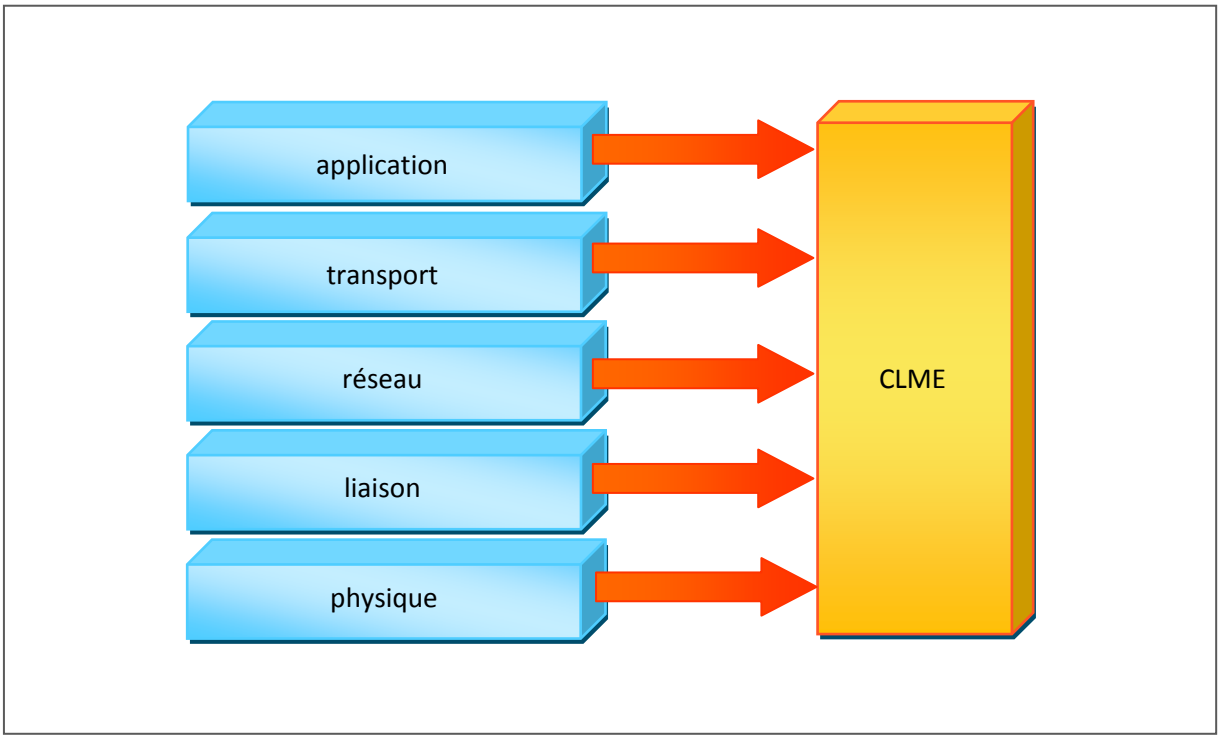


Figure 32 : CLIM

2.5. Conclusion

Un certain nombre d'architectures mettant en œuvre les concepts des communications cross-layer ont été proposées. Cependant, chacune répond à un besoin précis alors qu'il aurait été intéressant de concevoir une architecture modèle. En effet, il y a un risque à long terme d'avoir des systèmes sans fil incompatibles. Il faut toutefois noter qu'il y a eu quelques avancées dans le domaine du Cross-Layer, bien que cette notion soit vue comme une violation des préceptes du modèle en couches, ce qui légitime le fait de tenter d'augmenter les performances des réseaux en partageant des données entre les couches, même non voisines. De plus, il faut noter que la plupart des architectures existantes déploient une entité intermédiaire, ce qui est un bon compromis entre la volonté de garder les atouts du modèle en couche et le besoin d'améliorer la qualité de service des réseaux. Une des principales motivations de notre travail provient de l'indisponibilité ou de l'ultra-spécialisation de ces environnements qui les rendent impossible à utiliser dans notre contexte satellite et sans fil.

3. Interactions Cross-Layer pour l'optimisation de protocoles

3.1. Problématique dans les réseaux satellite DVB-S2 / RCS

Comme nous l'avons vu dans la section 1.2.3, les réseaux satellite DVB-S2/RCS présentent un certain nombre d'avantages, tels qu'une portée à l'échelle continentale, ou encore des mécanismes QdS permettant la gestion par priorité de la voie retour [42].

Cependant, plusieurs problèmes subsistent lorsqu'il s'agit de déployer des applications multimédia interactives. En effet, un délai de bout en bout maximum de 400ms est recommandé pour assurer un fonctionnement satisfaisant des applications multimédia interactives (c.f sections 1.1.1.1, 1.1.2.1), et le délai AIR via satellite est de 250ms. Deux solutions alternatives s'offrent à ces applications.

La première solution consiste à utiliser l'allocation statique pour la transmission des flux multimédia. Avec une quantité de ressources allouées suffisamment grande, ce type d'allocation garantit un délai de traversée du réseau satellite de 270ms, ce qui correspond sensiblement au délai de propagation du lien satellite. Cependant, les services utilisant l'allocation statique, capables de supporter ce type d'applications, sont souvent inaccessibles pour le particulier. En effet, les fournisseurs de service proposent ces services à des tarifs pour la plupart très élevés par rapport au budget de l'utilisateur grand public, du fait d'une consommation élevée des ressources.

La seconde solution est l'accès aux services utilisant l'allocation dynamique. Avec 3 types de requêtes dynamiques de ressources, ce type d'allocation permet d'optimiser l'utilisation des ressources du réseau. Cependant, face aux besoins en QdS des applications multimédia interactives, cette allocation, sans mécanismes adaptés, se révèle souvent inefficace, mal adaptée aux contraintes des applications multimédia.

Nous proposons de détailler cette problématique à travers un scénario exemple, qui servira également de contexte pour la suite des travaux de ce mémoire (c.f. Figure 33).

Un réseau satellite géostationnaire DVB-S2/RCS est utilisé comme réseau d'accès à Internet. Un client est placé dans le réseau satellite, derrière un ST. Il souhaite établir une communication multimédia avec un autre client situé de l'autre côté du réseau satellite, côté GW. Il peut cependant également être situé dans un réseau filaire. Pour établir la communication multimédia, le protocole SIP est utilisé, cependant, tout autre protocole de signalisation de session peut être utilisé.

Un proxy SIP est intégré au réseau satellite. Il peut être situé derrière le ST (donc dans le réseau du client), et est chargé de la gestion des sessions des clients dans le sous-réseau du ST. Il peut aussi être situé du côté GW. Dans ce cas là, il gère les sessions de l'ensemble des clients SIP du réseau satellite. Les options « loose-route » et « record-route » (c.f. section 1.5.2.2) sont utilisées afin de forcer le passage des messages d'initialisation de session par les proxys SIP.

L'allocation dynamique est utilisée pour la transmission des données sur la voie retour. Cependant, nous émettons l'hypothèse que les messages de gestion de sessions (INVITE, OK, ACK, BYE) sont transmis par l'utilisation de l'allocation statique : en effet, ces messages étant de très petite taille (environ 500 octets), ils consomment très peu de ressources.

3.1.1. Allocation dynamique en début de flux multimédia

Une fois l'établissement de la session SIP effectuée, les premières données multimédia du client SIP dans le réseau arrivent dans les tampons d'émission du ST. Ce dernier, afin de pouvoir transmettre le trafic multimédia sur la voie retour du système satellite, doit émettre des requêtes de ressources dynamiques auprès du GW/NCC (temps de propagation de la requête : 250ms). Ce dernier calcule et fournit le plan d'allocation des ressources (temps de propagation de la réponse : 250ms), afin que le ST puisse émettre le trafic multimédia sur la voie retour (temps de propagation des données : 250ms). On remarque alors que les premières données multimédia sont soumises à un délai de transmission au minimum de 750ms pour traverser le système satellite. D'après les recommandations de l'ITU-T, ce délai est incompatible avec les contraintes en Qualité de Service des applications multimédia interactives [3].Un délai maximum de 400ms est recommandé comme service minimum pour les applications multimédia interactives.

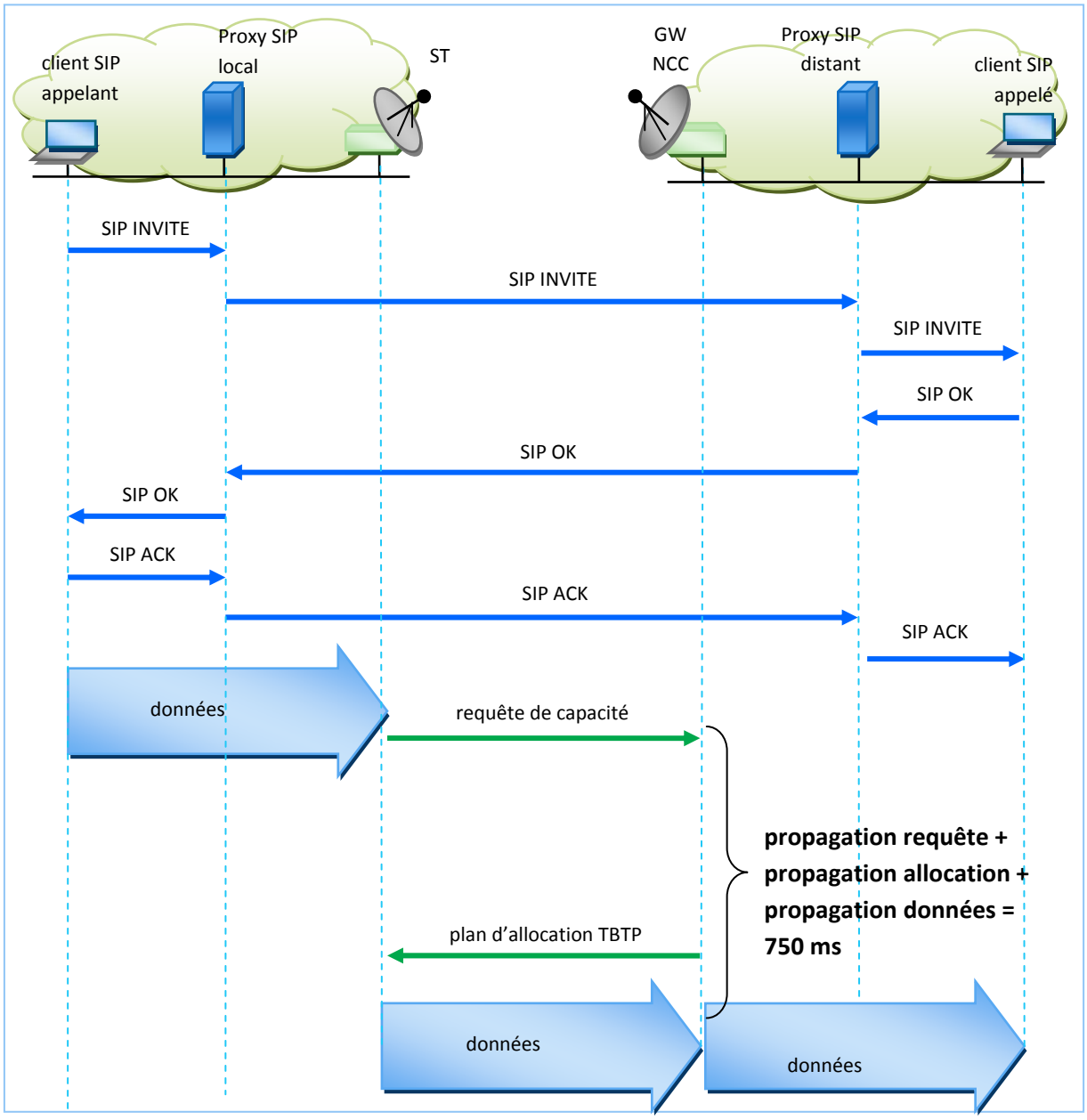


Figure 33 : session multimédia SIP dans réseau satellite avec allocation dynamique

Ce délai semble pouvoir être résorbé par la suite, au cours de la communication, en effectuant d'avantages de requêtes pour écouler les données multimédia encore en attente de ressources dans les tampons d'émission du ST. Or cette « accélération » du flux ne rattrape pas le retard cumulé par les paquets de données multimédia. Les applications multimédia interactives utilisent un flux de données temps-réel : les données sont estampillées et sont censées être lues à une date précise dans le tampon de lecture multimédia du client récepteur afin de fournir à l'utilisateur un message sonore compréhensible. Par conséquent, à la réception, les données arrivées en retard, sont considérées obsolètes et le client récepteur peut : (i) décider de jouer les données obsolètes pour ne pas les perdre. Cependant les données suivantes reçues à la bonne date restent en attente d'être jouées et sont par conséquent aussi retardées: le retard accumulé par les premières données se répercute sur le reste du flux multimédia ; (ii) décider d'écarter, de jeter les données reçues en retard, et de jouer directement les données suivantes reçues à la bonne date. Dans ce cas, une partie du flux multimédia est manquante et le taux de perte de paquets augmente. Dans les deux cas, la qualité de service à fournir à l'application multimédia est dégradée dès le départ.

3.1.2. Allocation dynamique en cours de flux multimédia

Durant la transmission du flux multimédia, la problématique reste sensiblement la même : les données arrivant dans le ST, doivent pouvoir être directement émises sur la voie retour avec un minimum de mise en tampon, afin d'être uniquement soumises au temps de propagation (250ms).

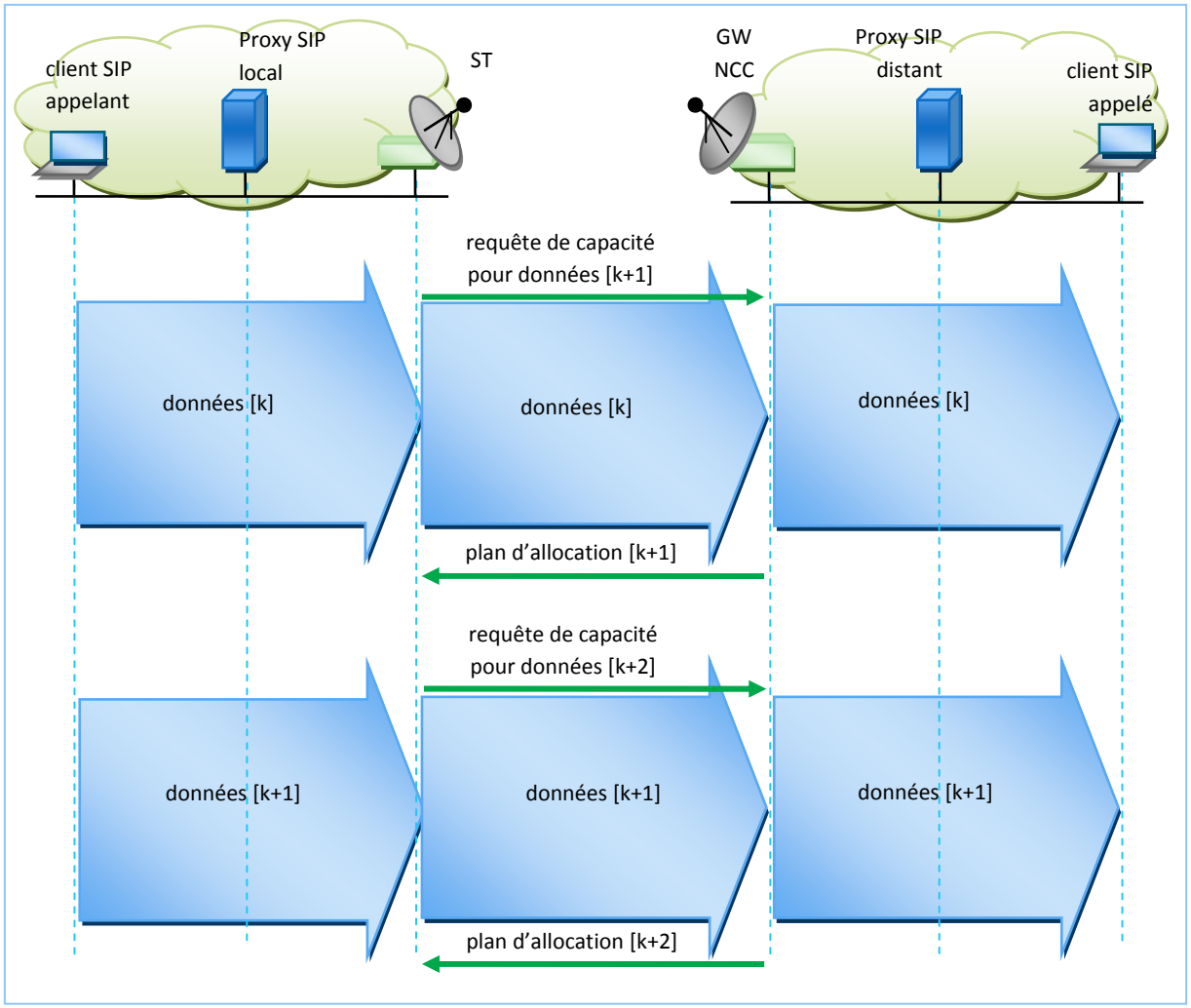


Figure 34 : schéma idéal d'anticipation pour l'allocation dynamique

Cela implique que les ressources nécessaires doivent être anticipées afin d'être disponibles auprès du ST concerné juste au début de la supertrame pour la transmission. De plus, la quantité de ressources anticipées auprès du NCC doit être suffisamment précise afin de ne pas gaspiller les ressources « précieuses » de la voie retour du réseau satellite.

Au final, on rejoint les deux objectifs fixés initialement : (i) fournir un service de qualité aux utilisateurs d'applications multimédia ; (ii) optimiser l'utilisation des ressources du système de communication dans un environnement réseau satellite.

La Figure 34, montre le schéma idéal attendu de l'allocation dynamique de ressources pour émettre les données multimédia sur la voie retour satellite. Pour les données arrivant à l'instant k+1 (supertrame) dans les tampons d'émission du ST, les requêtes de ressources devront être effectuées à l'instant k par le ST. Cet instant est appelé T_{MSL} (Minimum Scheduling Latency), il correspond au cycle de Requête/Allocation (au moins un aller-retour dans le réseau Satellite), soit environ 500ms. Ce scénario est alors possible en utilisant des mécanismes d'anticipation de ressources efficaces. Nous détaillerons par la suite les différents mécanismes d'anticipation envisagés dans ces travaux de thèse.

3.2. Présentation des contributions

Afin de répondre à ces problématiques liées au déploiement des applications multimédia interactives dans les réseaux satellite DVB-S2/RCS avec l'allocation dynamique, nous proposons une architecture à communication cross-layer. Elle est composée de trois sous parties (c.f. Figure 35) :

- **Les communications cross-layer intégrées au sein du système de communication existant afin d'en optimiser les performances ;**
- **Une entité cross-layer dédiée qui permet de faciliter l'intégration de ces nouvelles interfaces cross-layer au sein du système de communication ;**
- **Une base de données contenant les caractéristiques QdS des applications multimédia susceptibles d'être utilisées dans le réseau satellite.**

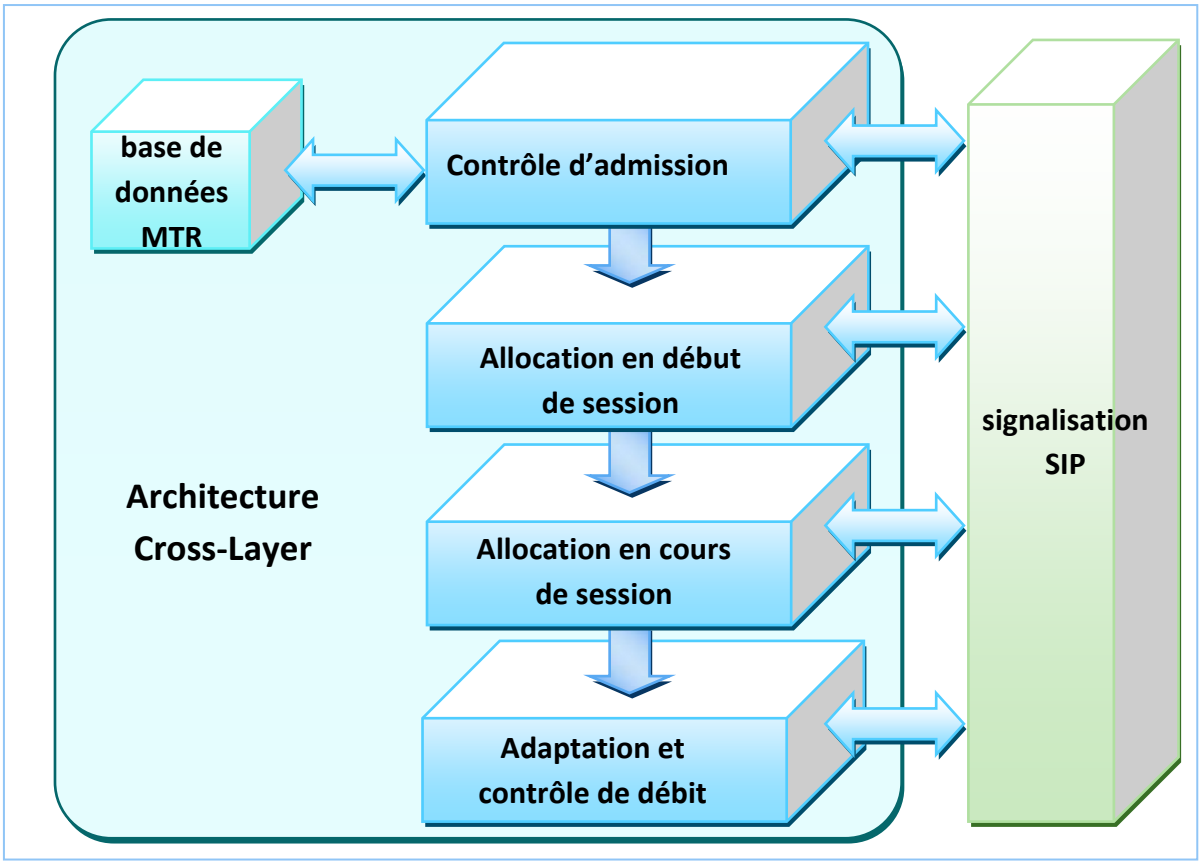


Figure 35 : architecture générale des contributions de thèse

La première contribution basée cross-layer a pour objectif d'améliorer l'allocation dynamique lors du démarrage d'une session multimédia. La seconde intervient durant la session multimédia. La troisième solution basée cross-layer proposée améliore le compromis entre optimisation des ressources réseau et qualité de service des applications multimédia : en adaptant le contrôle d'admission de nouvelles sessions multimédia et le contrôle de débit des sessions en cours, en fonction des conditions de charge du réseau.

Ces trois contributions utilisent un service de caractérisation, appelé service *QoS_Parameters* (présenté en section 5), des besoins en QdS des applications multimédia. À partir du type de média (ex. audio/vidéo) et de son format (nom de codec), il fournit les performances en QdS requises par

l'application sous forme de métriques telles que le délai de bout en bout maximum toléré, la gigue, le taux de perte, la bande passante. Ce service est proposé par une architecture orientée Web Services appelée MTR (Media Type Repository), faisant également partie des travaux de thèse. Dans un souci de lisibilité du mémoire, cette architecture est présentée par la suite, à la section 5.

.

3.3. Amélioration de l'allocation dynamique en début de session multimédia

3.3.1. Présentation de la contribution

Cette contribution [68] propose de réduire le délai initial des requêtes de ressources (c.f. section 3.1.1) subi par les données multimédia, dans un système d'allocation dynamique de ressources de réseau satellite DVB-S2/RCS.

Afin de réduire ce délai, nous proposons d'effectuer une requête anticipée de ressources pour le ST concerné par le flux multimédia afin que le ST dispose des ressources nécessaires à l'émission des données juste avant que ces dernières n'arrivent dans les tampons de transmission de ce ST.

Nous présentons le mécanisme dans un régime non congestionné sur la voie retour, à savoir que la quantité de ressources qui sera demandée au NCC sera accordée.

Pour cela, nous utilisons une approche cross-layer basée sur les informations de session. La Figure 36, montre la communication « cross-layer » directe que nous proposons d'établir entre la couche session proposée par le proxy SIP et la couche liaison (MAC, DVB) du NCC, dans le sens descendant. À partir des informations présentes dans les messages de signalisation de la couche session, nous proposons d'effectuer une requête anticipée de ressources pour le ST concerné par le flux multimédia, vers le NCC. Cette requête étant émise du côté GW/NCC, ceci permet de ne plus être soumis à la latence de la traversée du lien satellite. Le scénario amélioré est expliqué par la suite.

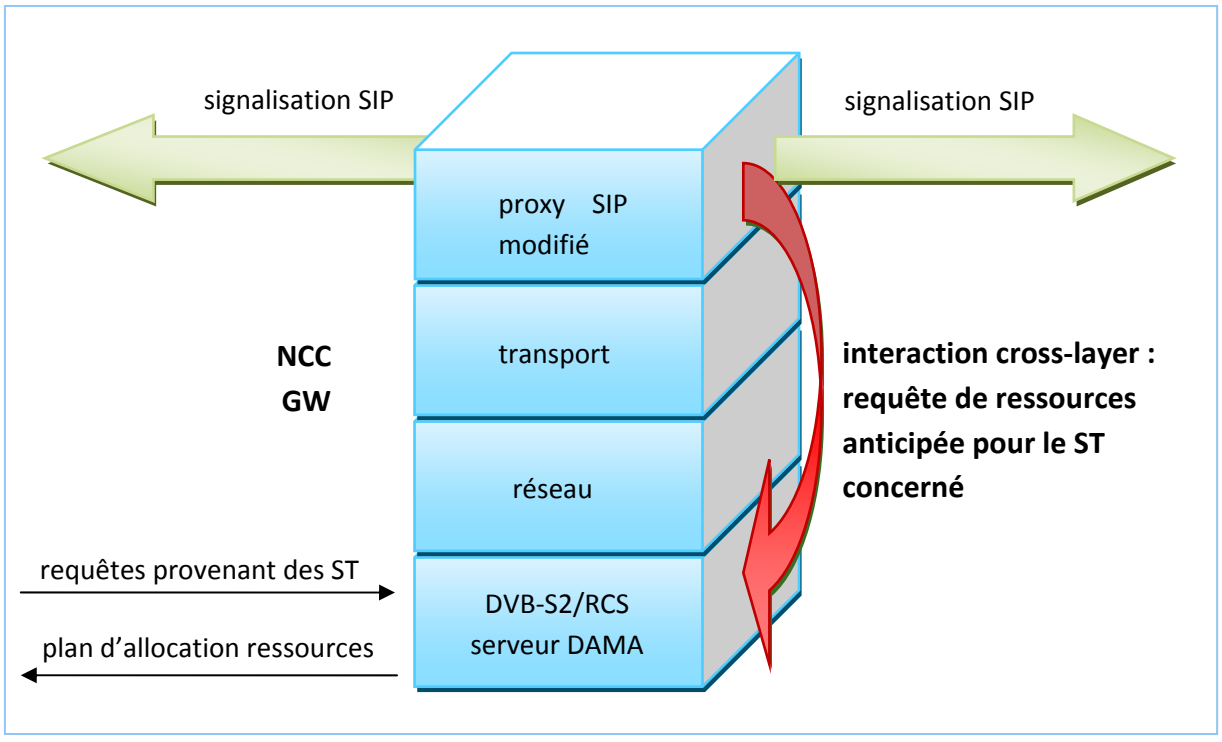


Figure 36 : interaction cross-layer SIP vers DAMA dans le NCC

Le proxy SIP est placé du côté GW/NCC (serveur DAMA). En dehors de l'établissement de la session multimédia, il est en charge d'effectuer la requête anticipée de ressources auprès du NCC. Afin de

recueillir les messages d'initialisation de session pour effectuer cette requête, on utilise les options SIP « loose-route » et « record-route ». Cette option oblige le passage de tous les messages de signalisation de session (établissement, modification, et terminaison de session) par le proxy SIP. Et ceci, même lorsque les clients SIP se connaissent mutuellement (adresses IP et numéro de ports).

La base de données MTR est également placée du côté GW/NCC afin de fournir les paramètres QdS à partir des informations de session recueillies par le proxy SIP.

Ainsi, en cas d'acceptation de l'appel multimédia par le client destinataire, ce dernier envoie le message réponse SIP OK au proxy SIP du domaine satellite. Le proxy SIP modifié en extrait l'adresse IP du client SIP présent dans le réseau satellite. Il en extrait également la liste finale de codecs supportables pour la communication multimédia. Le premier codec de la liste est celui choisi pour la communication. Le proxy SIP utilise le nom du codec choisi pour interroger la base de données MTR. Ce dernier répond en fournissant le débit de transmission associé à ce codec. À partir du débit du codec, le contrôle d'admission du réseau satellite évalue la quantité de ressources réseau (en quantité de cellules ATM) nécessaires par supertrame pour la transmission du flux multimédia sur la voie retour du satellite. Si les ressources requises par ce nouveau flux sont disponibles, une requête DAMA (MAC) est alors construite avec l'identifiant du ST et la quantité de ressources, puis transmise au NCC à travers la communication cross-layer dans le sens descendant. Cette requête cross-layer de ressources peut être effectuée par le proxy SIP modifié, ou tout autre entité placé du côté GW/NCC, avec une interface vers le serveur DAMA du NCC. À partir de notre requête, ainsi que celles provenant des STs, le NCC calcule le plan d'allocation de la voie retour. Le NCC est alors en mesure d'envoyer le plan d'allocation de ressources de la prochaine supertrame et le message SIP OK reçu par le proxy SIP est jusque là retenu par ce dernier.

Afin d'émettre cette requête de ressources dynamiques, deux paramètres sont requis :

- **la quantité de ressources nécessaire** pour cette session, plus particulièrement pour l'émission des premiers paquets de données multimédia (régime transitoire). En effet, nous supposons que l'algorithme du système d'allocation dynamique du ST se chargera, par la suite, de requérir les ressources nécessaires pour tout le reste de la communication multimédia (système en régime stationnaire). On propose que la quantité de ressources soit déduite à partir des informations de session en utilisant la base de données MTR décrite en section 5.
- **l'identifiant du ST** concerné par le flux multimédia afin d'effectuer la requête de ressources à sa place. On propose de se baser sur les informations d'adressage IP contenues dans les messages de session SIP. Le proxy SIP en extrait l'adresse IP du client SIP présent dans le réseau satellite. Sachant que le ST agit comme un routeur pour les postes clients de son sous-réseau, l'adresse IP du routeur en est déduite avec le masque de sous-réseau. Puis une table effectuant la correspondance entre identifiant MAC de chaque ST connecté et leur adresse IP, nous fournit l'identifiant MAC du ST concerné.

Rappelons que le délai à réduire est celui entre la réception des données multimédia dans les tampons d'émission du ST, et l'émission effective de ces données multimédia par le ST :

- la réception des données dans les tampons d'émission du ST correspond à la réception du message SIP OK par le client SIP appelant. En effet, le message SIP OK autorise l'envoi des

données utiles multimédia par le client SIP. De plus, on considère que la transmission des données entre le client SIP et le ST (routeur) est effectuée par une technologie réseau (ex. filaire, Ethernet) moins contraignante en terme de délai par rapport au système satellite. Par conséquent le délai de transmission entre le client SIP et le ST est négligeable ;

- l'émission effective des données multimédia par le ST correspond à la réception du plan d'allocation de ressources de la voie retour par le ST. Ce dernier a ainsi les ressources nécessaires pour émettre directement les données multimédia sur la voie retour du lien satellite.

Autrement dit, le délai à raccourcir est celui entre la réception du message SIP OK par le client SIP et la réception du plan d'allocation de ressources de la voie retour par le ST.

Or la période d'émission du plan d'allocation par le NCC (500ms) est indépendante de la transmission du message SIP OK par le proxy SIP. Le pire cas pouvant se produire est illustré en Figure 37. Le proxy SIP transmet le message SIP OK au client SIP, alors que le NCC n'a pas encore pris en compte la requête cross-layer, et par conséquent n'a pas encore transmis l'allocation au ST concerné. La session multimédia démarre et les paquets multimédia émis par le client SIP restent dans les files d'émission du ST en attendant la prochaine allocation de ressources, soit au maximum 500ms.

Afin de palier à ce problème, nous proposons d'utiliser la communication cross-layer dans l'autre sens (c.f Figure 39). Le proxy SIP bloque le message SIP OK jusqu'à ce que le NCC signale au proxy SIP de l'émission imminente du plan d'allocation de ressources au STs, à travers la communication cross-layer (c.f. Figure 38). Ainsi le proxy SIP libère le message SIP OK pour le transmettre au client SIP, en même temps que le NCC émet le plan d'allocation aux STs. Au niveau des clients SIP, cela prolonge la durée de tonalité d'appel sans affecter le bon fonctionnement de l'établissement de la session.

Le client SIP reçoit le message SIP lui autorisant de transmettre les données multimédia, en même temps que le ST reçoit l'allocation de ressources pour ce flux multimédia. Les données multimédia arrivent dans les tampons d'émission du ST, et sont transmises sur la voie retour du satellite avec le minimum d'attente dans les tampons de transmission.

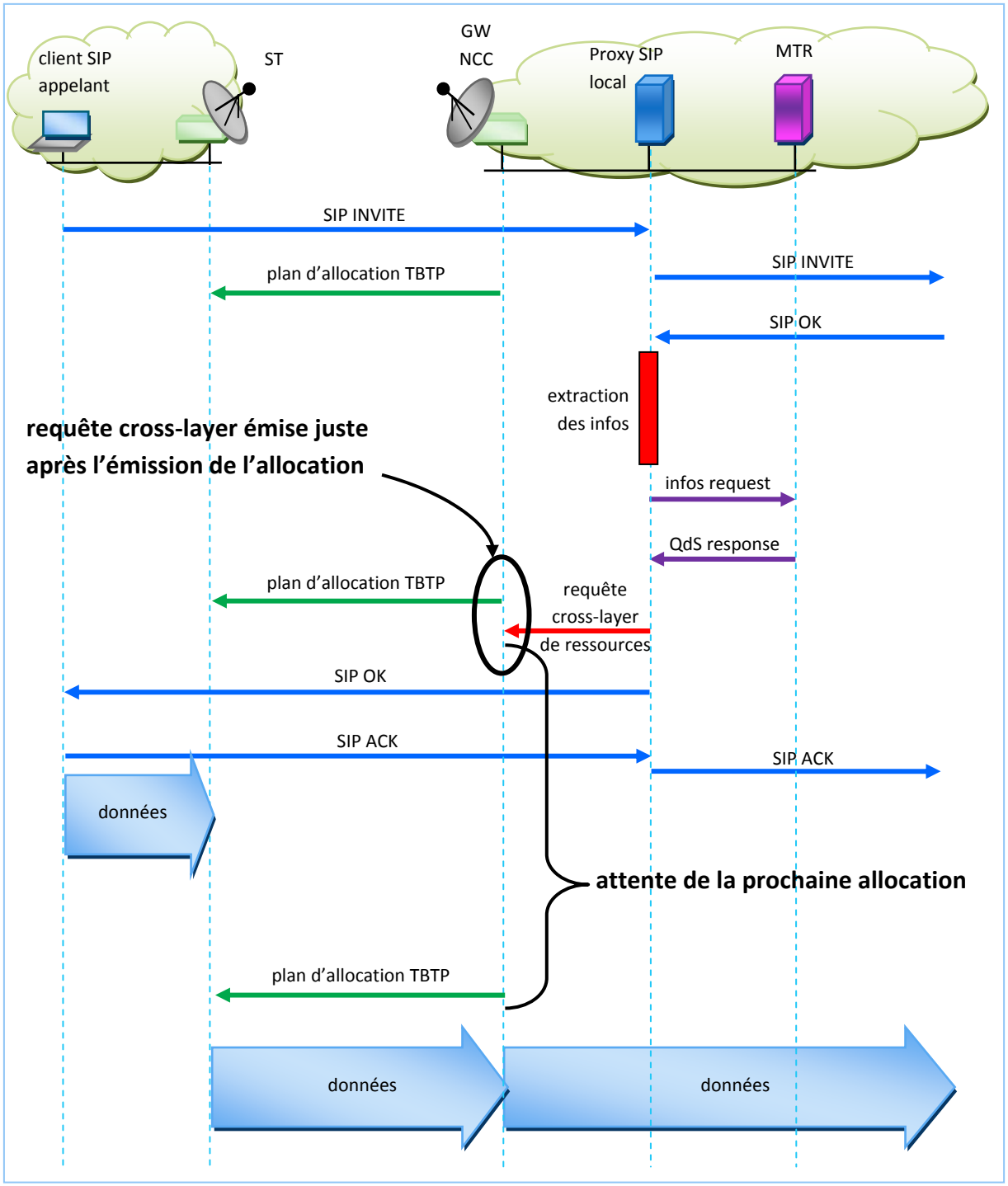


Figure 37 : Pire scénario de requête anticipée de ressources

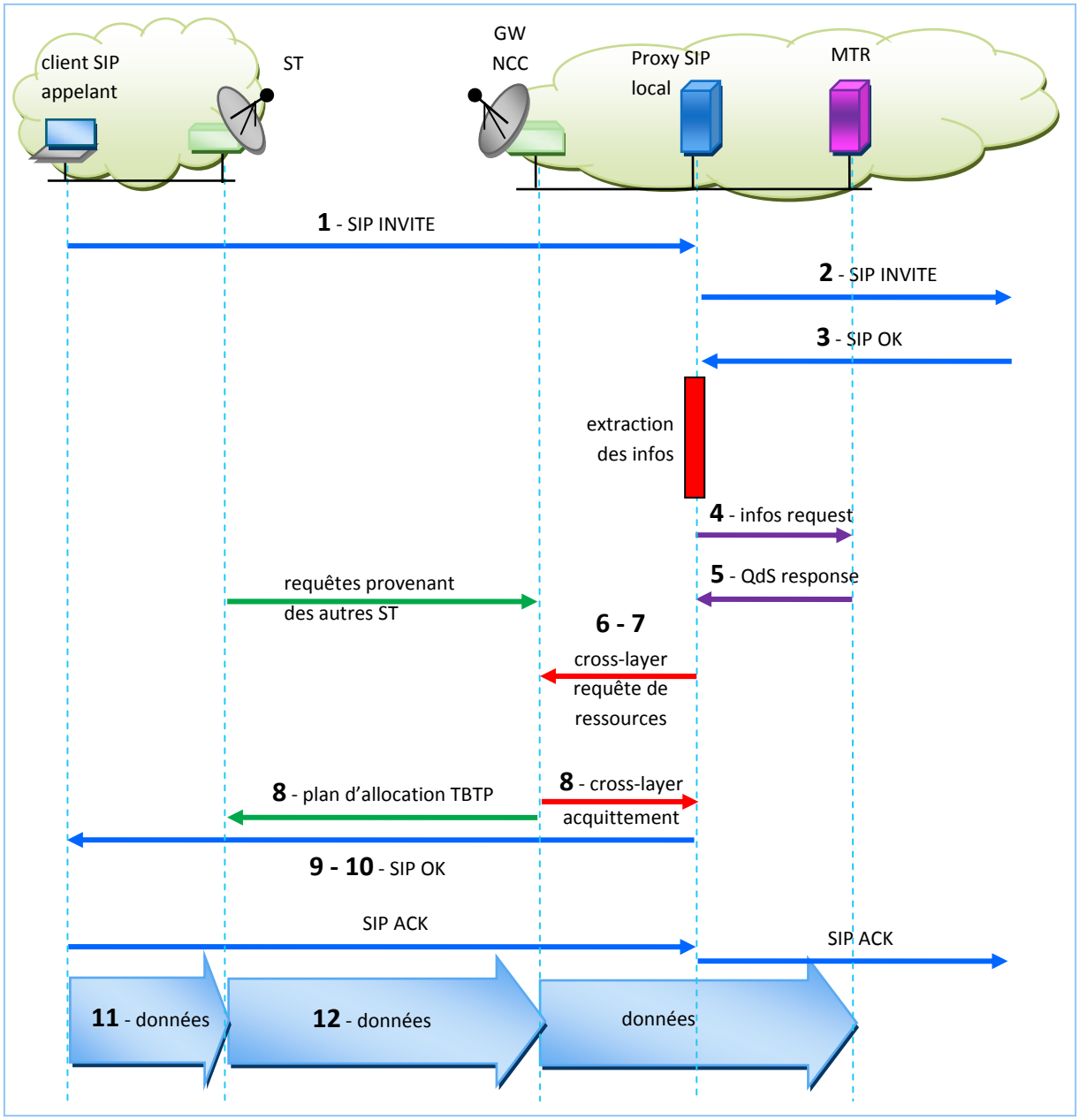


Figure 38 : scénario d'anticipation, cross-layer SIP-DAMA

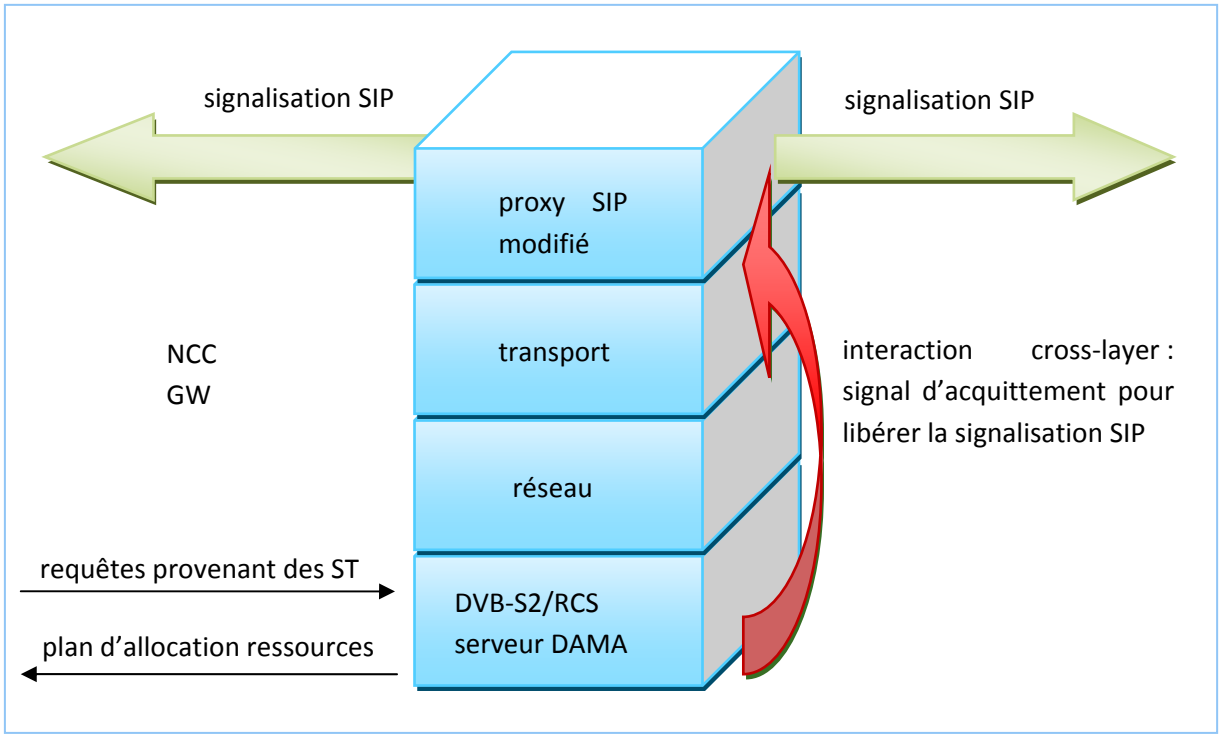


Figure 39 : interaction cross-layer DAMA vers SIP dans le NCC

3.3.2. Conclusion

Le démarrage d'un flux multimédia subit le délai engendré par le cycle de requête/allocation du mécanisme d'allocation dynamique. De part le caractère temps réel et interactif de la communication multimédia, ce délai ne peut pas être rattrapé durant la communication. Afin de palier à ce problème, nous proposons d'effectuer une requête de ressources anticipée, en amont du flux multimédia, pour le ST concerné. Pour cela, nous utilisons les informations de signalisation de session pour obtenir la quantité de ressources nécessaire à l'émission des données, puis nous établissons une communication cross-layer du proxy SIP vers le mécanisme DAMA du NCC afin d'y transmettre une requête de ressources. Puis nous synchronisons le démarrage du flux de données avec l'obtention effective des ressources par le ST. Cette synchronisation est effectuée par une communication cross-layer cette fois-ci du mécanisme DAMA vers le proxy SIP. La synchronisation bloque le message de signalisation SIP OK. Cette attente engendrée, du point de vue de l'utilisateur, est perçue comme le prolongement de la tonalité d'appel, par conséquent, n'a pas d'impact négatif sur la perception de l'utilisateur final.

3.4. Amélioration de l'allocation dynamique en cours de session multimédia

3.4.1. Présentation de la contribution

La seconde contribution [69] [70] concerne toujours l'allocation dynamique de ressources pour les applications multimédia. Cette fois-ci elle a pour objectif de réduire le délai de transmission durant la communication multimédia en régime stationnaire (c.f. section 3.1.2). Pour cela, l'allocation dynamique dans le réseau satellite nécessite un mécanisme d'anticipation de ressources, afin de diminuer l'attente des données dans les files d'émission du ST tout en évitant une sur-allocation des ressources de la voie retour satellite.

Un certain nombre de travaux ont déjà proposé des mécanismes d'anticipation de demande de ressources du DAMA [71] [72] [73] [74]. Notre contribution se base sur la proposition [75] qui a été retenue par l'architecture de QdS SatSix. L'estimation de la capacité requise $r_{REQ}[k]$ par un ST à la k^{ième} supertrame est exprimée à partir de [11] selon la formule suivante :

$$r_{REQ}[k] = \left[r_{IN}[k] + max\left\{0, \frac{1}{2}\left[q[k] - \left(r_{IN}[k] + (\propto -1)r_{IN}[k-1] + (\propto -1)r_{IN}[k-2] + r_{REQ}[k-1] + r_{REQ}[k-2]\right)\right]\right\}\right]$$

Où $r_{REQ}[k]$ est la quantité de ressources requise à la k^{ième} supertrame, $r_{IN}[k]$est la quantité de données reçues dans la file d'attente durant la (k-1)^{ième} supertrame, $q[k]$ est la taille de la file d'attente mesurée au début de la k^{ième} supertrame et $\propto$ un coefficient de pondération.

Le premier terme de l'équation correspond aux ressources requises pour écouler les données qui sont entrées dans la file d'attente durant la supertrame précédente.

Le deuxième terme de l'équation est proportionnel à la taille actuelle de la file d'attente à laquelle est retranchée la somme des requêtes encore en attente de réponse. Ce terme fournit également une anticipation classique pour les besoins à venir à la supertrame $k+T_{MSL}$. Il se base sur la quantité de données reçues durant les 2 supertrames précédentes, [k-1] et [k-2]. Il tient donc compte de l'état passé de la file d'attente.

Afin d'affiner cette anticipation, un facteur $\propto$ de pondération est introduit dans le deuxième terme de l'équation. Constant et compris entre 0 et 1, ce facteur $\propto$ permet de pondérer la quantité de données rentrées précédemment dans la file d'attente servant pour les futures demandes de ressources.

Seulement, comme on peut s'y attendre, en diminuant le paramètre $\propto$, on augmente le volume de ressources demandé dans les requêtes de capacité pour les futures consommations. Cependant, si le volume de ressources demandé n'est pas finalement consommé, on perd ces ressources non utilisées. Ce qui mène au problème de la sur-allocation (sous-utilisation) du lien satellite. Et réciproquement, en augmentant $\propto$, on diminue le volume de ressources requis pour les futures besoins. Si ce volume n'est pas suffisant pour écouler la file de transmission, les données non transmises devront attendre les prochaines ressources, donc subiront alors le délai du cycle requête/allocation.

Le paramètre ∝, tel que défini dans l'architecture SatIP6, est fixé par l'opérateur réseau et reste constant tout le long du service fourni. Cette constance place l'opérateur face au dilemme de l'optimisation de l'allocation, et donc de la rentabilité du réseau, avec la diminution de la qualité de service et donc de la valeur ajoutée. Ainsi ce paramètre α s'avère très délicat à configurer.

Nous proposons, dans le cas des applications multimédia avec signalisation (ex. SIP), de faire évoluer dynamiquement ce facteur α en fonction de l'évolution du trafic multimédia. En rendant le facteur d'anticipation dynamique, l'objectif est d'obtenir une forte anticipation quand la variabilité du trafic multimédia est élevée, c'est-à-dire quand le DAMA peut-être mis en défaut, et une faible anticipation (donc un fort taux d'utilisation) quand le trafic multimédia est stable. Ainsi nous réduisons le délai d'attente des données utilisateur dans les tampons d'émission du ST tout en garantissant une bonne utilisation des ressources de la voie retour satellite.

Pour cela, le facteur d'anticipation ∝ est calculé à partir du débit d'arrivée des paquets (d_I) mesuré sur la file d'émission du ST dédiée aux flux multimédia signalés, et aux caractéristiques annoncées du trafic multimédia annoncées par la signalisation de session SIP. Ainsi durant la communication multimédia, lorsque le débit d'arrivée mesuré est largement en dessous du débit annoncé, nous supposons que cet état est une phase transitoire, et pour anticiper le retour en régime stationnaire, nous augmentons la quantité de ressources requises pour la prochaine supertrame. Et inversement, lorsque le débit d'arrivée mesuré avoisine le débit annoncé, nous sommes en régime stationnaire, donc le facteur d'anticipation limite la quantité de ressources anticipées.

Ainsi à partir du débit d'arrivée des paquets (d_I) dans le ST et connaissant le débit annoncé (d_C) des applications multimédia, nous calculons le facteur dynamique α à travers le rapport :

$$\propto = \frac{d_I}{\sum d_C}$$

Ce calcul est effectué avant chaque requête de capacité donc avec la même période que celle d'une supertrame, cette dernière étant de 500ms.

Pour obtenir le débit d'arrivée d_I, nous utilisons dans un premier temps des mécanismes d'architecture DiffServ afin d'isoler dans une file d'émission dédiée les flux multimédia avec signalisation tel que SIP, des autres flux (ex. navigation web, transfert de fichier, etc.). Afin de garantir que ce rapport soit compris entre 0 et 1, nous utilisons les mécanismes de mis en forme de flux de DiffServ (ex. *dual token bucket*) à l'entrée des files d'émission du ST. Le débit d'arrivée d_I des paquets est alors mesuré en entrée de la classe de service, à partir d'une fenêtre glissante de même période que l'émission des requêtes de ressources.

Pour obtenir le débit d_C annoncé par le flux signalé, nous mettons en place, comme pour la contribution précédente une communication cross-layer, mais cette fois-ci au sein du ST, telle que présentée dans la Figure 40. Cette communication cross-layer permet de recueillir les informations de sessions, comme le débit du trafic multimédia annoncé, pour les fournir au client DAMA du ST, au niveau MAC, chargé de calculer la quantité de ressources nécessaires pour l'émission des données sur la voie retour satellite.

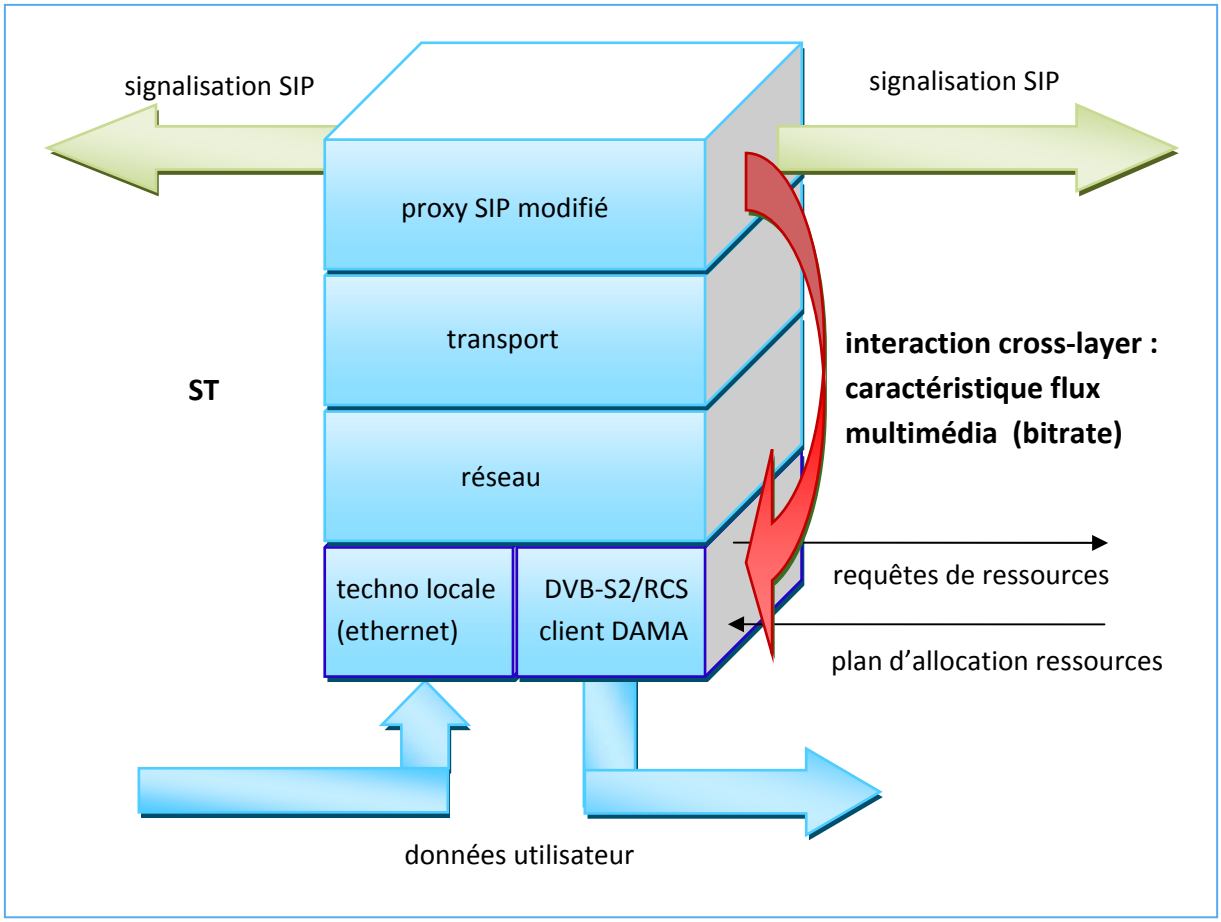


Figure 40 : interaction cross-layer SIP-DAMA dans le ST

Pour cela, le proxy SIP et la base de données MTR sont placés du côté ST. Dans ce scénario, ces deux entités sont distribuées dans chaque réseau utilisateur de ST. Cette topologie facilite la mise en place de la communication cross-layer avec le client DAMA du ST. Cependant, l'utilisation combinée des scénarii distribués (ici présent) et centralisés (proxy SIP et base MTR placés côté GW/NCC) est tout à fait possible : pour les clients SIP à l'intérieur du réseau satellite, c'est leur proxy SIP derrière chaque ST qui est reconnu, alors que pour l'extérieur, c'est le proxy côté GW/NCC qui est connu comme étant le proxy du réseau satellite. Ensuite les deux proxys SIP se contentent de relayer la signalisation SIP entre eux.

Comme le montre la Figure 41, durant la phase d'établissement de session, en cas d'acceptation de l'appel multimédia par le client destinataire, ce dernier envoie le message réponse SIP OK au proxy SIP côté ST. Le proxy SIP modifié en extrait la liste finale de codecs supportables pour la communication multimédia. Le premier codec de la liste est celui choisi pour la communication. Le proxy SIP utilise le nom du codec choisi pour interroger la base de données MTR qui est placée du côté ST afin de fournir les paramètres QdS. Ce dernier répond en fournissant le débit de transmission associé à ce codec. Le débit annoncé du flux multimédia est alors fourni par communication cross-layer au client DAMA du ST.

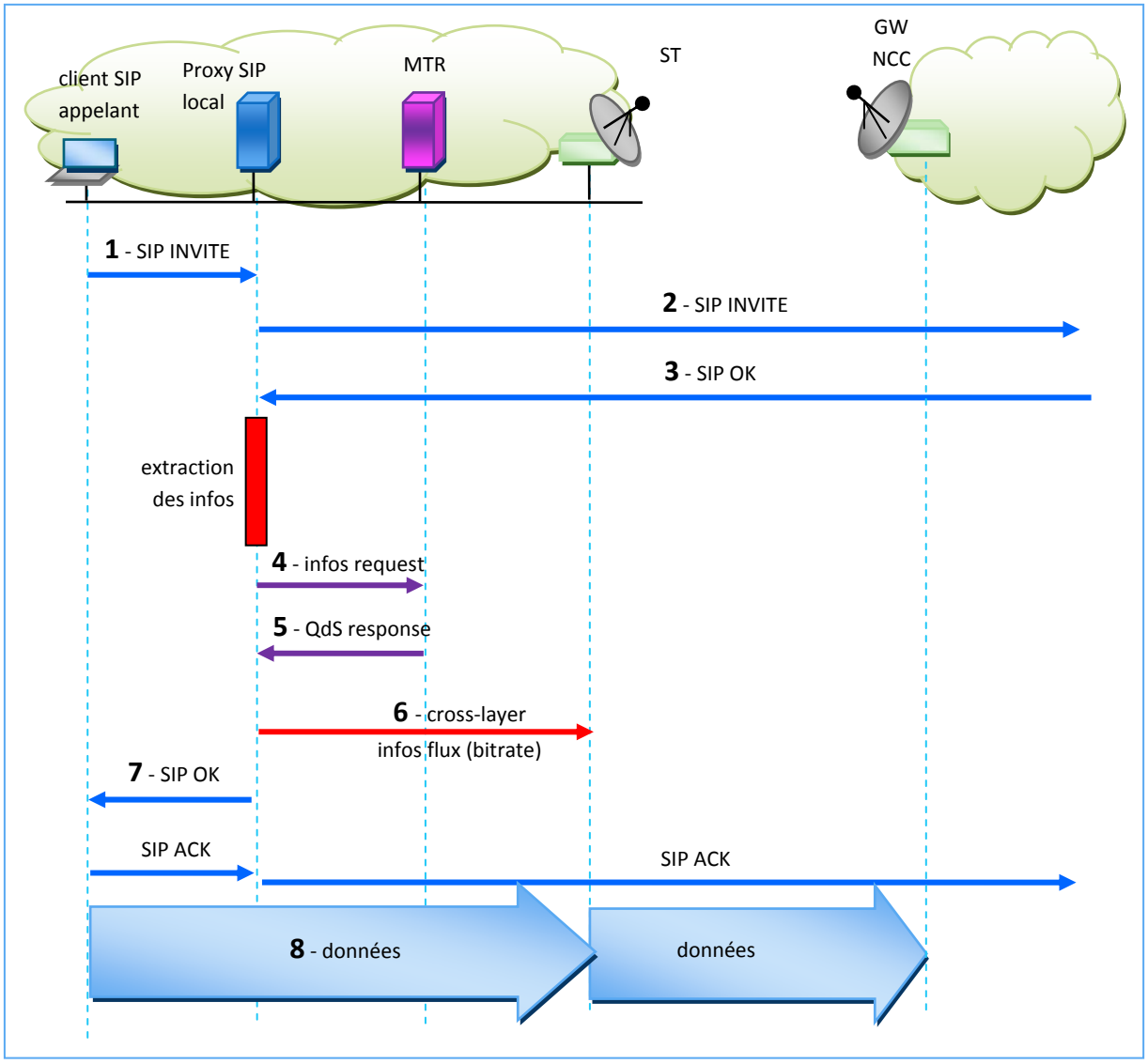


Figure 41 : scénario de cross-layer SIP-DAMA client

Une fois que le flux de données multimédia démarre, le mécanisme de requêtes dynamiques de ressources se met en place dans le ST et le client DAMA est en mesure d'effectuer des requêtes de ressources suffisamment précises pour limiter le délai d'attente des données multimédia dans les tampons de transmission du ST, tout en optimisant les ressources disponibles sur la voie retour du système satellite.

3.4.2. Conclusion

Dans cette contribution, nous avons proposé de réduire le délai de transmission des données multimédia tout en utilisant de manière efficace les ressources réseau satellite. Pour cela, nous paramétrons dynamiquement le mécanisme de requêtes d'allocation de ressources du ST concerné (client DAMA) à l'aide d'une interaction cr0ss-layer session-MAC intégré au sein du ST. Cette interaction fournit les informations nécessaires à cette configuration dynamique, telles que le débit de transmission annoncé par l'application multimédia.

3.5. Contrôle d'admission et de débit pour session multimédia

3.5.1. Présentation de la contribution

Nous avons précédemment proposés deux solutions [76] [77] basées sur les interactions cross-layer (SIP-MAC) permettant la transmission de flux multimédia interactifs en utilisant l'allocation dynamique de ressources dans un système satellite DVB-S2/RCS (c.f. section 1.2.3.2). En effet, jusque là, les services capables de répondre aux besoins de ce type d'applications étaient souvent inaccessibles aux utilisateurs grand public dû essentiellement à une allocation statique. Les fournisseurs de service proposent ces services à des tarifs pour la plupart très élevé par rapport au budget de l'utilisateur grand public. Par conséquent le nombre d'utilisateurs ayant les moyens financiers de s'octroyer un service acceptable pour une communication multimédia à travers un lien satellite restent faible. Face à ce constat, les ressources non utilisées par ces services de classe supérieure sont finalement allouées au service le plus accessible, le moins cher, le service Best Effort. Ce dernier représente la majeure partie de l'allocation totale de ressources du lien satellite. Cependant, le service Best-Effort n'est pas capable de fournir de garantie sur le délai et encore moins sur la gigue. Ces deux métriques sont capitales pour les applications multimédia.

L'objectif est donc de mettre en place une classe de service et des mécanismes QdS de contrôle d'admission et de contrôle de débit dédiés aux applications multimédia avec signalisation. Cette classe doit être à la fois capable de :

- **supporter une quantité élevée de clients ;**
- **garantir une QdS aux applications multimédia interactives.**

Ce service qui, jusque là était à des coûts élevés, verrait ses tarifs diminuer afin d'être accessible à un maximum d'utilisateurs à revenu « moyen ».

Ce service, en ordre de priorité, serait entre le service supérieur « Premium » et le service Best-Effort. Il utilise une allocation dynamique (RBDC : Rate Based Dynamic Capacity) qui est prise en compte juste après l'allocation statique.

Afin de répondre à ce besoin, on propose, dans un premier temps, d'intégrer un système de contrôle d'admission dans le réseau satellite.

3.5.2. Positionnement du problème de contrôle d'admission dans un réseau satellite DVB-S2/RCS

Le contrôle d'admission permet de garantir la qualité de service fournie aux connexions multimédia admises dans le réseau satellite. Dans notre cas, il se base sur les informations recueillies dans la signalisation d'établissement de session permettant ainsi de spécifier les besoins en QdS de la communication multimédia à venir. Nous sommes ainsi capables de prendre une décision d'admission avec support de QdS ou pas pour le flux multimédia. Pour cela, les applications multimédia interactives concernées par cette contribution, nécessitent l'utilisation d'un protocole de signalisation de session. Dans le cadre de ces travaux, le protocole SIP est choisi ; cependant la proposition ici présentée est en mesure de fonctionner avec tout autre protocole de signalisation de session.

La proposition de contrôle d'admission basée sur la signalisation de session SIP est décrite sur la Figure 42. Nous plaçons du côté GW/NCC un proxy SIP chargé des communications multimédia, un serveur MTR chargé de fournir les spécifications en QdS, et une entité chargée du contrôle d'admission des sessions multimédia.

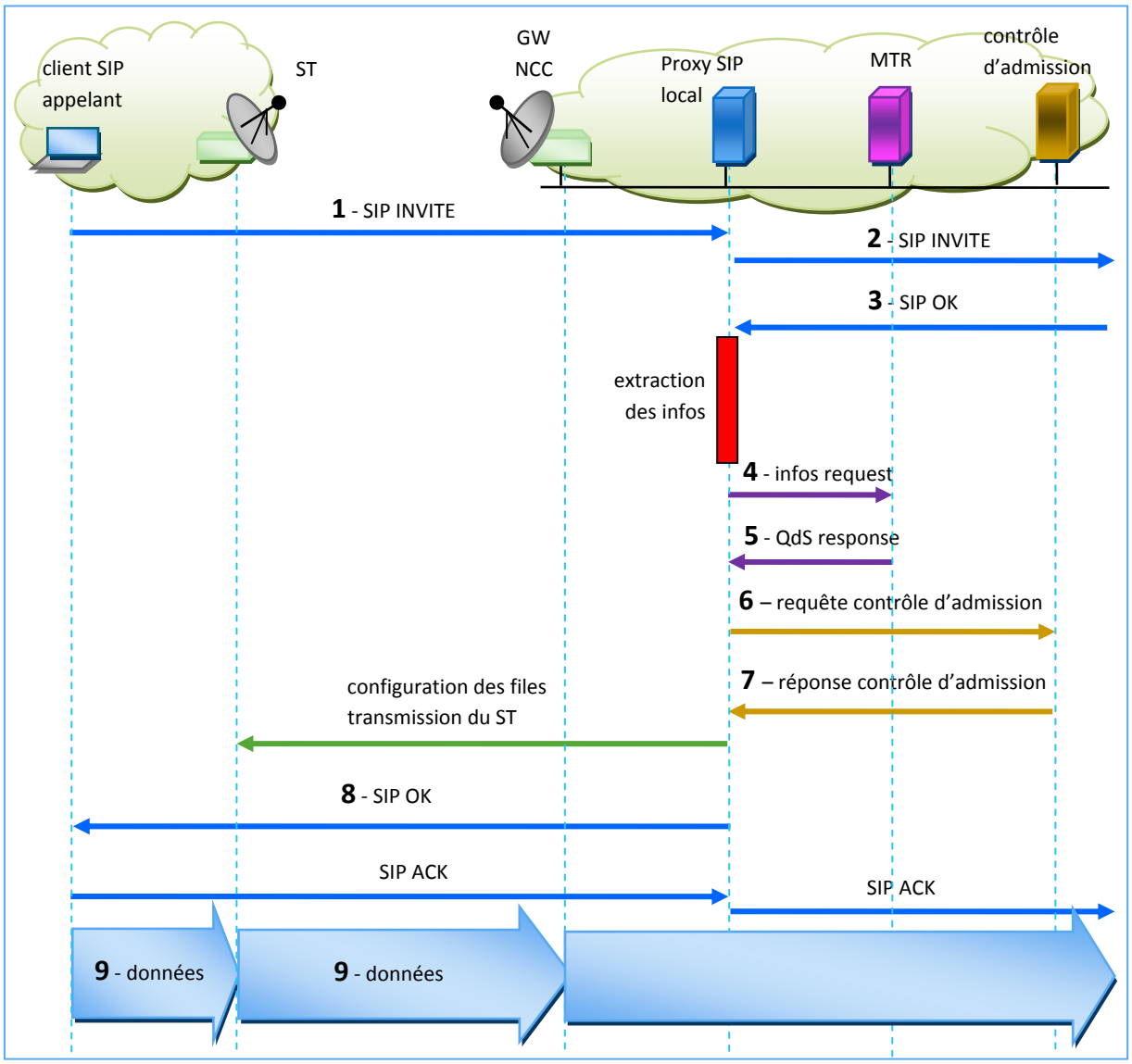


Figure 42 : session SIP avec contrôle d'admission dans un réseau satellite

Jusque là, le processus est le même que celui présenté dans les deux dernières contributions : en cas d'acceptation de l'appel multimédia par le client destinataire, ce dernier envoie le message réponse SIP OK au proxy SIP du domaine satellite. Le proxy SIP modifié en extrait les identifiants des clients (adresses SIP (ex. nivor@laas.fr), adresses IP et numéros de ports source-destination). Il en extrait également la liste finale de codecs supportables pour la communication multimédia. Le premier codec de la liste est celui choisi pour la communication. Le proxy SIP utilise le nom du codec choisi pour interroger la base de données MTR. Ce dernier répond en fournissant le débit de transmission associé à ce codec. Les identifiants des clients sont utilisés pour valider leur authentification avec les

contrats de QdS (ex. SLA). À partir du débit du codec, le contrôle d'admission du réseau satellite vérifie l'admissibilité du flux pour la classe de service avec support de QdS.

Si le débit de transmission du codec du nouveau flux à admettre, ajouté à la somme des débits de flux actuellement présents dans la classe est :

a. inférieur à la capacité totale allouée à la classe, le flux est alors accepté dans la classe de service. Les mécanismes de réservation de ressources sont mis en place (dans le ST et le NCC) afin d'assurer un support de QdS au flux multimédia. Une signalisation permet de configurer les files de transmission du ST.
b. supérieur à la capacité totale allouée à la classe avec support QdS, la communication est alors établie, mais sans support de QdS.

Lorsque le flux de données multimédia arrive dans le ST pour être transmis sur le réseau satellite, il est dirigé vers la classe avec ou sans support de QdS, selon la décision précédente du contrôle d'admission.

Ce scénario de base peut être amélioré. En effet, lors de la phase d'établissement de la session multimédia, le client SIP appelant offre une liste contenant souvent plusieurs codecs capable d'être supportés durant la communication. Le client appelé reçoit cette liste et la croise avec la sienne, afin d'obtenir une liste finale commune de codecs capables d'être supportés par les deux clients. Le client SIP appelé n'étant pas conscient de la QdS, se contente de choisir arbitrairement le premier codec de la liste pour la communication multimédia. Si le débit de transmission de ce premier codec dans la liste est trop élevé par rapport aux ressources disponibles dans le réseau, il ne sera pas admis par le contrôle d'admission dans une classe de service avec support de QdS.

Or, en choisissant un autre codec commun dans cette liste ayant un débit de transmission plus faible, la communication multimédia aurait été acceptée par le contrôle d'admission pour la classe de service avec un support QdS adapté au flux multimédia. Le débit étant plus faible, la qualité du média est également plus faible. Cependant, les contraintes de délai de bout en bout et de gigue qui sont primordiales pour les flux temps-réel sont assurées.

3.5.3. Amélioration du contrôle d'admission basé sur un scénario SIP

Pour résoudre ce problème, on propose d'apporter une amélioration à la phase de contrôle d'admission durant l'établissement de la session SIP dans le réseau satellite. Lorsque le proxy SIP du réseau satellite reçoit la liste de codecs que peut supporter la communication, on propose, comme le montre la Figure 43, de réorganiser cette liste de codecs selon les conditions d'admissibilité dans le réseau satellite. Cette amélioration apportée est complètement transparente pour les clients SIP.

Lorsque le proxy SIP reçoit le premier message SIP INVITE, il en extrait la liste de codecs et sauvegarde cette liste initiale dans une base de données. Puis pour chaque codec présent dans la liste, il interroge la base MTR pour obtenir leur débit de transmission associé. **Le débit de transmission associé à chaque codec de la liste est alors envoyé au contrôle d'admission pour vérifier leur admissibilité pour la classe avec support de QdS. Les codecs dont le débit ne pourra pas être supporté par le contrôle d'admission sont supprimés de la liste. Une nouvelle liste est**

ainsi obtenue, qui contient uniquement les codecs ayant reçu une réponse positive d'admission par le contrôle.

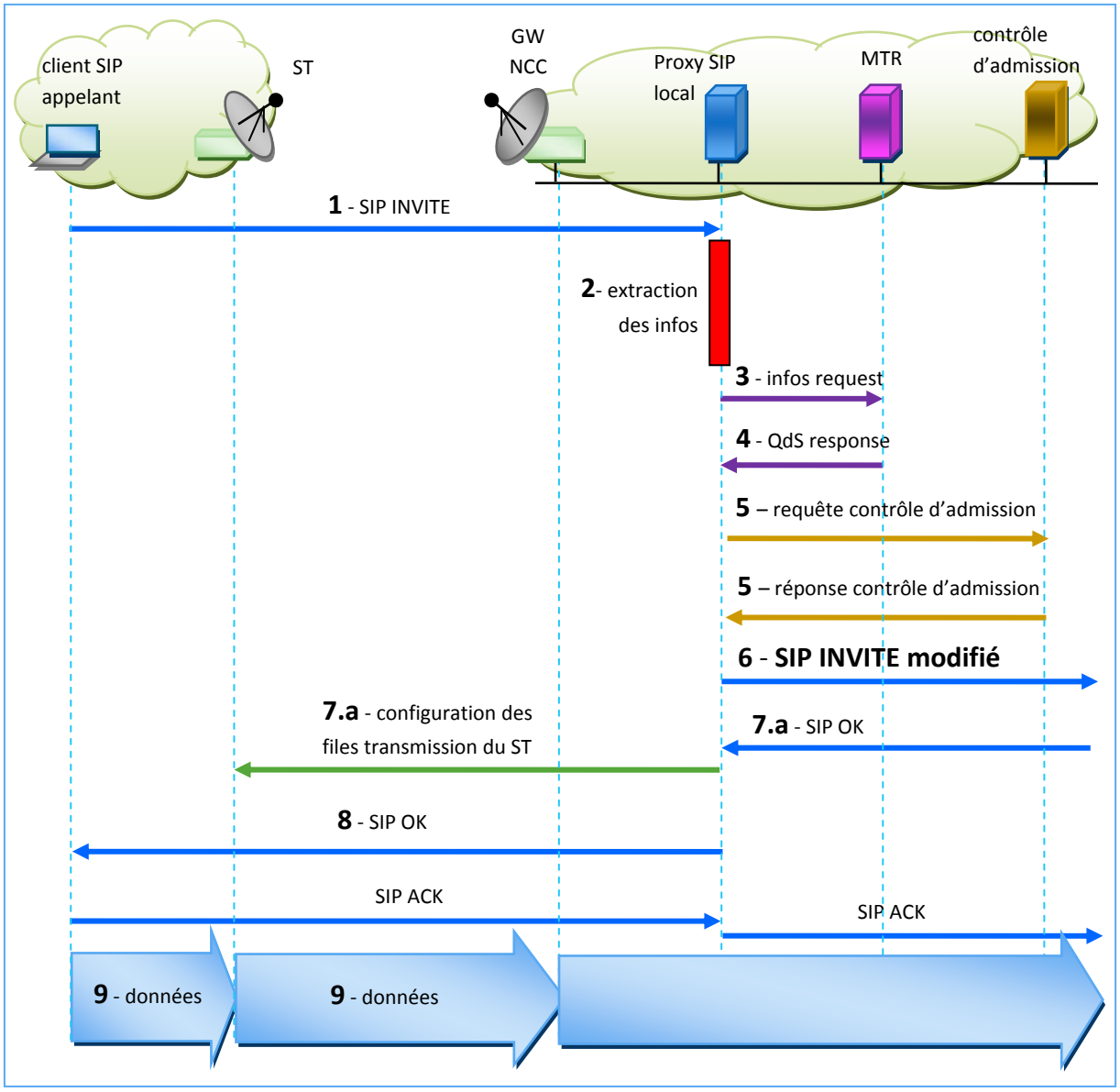


Figure 43 : session SIP acceptée avec contrôle d'admission amélioré

Il est primordial d'effectuer ce contrôle d'admission préalable sur le message SIP INVITE, avant qu'il n'arrive au client SIP destinataire. Car une fois que ce dernier répond avec le message SIP OK, la liste de codecs utilisés pour la communication est alors fixée. Si le client destinataire choisit arbitrairement un codec à débit élevé, le contrôle d'admission, en cas de ressources limitées, serait dans l'obligation de refuser le support QdS pour cette communication.

Le proxy SIP intègre cette nouvelle liste de codec dans le message SIP INVITE en remplacement de celle initialement fournie par le client SIP appelant. Puis il le transmet au domaine distant (proxy SIP distant puis client SIP appelé).

Lorsque le client SIP distant (appelé) reçoit le message SIP INVITE et consulte la nouvelle liste de codecs proposée, deux cas de figure peuvent se présenter :

1. **Le client SIP distant supporte au moins l'un des codecs proposés dans la liste modifiée.** Il accepte la communication multimédia en répondant avec le message SIP OK. Ce dernier contient la liste avec au moins un codec commun choisi pour la communication. Le proxy SIP reçoit ce message OK et lance une réservation de ressource pour le débit du codec choisi. En effet, **ce codec a déjà passé le contrôle d'admission en amont**. Ainsi la session multimédia démarre et son flux de données arrivant dans le ST et est dirigé vers la classe de service avec support de QdS.
2. **Le client SIP distant ne supporte aucun des codecs proposés dans la liste modifiée.** Il renvoie alors un message de refus de communication (415 Unsupported Media Type) jusqu'au proxy SIP du réseau satellite. Cette incompatibilité est peut être causée par la diminution du nombre de codecs, due à la suppression de codecs durant la phase de contrôle d'admission en amont. Dans tous les cas, cela signifie que **les codecs compatibles entre les deux clients terminaux ne sont pas admissibles dans le réseau satellite pour un support QdS**. Face à ce cas de figure, il est possible de :
    a. Accepter que la communication soit établie sans support de QdS. Ainsi, lorsque le proxy SIP reçoit le message SIP de refus de communication, on propose de rejeter ce message (c.f. Figure 44). Puis le proxy SIP retransmet un nouveau message SIP INVITE au client appelé (de manière transparente pour le client appelant). Cette fois-ci, le proxy SIP intègre dans le message SIP INVITE la liste initiale de codecs (sauvegardée) proposée par le client appelant. Ainsi le prochain refus de communication ne sera pas causé par notre contribution. Après cela, si la communication est établie, le flux multimédia sera dirigé vers la classe Best-Effort et ne recevra pas de support QdS.
    b. Empêcher l'établissement de la communication multimédia sans support QdS. Par conséquent, le proxy SIP transmet le message de rejet de communication au client SIP appelant. Et la communication ne démarre pas.

3.5.4. Adaptation de débit basée sur l'architecture SIP

Il est toujours possible d'accepter davantage de sessions multimédia dans le réseau en adaptant le débit de transmission des applications multimédia en fonction de la charge du réseau satellite. Pour cela, on propose d'apporter une autre amélioration à notre système de contrôle d'admission en s'appuyant sur un contrôle de débit des flux multimédia présents dans la classe de service dédiée.

- Lorsqu'une nouvelle connexion multimédia souhaite démarrer avec support de QdS, et que les ressources disponibles sont insuffisantes, on propose de diminuer la charge de la classe à QdS en diminuant le débit de transmission des autres connexions actuellement en cours. Pour cela, on change leur codec actuel en des codecs de plus faible qualité, requérant moins de bande passante. Ainsi des ressources réseau sont libérées. Ces dernières sont alors utilisées pour admettre et supporter la nouvelle connexion multimédia tout en garantissant une qualité minimum aux autres flux. Ainsi on satisfait le premier critère : **supporter une capacité élevée de clients**.
- Lorsque l'une des connexions multimédia se termine, on propose de redistribuer ses ressources réseau aux autres connexions multimédia toujours actives. Leurs codecs actuels sont alors changés pour des codecs à plus grand débit, leur garantissant un média (audio et/ou vidéo) de meilleure qualité, et à fortiori un meilleur service. Ceci qui garantit le deuxième critère : **fournir un service de qualité aux applications multimédia interactives**.

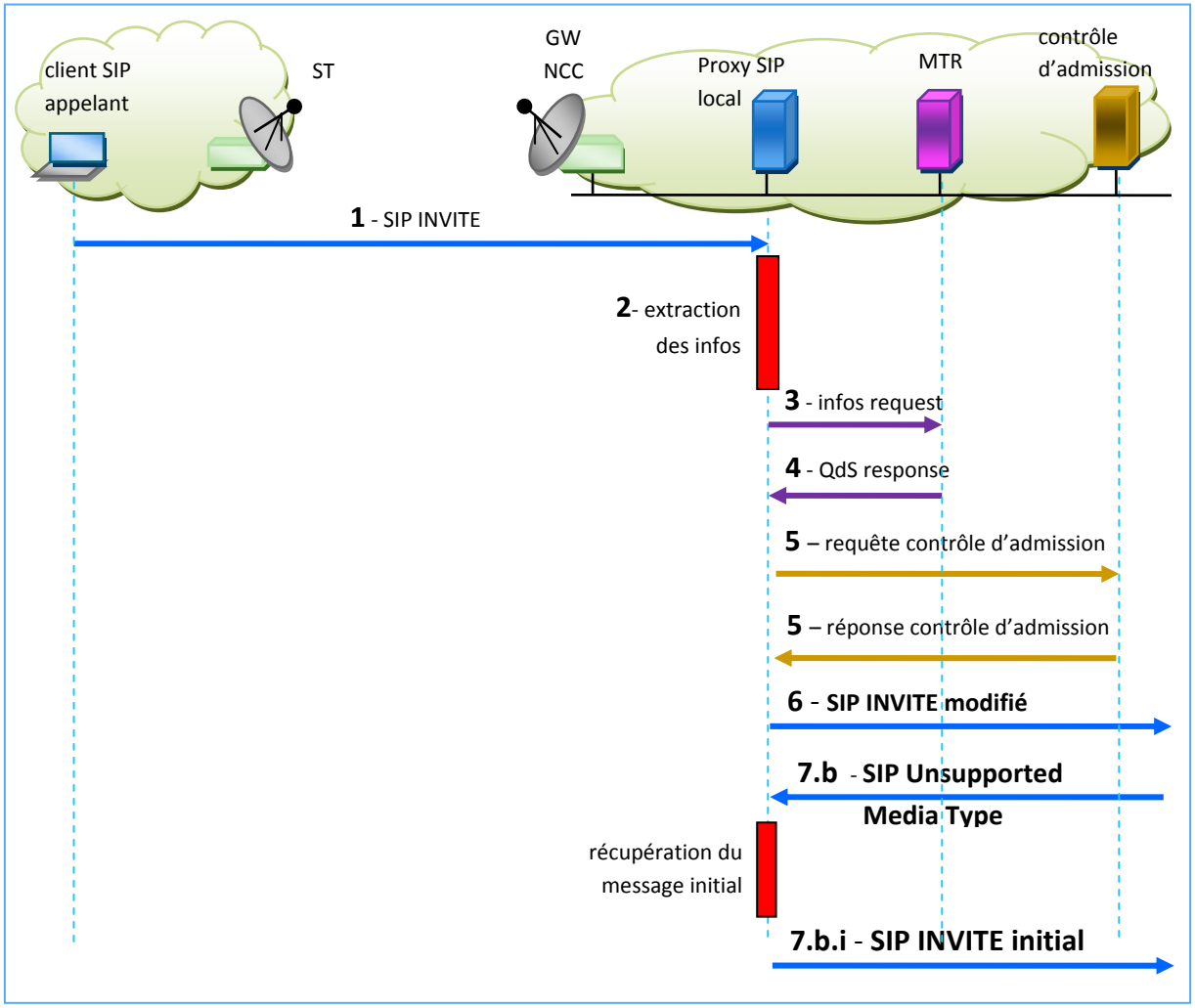


Figure 44 : session SIP refusée avec contrôle d'admission amélioré

Pour pouvoir changer le codec des flux actifs présents dans la classe de service de manière transparente pour les clients SIP, il est nécessaire d'avoir connaissance de la liste des codecs capables d'être supportés par chaque flux multimédia en cours. Cette liste est préalablement négociée durant de la phase d'établissement de session. On propose alors d'enregistrer cette liste de codecs et de l'utiliser afin de pouvoir changer de codecs durant la communication multimédia.

Deux scénarios typiques permettent de présenter cette amélioration.

Réseau Satellite peu chargé

Dans le premier scénario, le réseau satellite est peu chargé, la plupart des ressources réseau sont disponibles, par conséquent les demandes d'établissement de session multimédia avec support QdS sont acceptées. Cependant nous proposons tel que décrit en Figure 45, **d'enregistrer dans une base de données, toutes les informations des sessions qui s'établissent : les adresses IP et numéros de port des clients SIP, la liste de codecs finale, et le codec choisi pour la communication, l'identifiant de la communication SIP.**

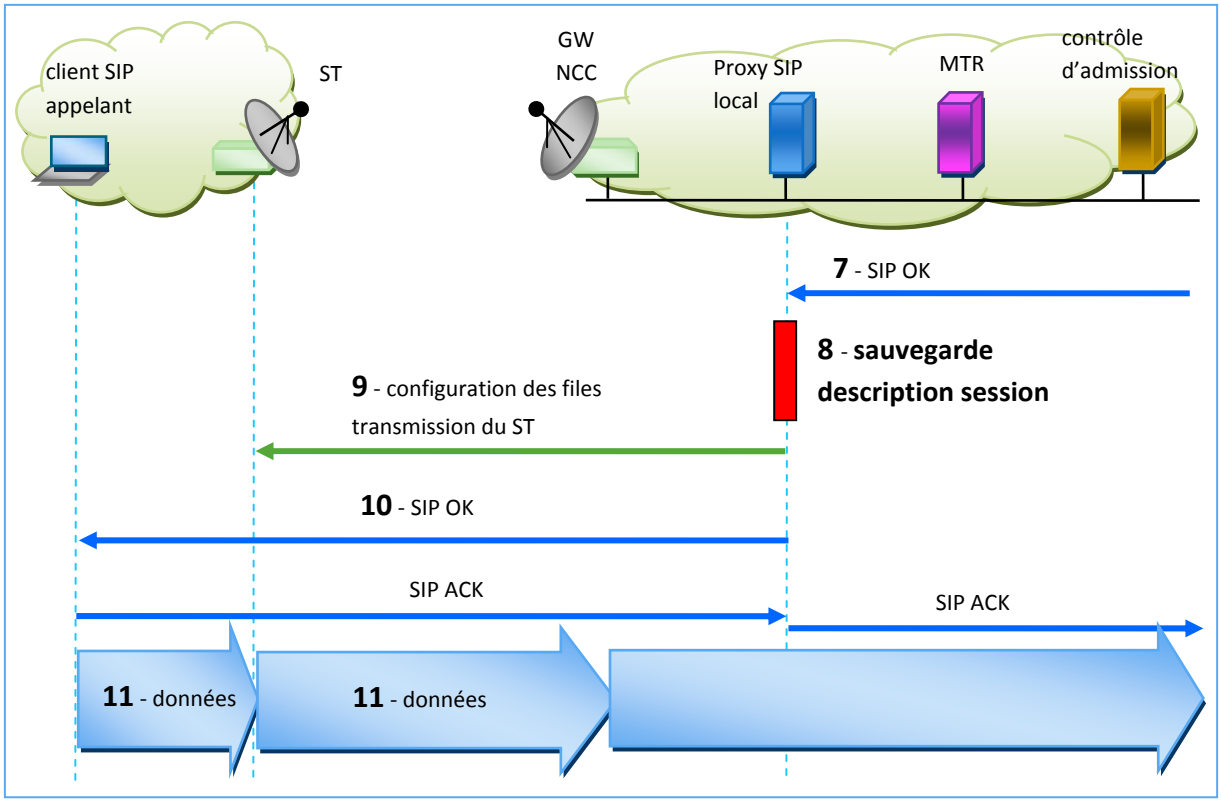


Figure 45 : session SIP acceptée avec sauvegarde de la description de session

Ces informations sont utilisées lorsque la voie retour du système satellite se retrouve en régime saturé.

Réseau Satellite saturé

Dans ce deuxième scénario, le réseau satellite est complètement saturé. L'ensemble des flux multimédia en cours dans la classe de service avec support QdS occupe la totalité des ressources allouées à cette classe. Dans ce scénario (c.f. Figure 47), durant l'établissement de session, lorsque le contrôle d'admission vérifie l'admissibilité des codecs pour la classe avec support de QdS, **aucun des codecs n'est admissible**.

Afin d'accepter le nouveau flux multimédia, soit on augmente la capacité maximale de la classe du ST, soit on diminue sa charge actuelle. On choisit de diminuer la charge de trafic présent de la classe de trafic. Pour cela, on propose de diminuer le débit des flux en cours. Sachant que les informations concernant les flux en cours ont été enregistrées dans une base de données, nous utilisons une méthode de changement de codec en cours de communication. Le codec en cours de chaque flux multimédia est remplacé par un codec à plus faible débit d'encodage, donc avec un plus faible débit de transmission. Cela permet alors de libérer des ressources réseau afin d'accepter le nouveau flux.

Afin de choisir le ou les codecs qui remplaceront le codec actuel de chaque flux en cours dans la classe (c.f. Figure 46), les données concernant les flux en cours dans la classe sont récupérées à partir de la base de données (liste de codecs supportables, codec en cours, débits associés, localisation des clients).

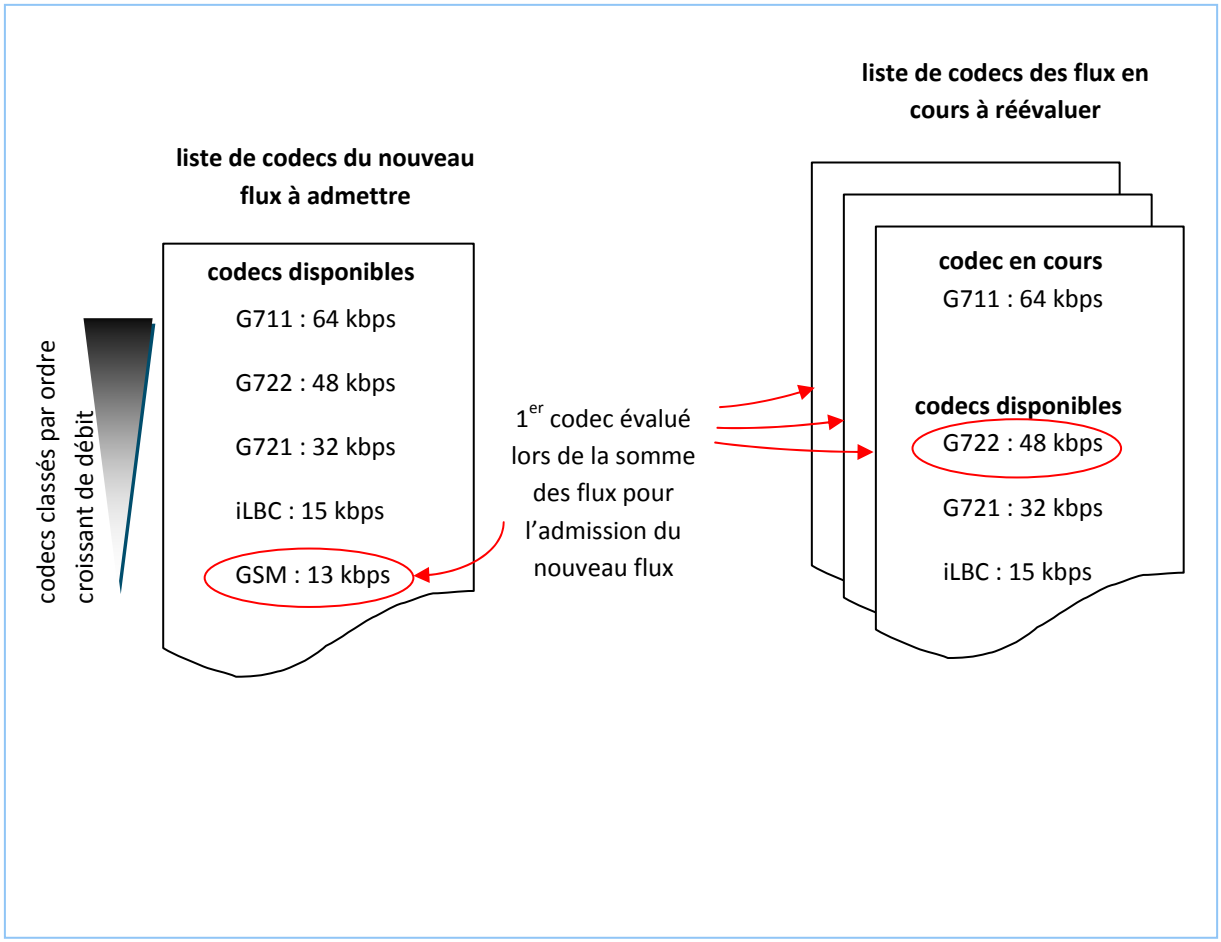


Figure 46 : codecs utilisés pour l'évaluation de l'admission du nouveau flux

Pour le nouveau flux à admettre, on commence par sélectionner ses codecs avec les plus faibles débits, puis remonter jusqu'à ceux ayant des débits proches des codecs utilisés par les flux en cours. En effet, l'objectif est d'homogénéiser la qualité du nouveau flux à admettre avec celle des flux en cours dans la classe.

Pour les flux en cours dans la classe, on cherche à diminuer leur somme de débit de flux, alors pour chaque flux, on sélectionne dans leur liste de codecs, ceux ayant un débit juste inférieur au codec actuellement utilisé.

À partir de cette proposition de sélection, on évalue une nouvelle somme de débit des flux en cours.

Si cette somme, ajoutée au débit du codec du nouveau flux à admettre est supérieure à la capacité totale de la classe, alors on recommence ce calcul en sélectionnant, pour chaque flux en cours, le codec suivant à débit inférieur.

Si cette somme, ajoutée au débit du codec du nouveau flux à admettre est inférieure à la capacité totale de la classe, alors la sélection de codecs utilisés pour effectuer cette somme est enregistrée pour le changement de codecs des flux en cours respectifs. On recommence l'évaluation de la somme, avec le codec suivant à débit juste supérieur jusqu'à homogénéiser la qualité du nouveau flux à admettre avec celle des flux en cours dans la classe.

Au final, on obtient :

- Pour le nouveau flux à admettre, une liste de codecs capables d'être supportés par la classe de service. Cette liste contient des codecs avec une qualité allant jusqu'à celle des codecs des flux multimédia en cours. L'objectif étant de respecter une certaine équité dans la qualité.
- Pour les flux en cours, une proposition de liste de codecs pour remplacer leur codec actuel.

Sachant que la nouvelle connexion multimédia est toujours en attente (le proxy SIP retient le message SIP INVITE), le proxy SIP intègre la nouvelle liste de codec dans le message SIP INVITE en remplacement de celle initialement fournie par le client SIP appelant. Puis il le transmet au domaine distant (proxy SIP distant puis client SIP appelé).

Le proxy SIP reçoit le message SIP OK, et à ce stade, deux opérations s'effectuent en parallèle : la fin de l'établissement de la nouvelle session et la mise à jour des sessions en cours nécessitant un changement de codec.

Fin de l'établissement de la nouvelle session SIP

La fin de l'établissement de la nouvelle session se poursuit en parallèle. Le proxy SIP (du réseau satellite) enregistre les informations concernant la nouvelle session dans une base de données : les adresses IP et numéros de port des clients SIP, la liste de codecs finale, et le codec choisi pour la communication, l'identifiant de la communication SIP. Le flux est alors accepté dans la classe de service et les mécanismes de réservation de ressources sont mis en place (dans le ST et le NCC) afin d'assurer un support de QdS au flux multimédia. Lorsque le flux de données multimédia arrive dans le ST pour être transmis sur le réseau satellite, il est dirigé vers la classe de service avec support de QdS.

Méthode de changement de codec pour les sessions en cours

Maintenant que nous avons connaissance des codecs qui peuvent être utilisés pour remplacer les codecs actuels des flux en cours, on propose d'utiliser la méthode SIP Re-INVITE. Elle permet de mettre à jour les paramètres (entre autre le codec utilisé) d'une session SIP en cours.

SIP Re-INVITE

Lorsqu'un client SIP souhaite modifier des paramètres de session, il envoie le message SIP Re-INVITE à son destinataire. Ce dernier en extrait les paramètres et, en cas d'approbation, modifie la session. Puis il répond avec le message SIP OK. Cette signalisation peut être effectuée sans interrompre le flux de données multimédia.

Pour les flux multimédia déjà présents dans la classe qui nécessitent de changer de codecs, cette opération doit s'effectuer de manière transparente pour les clients SIP. En effet, les clients SIP ne sont pas conscients de la QdS dans le réseau. Par conséquent, ils ne savent pas à quel moment déclencher le changement de codec. Nous proposons une méthode permettant à une tierce entité dans le réseau d'initier le changement de codec à la place des clients SIP. Pour cela, nous utilisons une méthode connue du monde de la sécurité réseau appelée « spoofing ».

Dans notre scénario, afin d'initier le changement de codec pour une communication SIP en cours, c'est notre proxy SIP qui joue le rôle d'attaquant. Pour chaque communication nécessitant un changement de codec, on propose que le proxy SIP crée et émette le message SIP Re-INVITE, avec les paramètres corrects des sessions en cours, à chacun des deux participants de la communication.

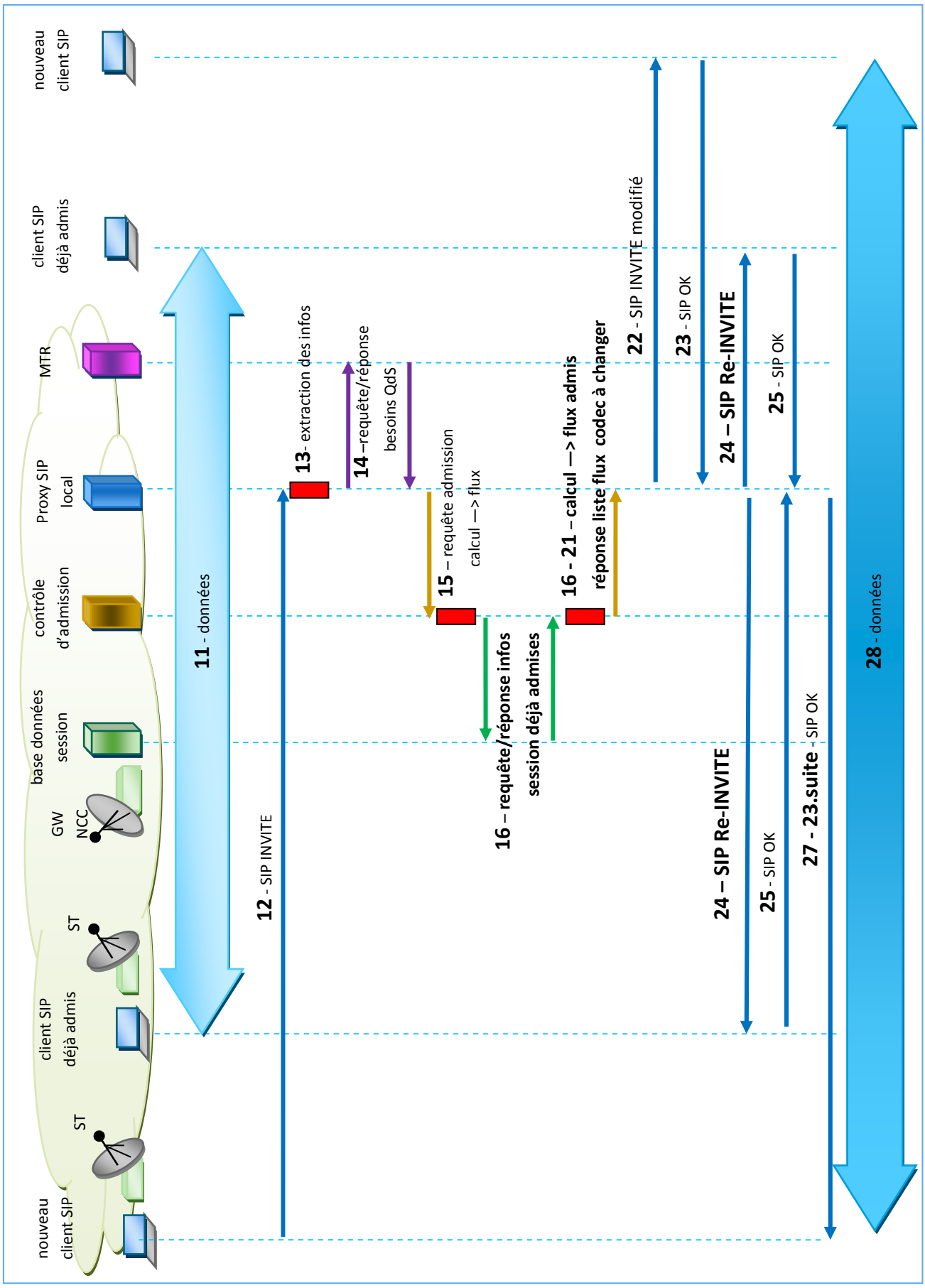


Figure 47 : scénario changement de codec des flux en cours

Pour notre contribution, la méthode de spoofing s'effectue au niveau session. En effet, il n'est pas nécessaire d'usurper l'identité au niveau réseau (adresse IP+numéro de port) d'un client SIP pour communiquer avec l'autre client SIP. Car nous utilisons l'option SIP « loose route » : tous les messages de signalisation de session, y compris les messages Re-INVITE doivent passer par les proxys SIP. Ainsi, on propose que notre proxy SIP construise le message SIP Re-INVITE adéquat pour chaque communication multimédia en cours, nécessitant un changement de codec.

À partir des informations de session (adresses IP, numéros de port, identifiant session SIP) recueillies dans notre base de données de session, on propose de construire l'en-tête du message Re-INVITE afin de donner l'impression qu'il a été créé par le client SIP, puis envoyé à notre proxy SIP (celui qui a créé le message) , qui lui se contente de le transmettre au client SIP distant. Le nouveau codec à utiliser pour la communication multimédia est introduit dans le corps du message SIP Re-INVITE (c.f. Figure 48).

Le proxy SIP envoie le message SIP Re-INVITE aux deux clients SIP de chaque communication multimédia nécessitant un changement de codec. Ces clients SIP reçoivent le message Re-INVITE et acceptent le changement de codec pour la session. Ils renvoient chacun leur message d'accord SIP OK au proxy SIP. À ce stade, le proxy SIP ne doit pas transmettre les messages SIP OK aux clients SIP. En effet, du fait de la méthode de spoofing utilisée pour le changement de codec, aucun des clients SIP n'a émis de message SIP Re-INVITE à l'autre, donc ils ne sont pas censés recevoir le message SIP OK. Une fois que cette méthode est appliquée pour tous les flux multimédia concernés par le changement de codecs, des ressources réseau sont libérées (par les clients SIP). En parallèle le nouveau flux multimédia est dirigé vers la classe et reçoit le support QdS nécessaire pour transmettre sur le réseau satellite.

Terminaison d'un flux multimédia dans la classe

Notre méthode de changement de codec s'effectue aussi lorsque l'un des flux multimédia dans la classe avec support QdS se termine. Ses ressources libérées sont réallouées aux flux toujours en cours. Les flux en cours dont le codec a été diminué, ont la priorité, afin de retrouver une meilleure qualité de média.

Un exemple de scénario est représenté en Figure 48, où deux communications multimédia sont en cours dans le système satellite : entre les clients SIP 1 et 4 et entre les clients SIP 2 et 3. La communication entre les clients SIP 1 et 4 se terminent.

Ainsi le proxy SIP reçoit le message SIP BYE, en extrait les identifiants de la session (adresses IP et numéros de port). Le contrôle d'admission utilise ces identifiants pour rechercher dans la base de données le codec en cours et le débit associé à ce flux qui se termine. Il récupère également dans la base, la liste de codecs et le codec en cours des flux actifs, dont le codec en cours a été diminué.

À partir de ces informations recueillies, il évalue quel codec à débit supérieur peut remplacer le codec actuel sans dépasser les ressources libérées par le flux se terminant. Ainsi, on obtient une proposition de codec pour remplacer leur codec actuel. Le contrôle d'admission fournit au proxy SIP la liste des flux avec les codecs de remplacement. La méthode précédente de changement de codec est appliquée. En même temps le proxy SIP continue la terminaison de la communication multimédia en demandant au réseau sous-jacent la libération des ressources pour le flux concerné.

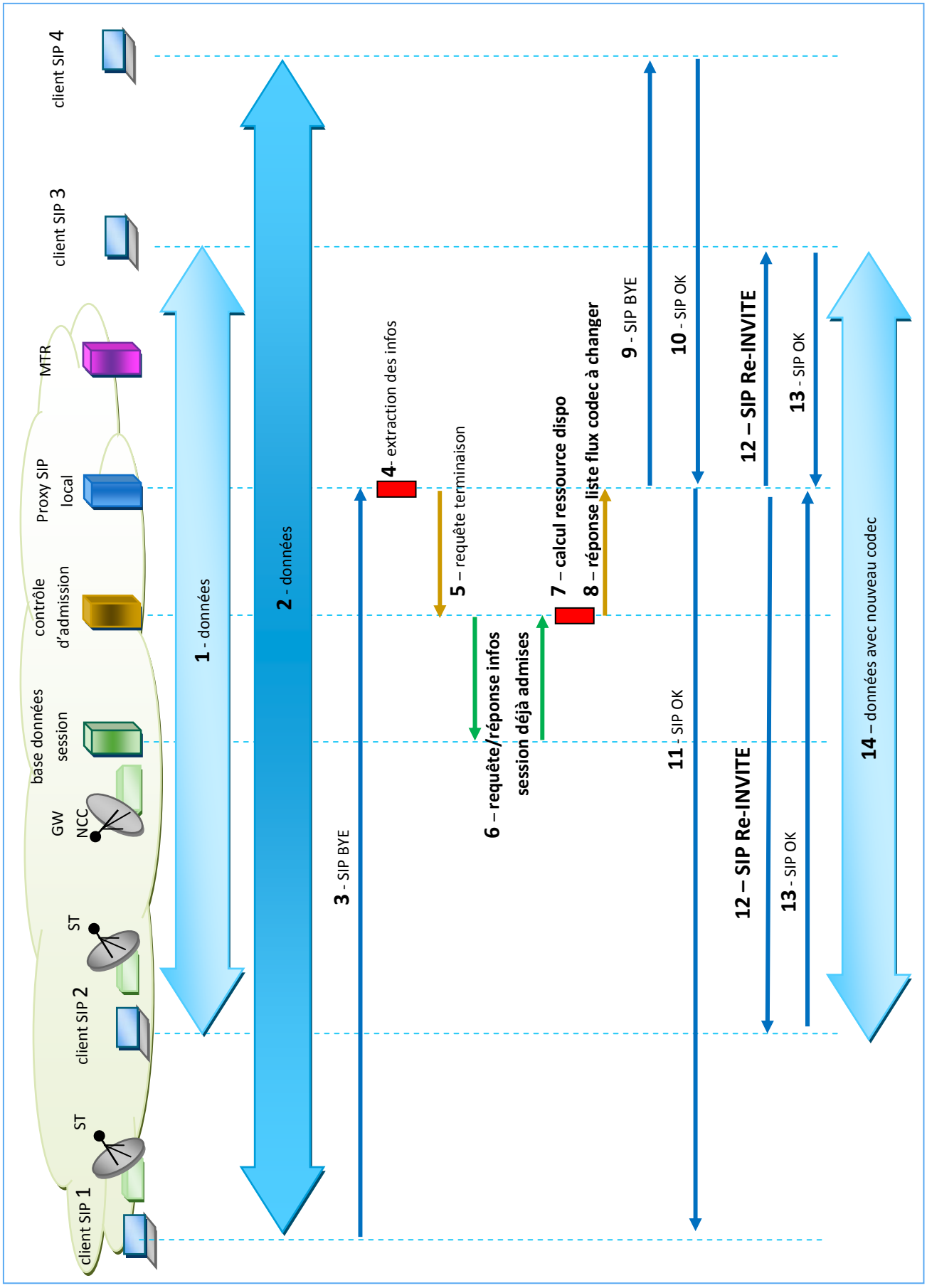


Figure 48 : changement de flux à la terminaison de session

3.5.5. Contrôle de débit des applications multimédia pour la classe de service optimisée (Best Effort) : Interaction Cross-Layer (liaison – session)

Jusque là, les propositions précédentes s'appliquent aux flux multimédia signalés présents dans une classe de trafic avec QdS garantie entièrement dédiée à ces flux. La signalisation de session SIP ainsi que la classe de service dédiée offrent ainsi un environnement contrôlé : la quantité minimale et/ou maximale de ressources attribuée à cette classe est connue. La quantité de ressources utilisées pour chaque flux présent dans cette classe est également connue grâce à la signalisation de session SIP (établissement, changement de codec, terminaison).

Maintenant, dans un cadre de type service optimisée (Best-Effort), l'environnement est bien différent :

- La capacité réseau de la classe best effort est dépendante des ressources utilisées par les classes à priorité supérieure. En effet, la classe best effort se voit allouer les ressources non utilisées par ces classes à priorité supérieure.
- La majeure partie de la charge de la classe est constituée de flux non signalés (ex. transfert de fichiers, navigation web, etc). Il y a donc peu de moyen d'appliquer un contrôle d'admission et un contrôle de flux précis sur la classe.

Comment peut-on utiliser les mécanismes précédemment présentés pour les applications multimédia dans un environnement réseau aussi variable et imprédictible ?

Face à ce constat, on propose d'étendre la proposition du contrôle d'admission et contrôle de débit pour les applications multimédia dans la classe de service de type Best Effort.

Contrôle d'admission basé sur SIP pour la classe de service Best Effort

Afin d'appliquer le contrôle d'admission précédemment proposé, sur la classe Best-Effort, on ne considère que les flux multimédia à l'intérieur de cette classe, et sont prioritaires par rapport aux autres flux présents dans la classe (ex. web, ftp, mail, etc.). En faisant cette abstraction, la méthode de changement de codec est alors applicable lors de l'admission ou la terminaison d'un flux multimédia, si le besoin est.

Il reste cependant un point à traiter : il s'agit de la quantité de ressources satellite attribuées à la classe qui est très variable. Si pour une supertrame donnée, la classe de service Best Effort reçoit moins de ressources nécessaires pour la transmission de l'ensemble des flux multimédia dans les délais impartis, des paquets seront retardés dans la file de transmission Best Effort et ils seront obligés d'attendre le prochain cycle requête/allocation pour espérer être transmis.

Une solution est d'appliquer le changement de codec auprès des flux multimédia afin de baisser le débit de transmission des flux lorsque les ressources nécessaires sont insuffisantes.

Interaction Cross-Layer Liaison vers Plan de contrôle

La quantité de ressources attribuées, à savoir le plan d'allocation de ressources, est reçu par les STs du réseau satellite. À partir du plan d'allocation, nous pouvons déduire la quantité ressources

attribuées à chaque classe de service implémentée dans le ST. Nous créons alors une interaction Cross-Layer entre la couche liaison du ST et le plan de contrôle.

La couche liaison du ST, plus précisément le client DAMA est chargé de calculer les requêtes de ressources pour les files de transmission du ST. Lorsqu'il reçoit le plan d'allocation de ressources, il est en mesure de le fournir aux couches supérieures à travers l'interface cross-layer.

Le contrôle d'admission installé du côté ST, reçoit le plan d'allocation de ressources à travers l'interface cross-layer, et vérifie la disponibilité des ressources pour la transmission des flux multimédia de la classe de service BE. Si la quantité de ressources attribuées à la classe BE est inférieure à la quantité de ressources nécessaires pour la transmission des flux multimédia de la classe BE, le contrôle d'admission interroge la base MTR pour récupérer les informations de sessions multimédia présentes dans la classe BE. Comme pour la contribution précédente, il réévalue la charge de la classe avec les différents codecs disponibles des flux. Si une proposition de changement de codec permet la diminution de la charge de la classe à hauteur des ressources disponibles, il la transmet au proxy SIP qui lui va déclencher le changement de codec auprès des clients SIP désignés par le contrôle d'admission.

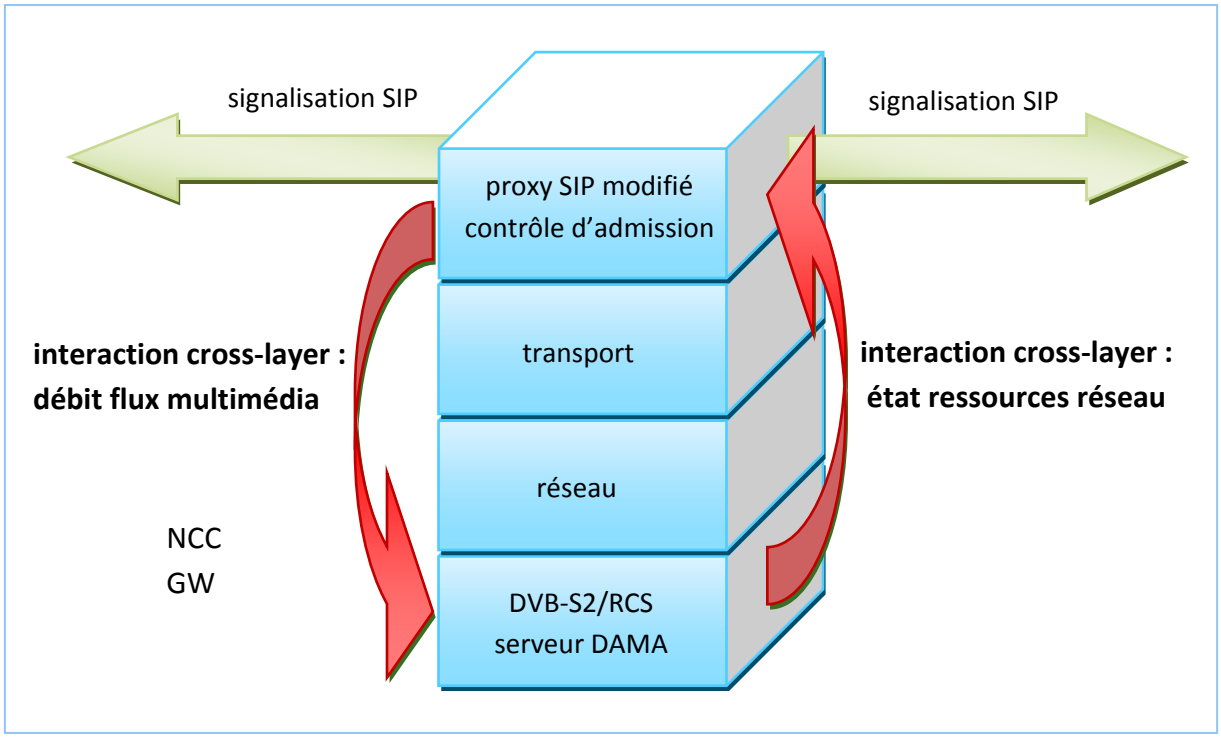


Figure 49 : interactions cross-layer double SIP - MAC dans le NCC

Le contrôle d'admission dispose du plan d'allocation de ressources selon sa période d'émission par le NCC. Selon les systèmes satellite, cette période varie entre 50ms et 500ms. Afin d'éviter une émission trop fréquente du plan d'allocation à travers l'interaction cross-layer, nous proposons que la couche session spécifie ses besoins en ressources à la couche liaison du ST.

Interaction Cross-Layer plan de contrôle vers la couche liaison

Ainsi la couche liaison émettra l'information de manque de ressources uniquement lorsque cela sera nécessaire. Pour que le plan de contrôle, plus précisément le mécanisme de contrôle d'admission, fournisse le débit de chaque session multimédia admise au client DAMA du ST, nous créons une interaction cross-layer dans le sens contrôle d'admission vers le client DAMA. Ce dernier doit alors, à chaque réception du plan d'allocation, vérifier la disponibilité des ressources pour les flux multimédia. Si la quantité de ressources reçues est inférieure à celle nécessaire pour les flux multimédia, le client DAMA envoie au contrôle d'admission, à travers l'interaction cross-layer, cette différence de quantité de ressources. Le contrôle alors recherche les flux dont le codec actuellement utilisé peut être changé pour un, moins gourmand en débit.

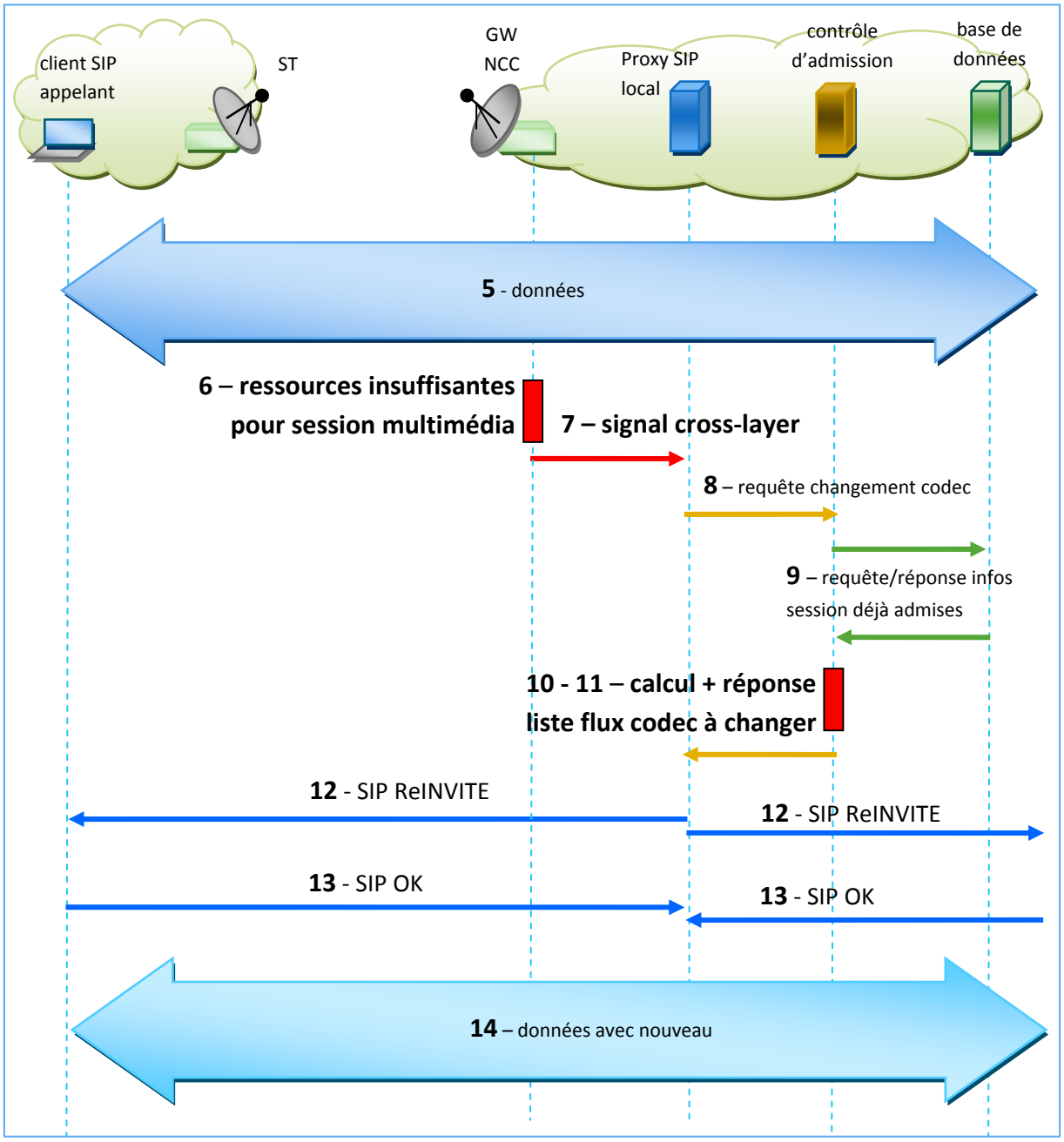


Figure 50 : scénario de changement de codec dans un environnment Best Effort

En cas de réussite le contrôle d'admission demande au proxy SIP de déclencher le changement de codec pour les sessions multimédia concernées.

Pour que la couche MAC du NCC soit en mesure de signaler à la couche session SIP les moments de pénurie de ressources réseau pour le ST concerné par le ou les flux multimédia, il est nécessaire qu'il puisse identifier ces STs et la quantité de ressources seuil. La Figure 50 présente la communication cross-layer. On propose, à la fin de l'établissement de chaque nouvelle session SIP dans la classe BE, que le proxy SIP communique au NCC le débit du codec utilisé, ainsi que l'adresse IP du client SIP présent dans le réseau. Une interaction Cross-Layer est établie entre la couche MAC du NCC et la couche session SIP. Le sens de circulation de ces informations véhiculées par l'interaction se fait de la couche SIP vers la couche MAC.

3.5.6. Conclusion

Cette contribution propose un contrôle d'admission amélioré avec un contrôle de débit pour les applications multimédia dans un environnement réseau satellite d'accès DVB-S2/RCS. Ces améliorations permettent d'optimiser l'utilisation des ressources réseau afin d'accepter un maximum d'utilisateurs d'applications multimédia dans le réseau satellite. Pour cela, un mécanisme de changement de codecs pour les flux multimédia est établi pour diminuer la charge du réseau lors de l'admission d'un nouveau flux, et augmenter la charge réseau (augmenter la qualité des flux) lorsqu'un flux se termine. Ainsi, les applications multimédia gardent un certain niveau de qualité de service et l'utilisation des ressources réseau est optimisée.

Cependant, il est important de noter que le contrôle d'admission se fait au niveau d'un utilisateur final (fournie par la signalisation SIP), et non pas du client du réseau satellite (qui paie pour les ressources satellite et éventuellement pour la mise à disposition du ST). En général on ne connaît donc pas ces utilisateurs finaux, il peut donc être difficile de faire un contrôle d'admission très précis. Nous n'allons pas résoudre ce problème ici, mais il est nécessaire de signaler son existence.

3.6. Conclusion des solutions basées Cross-Layer

Dans ce chapitre, nous avons proposé plusieurs solutions basées sur l'approche cross-layer dans un environnement réseau satellite DVB-S2/RCS. Elles permettent le déploiement des applications multimédia interactives en utilisant l'allocation dynamique de ressources du réseau satellite. La première contribution réduit le délai d'attente des premières données arrivant dans les tampons d'émission des ST au démarrage flux multimédia. Cette réduction du délai est obtenue par une anticipation des besoins en bande-passante de l'application. Une communication cross-layer permet de synchroniser l'obtention effective des ressources réseau par le ST et l'arrivée des données multimédia dans le ST pour leur transmission.

Puis la suite de la transmission des données multimédia est prise en charge par un mécanisme d'anticipation de ressources. Ce dernier se base à la fois sur l'état passé (d'ordre 2) du trafic multimédia, et également sur un facteur d'anticipation. Ce facteur d'anticipation, par le biais d'une interface cross-layer SIP-MAC, utilise les caractéristiques annoncées (ex. débit) par les flux multimédia signalés et l'état actuel de ces flux multimédia afin d'anticiper les futurs besoins en ressources durant la transmission.

Enfin pour optimiser les ressources réseau satellite, un contrôle d'admission et un contrôle de débit permettent une adaptation du trafic multimédia en fonction de l'état du réseau qui est fourni par la couche MAC du NCC à travers une interaction cross-layer.

Ce chapitre a permis de montrer l'utilité et l'efficacité des interactions cross-layer. Dans notre cadre de travail, elles offrent la possibilité de déployer des services multimédia interactifs dans un environnement sans fil, tel que le satellite. Nous proposons dans le chapitre suivant d'uniformiser les différentes propositions d'interactions cross-layer dans la conception d'une architecture dédiée.

4. Architecture à Communication Cross-Layer

4.1. Motivations

À travers les trois solutions précédentes (c.f. sections 3.3, 3.4 et 3.5), nous avons montré clairement, dans un environnement réseau sans fil (ex. réseau satellite), le besoin de communication cross-layer. Cependant, la création de diverses interactions cross-layer directement intégrées dans la pile de communication TCP/IP, selon les besoins de chacun, risque à long terme, de ne plus être exploitable du point de vue des concepteurs. Nous souhaitons, à travers une architecture cross-layer dédiée, améliorer les faiblesses (en termes de performance principalement) de la pile protocolaire Internet tout en limitant ses modifications pour conserver une compatibilité maximale avec les services existants. Pour cela nous proposons une architecture à communication cross-layer via une entité intermédiaire. Cette dernière permet de garder une compatibilité avec l'architecture classique en couche. Cela nous permet donc de maintenir tous les avantages inhérents à une architecture modulaire en couches isolées, tels que la robustesse ou la facilité d'évolutivité. Elle permet également une évolution continue de l'entité cross layer, par ajout ou suppression de protocoles. De plus, en centralisant les informations cross-layer à partager, les différentes couches sont bien moins sollicitées. Après analyse des deux technologies réseau sans fil Wifi et Satellite décrites respectivement en section 1.2.1 et 1.2.3, nous avons dégagé un certain nombre de paramètres pertinents susceptibles d'être exploitables à différents niveaux du système de communication afin d'en tirer profit pour l'optimisation des communications (c.f. Table 3).

Table 3 : paramètres exploitables par les communications cross-layer

		Technologies réseau	
		Wifi	**Satellite**
Couches	**Physique**	-Fréquence canal -Technique de modulation, type de codage -La bande passante -ACM (adaptative codage modulation) -Le débit -Compression (s'il y a ou pas) -Information sur la carte réseau -Information sur les cartes drivers -La puissance d'émission PE -La consommation d'énergie -Les statistiques d'utilisation du matériel -Contrôle de délai -Le rapport signal sur bruit Eb/N0 -BER (bit error rate)	-Bande passante -Efficacité de la bande passante (en bps/hz) -Eb/N0 -Technique de lutte contre les affaiblissements - Schéma de modulation/codage utilisable par chaque RCST actif -le nombre de timeslots demandés par le RCST et sujets au « dama controller algorithm »
	Liaison	-Paramètres de la QoS courante - Paramètres de la QoS désirés -Back-off actif (rapport avec CSMA/CA) -Timeout delay -Longueur et état des buffers MAC - notification de handoff (transport, liaison)	- tailles des files d'émission MAC - taux de disponibilité instantané MAC - latence du DAMA - capacité du lien - Période de la supertrame - types requête - période requête – historique requêtes - fréquence et time slot - allocations reçues par requête et par classes - Maximum transmission unit (MTU) -Delais et gigue de tcp

Cette liste de paramètres est loin d'être exhaustive, mais permet d'avoir une première idée des données pouvant être mises à disposition et partagées afin d'optimiser les performances des systèmes de communication vis-à-vis des applications critiques.

4.2. Architecture Globale

L'architecture cross-layer proposée dans ces travaux de thèse traitent uniquement les interactions cross-layer locales, cependant une évolution en modèle globale est envisagée (c.a.d. une intégration horizontale). Cette architecture est composée de différents modules (c.f. Figure 51).

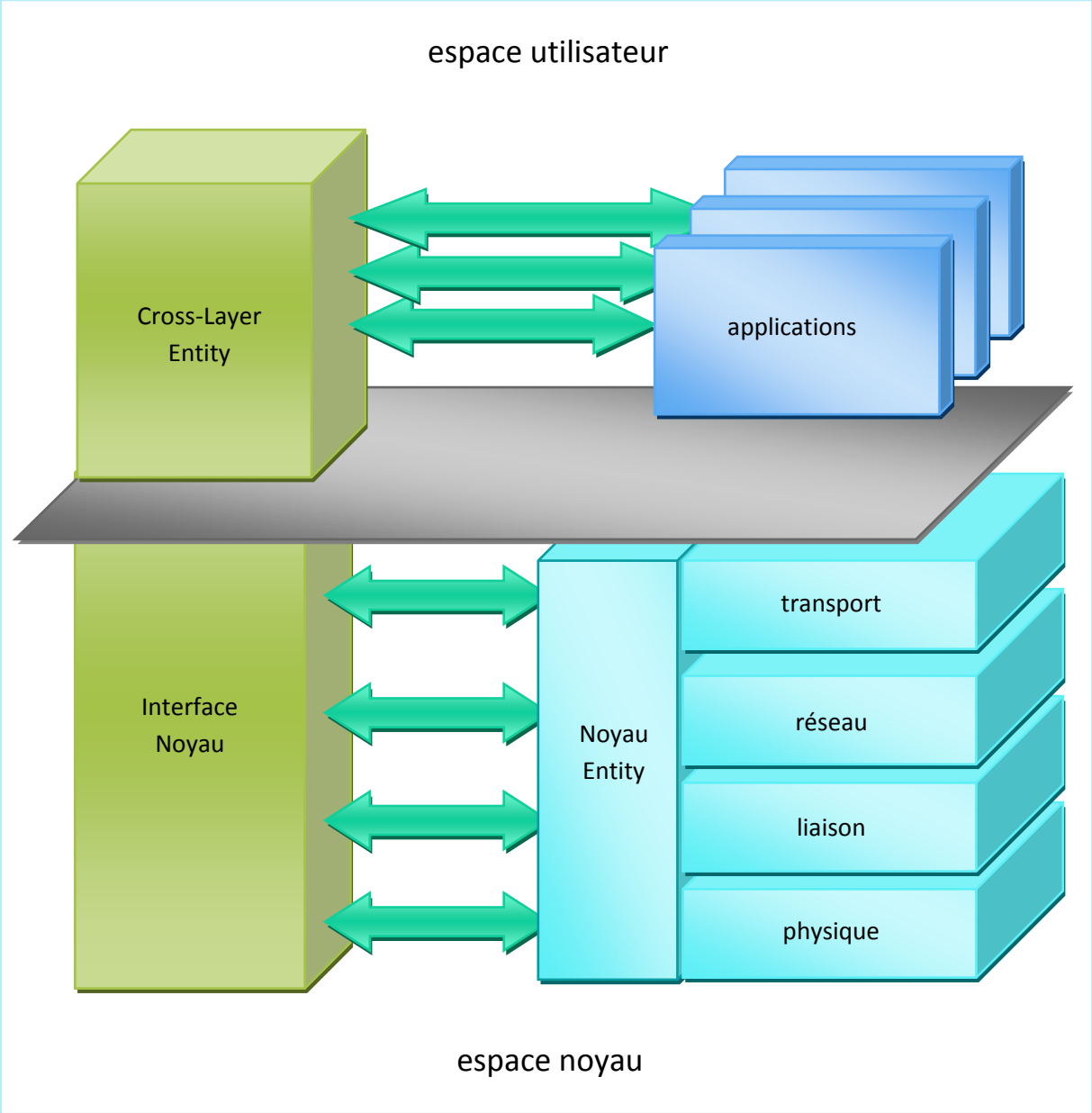


Figure 51 : architecture de communication cross-layer

- **Cross-Layer Entity (CL_Entity)** : dans l'espace utilisateur, ce module sert d'interface de dialogue entre les applications, et le noyau du système ; il permet également de stocker les données et leurs valeurs mises à jour. Rappelons que les systèmes actuels implémentent un noyau qui regroupe les protocoles de couches basses. Cela va de la couche physique, jusqu'à la couche transport.
- **Interface Noyau (I_Noyau)** : est l'interface entre l'espace noyau et le CL_entity. En effet, il est le module de l'espace utilisateur communiquant avec l'espace noyau d'un côté, et avec le CL_entity de l'autre. Il permet d'une part, selon les messages envoyés depuis l'espace noyau,

de construire les messages adéquats et de les envoyer au CL_entity, et d'autre part de récupérer les demandes du CL_entity et de les traiter en communiquant avec l'espace noyau.

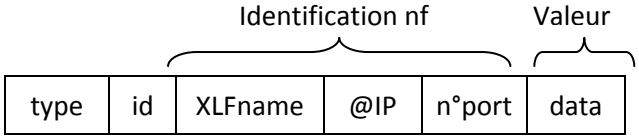
- **Noyau Entity** : Enfin, dans l'espace noyau, un module noyau permet de récupérer les données exportées des couches basses du noyau et de dialoguer avec l'espace utilisateur.

4.3. Protocole de communication

Afin de pouvoir échanger des informations, un protocole de communication est mis en place, il est représenté en Figure 52 et Figure 53. Les données échangées sont appelées les « Cross-Layer Feature »(XLF). Les méthodes suivantes sont proposées.

4.3.1. Enregistrement de XLF

Cette méthode permet au noyau, lors de l'initialisation des paramètres des couches basses, de signaler au CL_entity les données qui sont mises à disposition pour partage par les différents protocoles. Le format du message est le suivant :



Le champ *type* correspond au type du message envoyé. Ici *type*=3. Le *id* est un identifiant unique permettant d'identifier l'échange. Le champ *XLFname* est le nom de la donnée mise à disposition, le champ *@IP* est l'adresse IP du système. Chaque couche basse utilise le champ *n°port* comme numéro de port pour la communication. Le champ *data* contient la valeur initiale de la donnée.

Le CL_entity reçoit cette demande et enregistre la donnée et sa valeur initiale dans la base de données.

À partir de là, lorsque la donnée atteint un seuil prédéfini dans le noyau, ce dernier envoie la mis à jour de la donnée au CL_entity, avec le champ *type*=2.

Le CL_entity met à jour sa base de données avec la nouvelle valeur.

4.3.2. Demande de liste de XLF

À partir de cela, les applications sont en mesure de requérir la liste de données XLF enregistrées disponibles du système, avec *type*=5 :

Type	Id

Le CL_entity consulte sa base de données et répond avec la liste des XLF disponibles :

Type	Id	NbXLFregistered	Name 1	Name 2	

Le champ *NbXLFregistered* représente le nombre de XLF disponibles qui vont suivre dans le reste du message. Les champs *Name* fournissent le nom de chaque donnée disponible.

4.3.3. Demande de notification de XLF

Une application peut demander au CL_entity à être notifiée de la valeur d'une donnée disponible. Il peut demander :

- la mis à jour d'une donnée (en cas de changement de valeur), champ *type*=6 ;
- de manière ponctuelle la valeur de la donnée, champ *type*=8 ;

Type	Id	XLFname	@IP	n°port

Dans le cas d'une mis à jour, le CL_entity consulte sa base de données et renvoie un message avec le champ *type*=2 et le champ *valeur* :

Type	Id	XLFname	Valeur

Dans le cas d'une demande ponctuelle, le CL_entity transmet cette demande au noyau.

Le noyau répond avec le message, *type*=0.

Le CL_entity peut également requérir auprès du noyau, un envoi périodique de donnée. Un champ *periode* permet de spécifier la période souhaitée, et *type*=0 :

Type	Id	XLFname	@IP	n°port	Periode

Le noyau répond avec le message, *type*=1.

4.3.4. Demande de modification de XLF

L'application a également la possibilité, lorsqu'elle l'estime, de modifier la valeur d'une donnée dans les couches basses du noyau :

Type	Id	XLFname	@IP	n°port	data

Le champ *data* contient la nouvelle valeur souhaitée, avec le champ *type*=7.

Le CL_entity transmet ce message au noyau, avec *type*=2.

Le noyau renvoie alors au CL_entity la valeur modifiée, qui correspond à la valeur exacte demandée ou la plus proche possible. Le champ *type*=4.

Le CL_entity renvoie la valeur à l'application avec *type*=1.

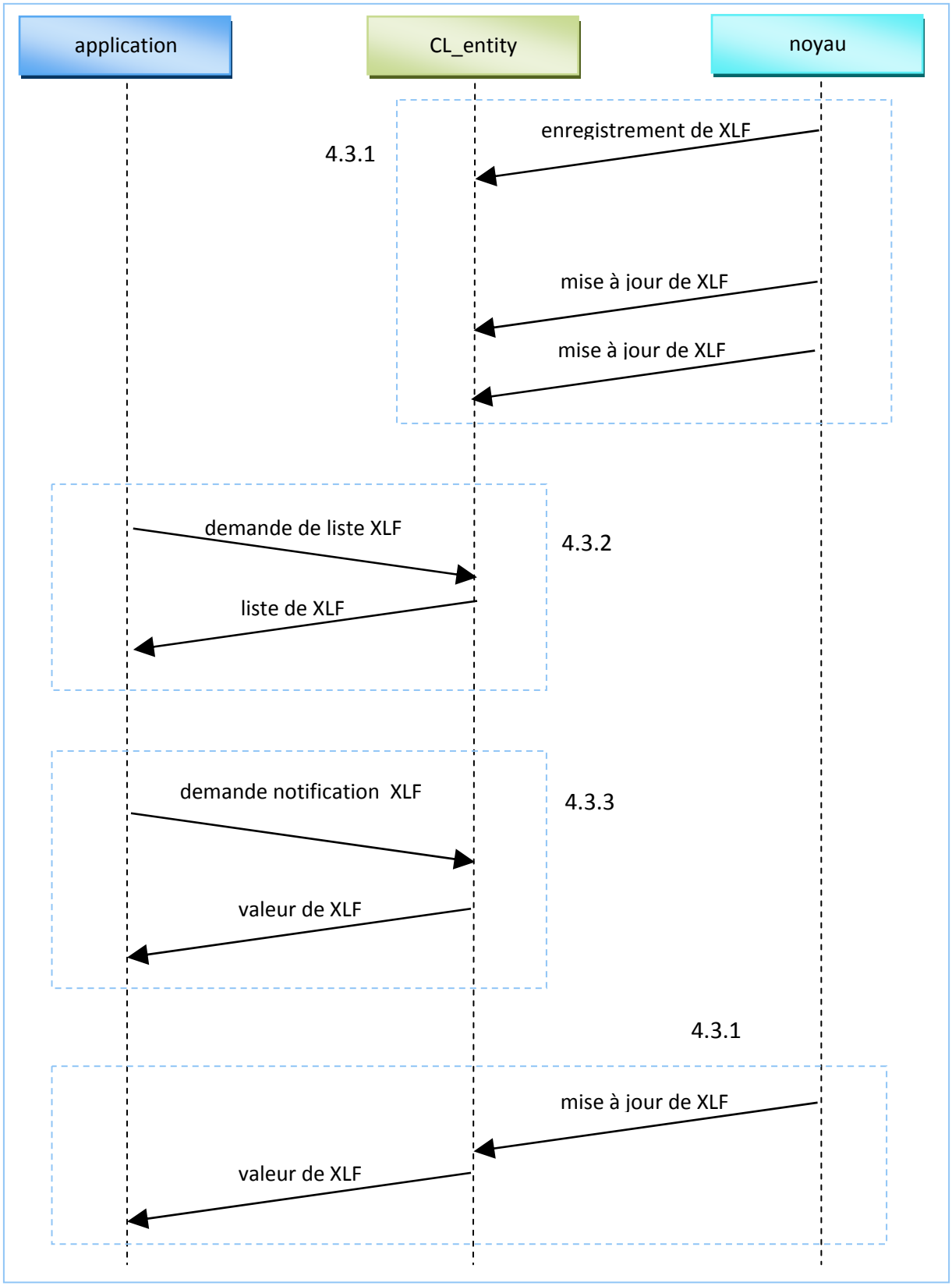


Figure 52 : protocole de communication de XLF

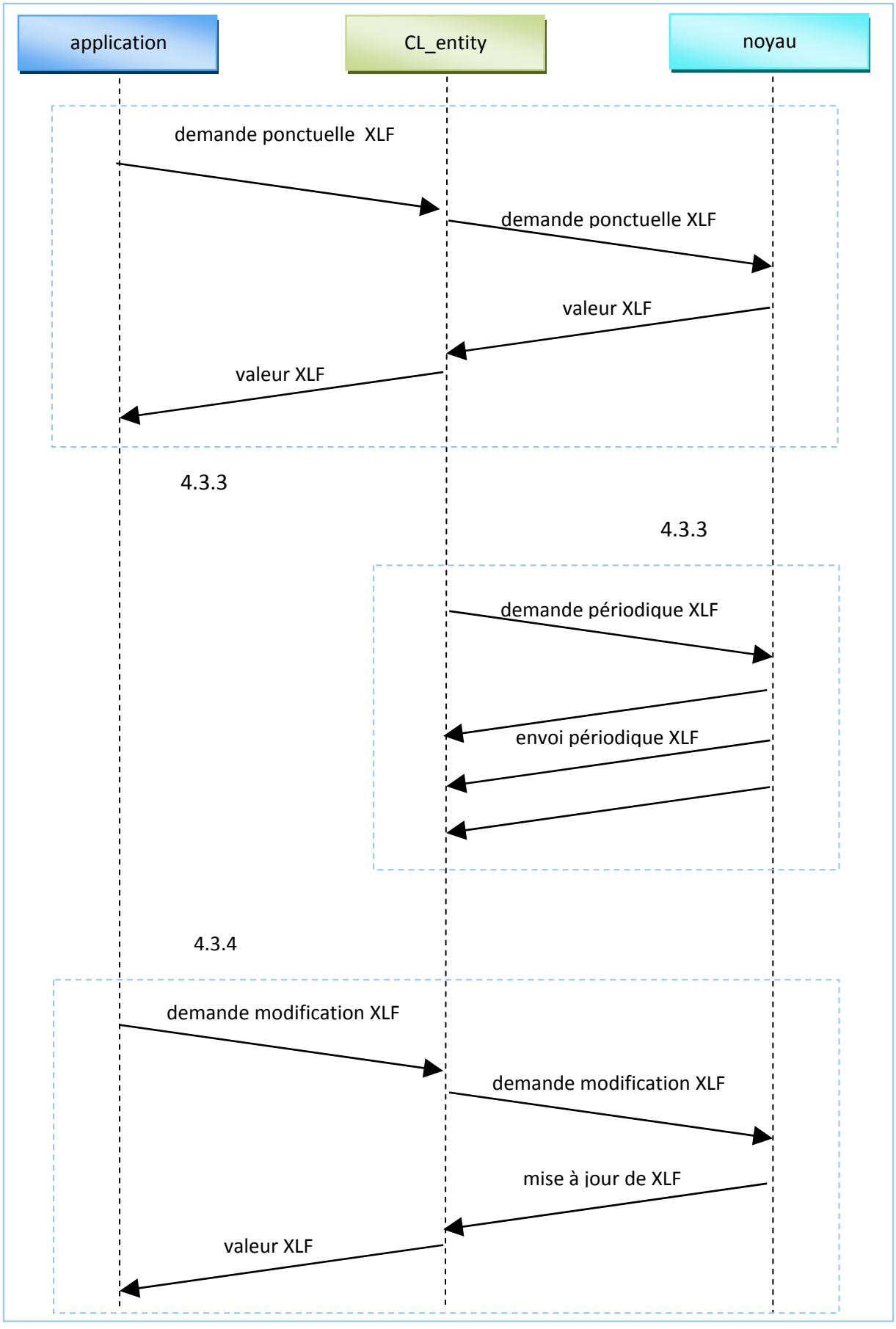


Figure 53 : protocole de communication de XLF (suite)

4.4. Implémentation, développement

Nous présentons brièvement le développement de notre architecture cross-layer.

Cette architecture est développée dans l'environnement Linux, en utilisant les langages de programmation C et C++. Pour les communications entre l'espace utilisateur et l'espace noyau, nous utilisons le mécanisme Netlink.

Une socket Netlink [78] est un processus interne de communication spécial pour transférer des informations. Netlink permet des communications d'informations entre les modules du noyau et les processus de l'espace utilisateur, dans les deux sens et sont compatibles avec les protocoles IPv4 et IPv6. Il consiste en une interface basée sur les sockets standards pour les processus utilisateur et d'une API interne pour les modules du noyau. On peut noter que c'est un service orienté datagramme. Il est possible de travailler en multicast avec netlink. En effet, chaque famille netlink à un ensemble de 32 groupes multicast. Cependant, Netlink n'est pas un protocole fiable. Des messages peuvent être perdus s'il n'a pas assez de mémoire ou si une erreur se produit. Pour assurer un transfert fiable, des mécanismes de retransmission doivent être mis en place.

Le Figure 54 présente le diagramme de classe et la Figure 55 définit les différentes interactions au sein de l'architecture cross-layer.

La classe Socket gère des fonctions relatives à la communication : création et terminaison de communication ; envoi et réception de messages.

La classe gestion_paquet gère l'encapsulation des messages.

La classe CL_entity gère les communications cross-layer. Il interagit avec l'interface noyau et les applications. La classe CL_entity hérite des classes socket et gestion_message, cela lui permet de communiquer (écoute, envoi) et d'appliquer le comportement adéquat (cf. scenarii) en fonction du type de message reçu.

La classe I_noyau est l'interface entre l'espace noyau et le CL_entity. En effet, il est le module de l'espace utilisateur communiquant avec l'espace noyau d'un côté, et avec le CL_entity de l'autre. Il permet d'une part, selon les messages envoyés depuis l'espace noyau, de construire les messages adéquats et de les envoyer au CL_entity, et d'autre part de récupérer les demandes du CL_entity et de les traiter en communiquant avec l'espace noyau.

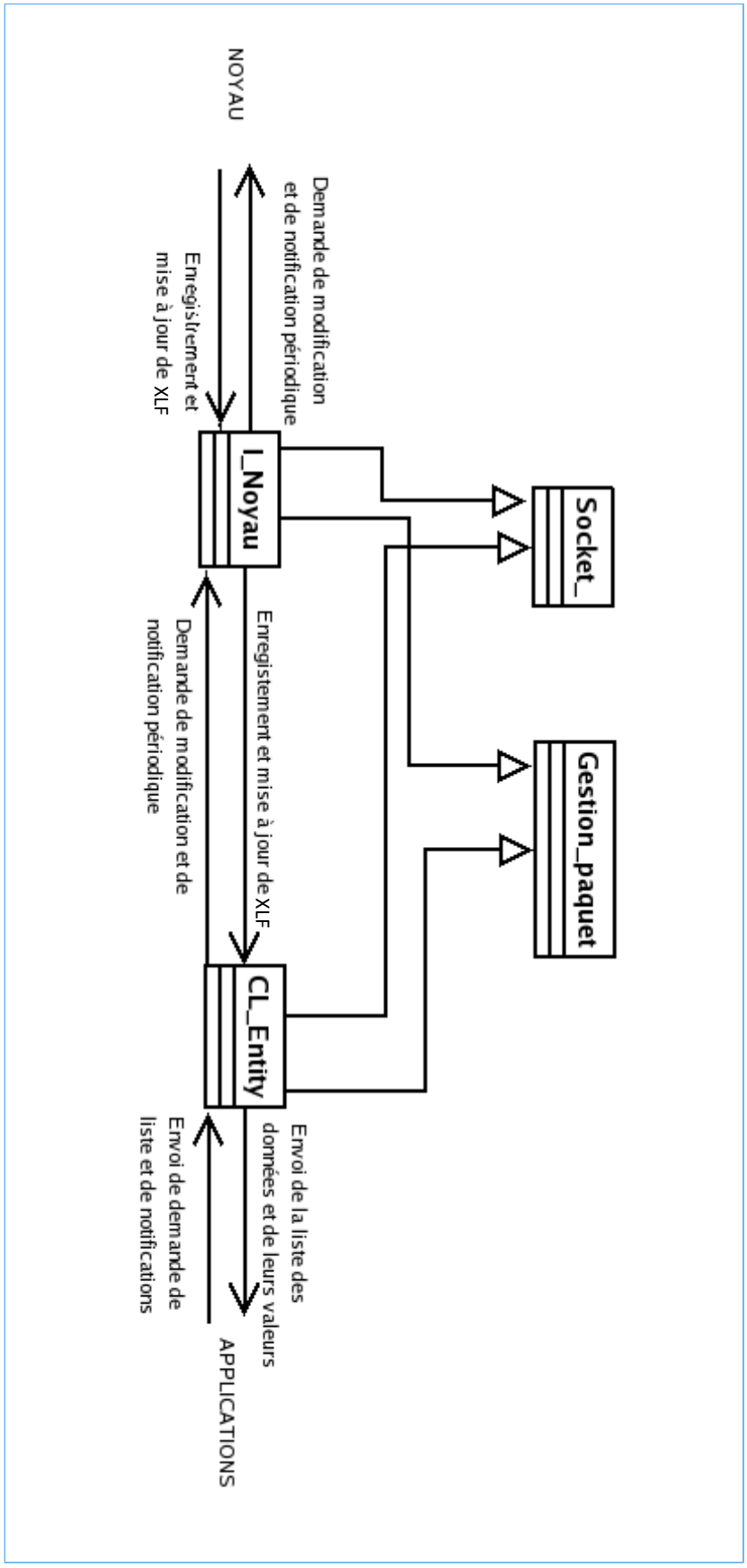


Figure 54 : diagramme de classes

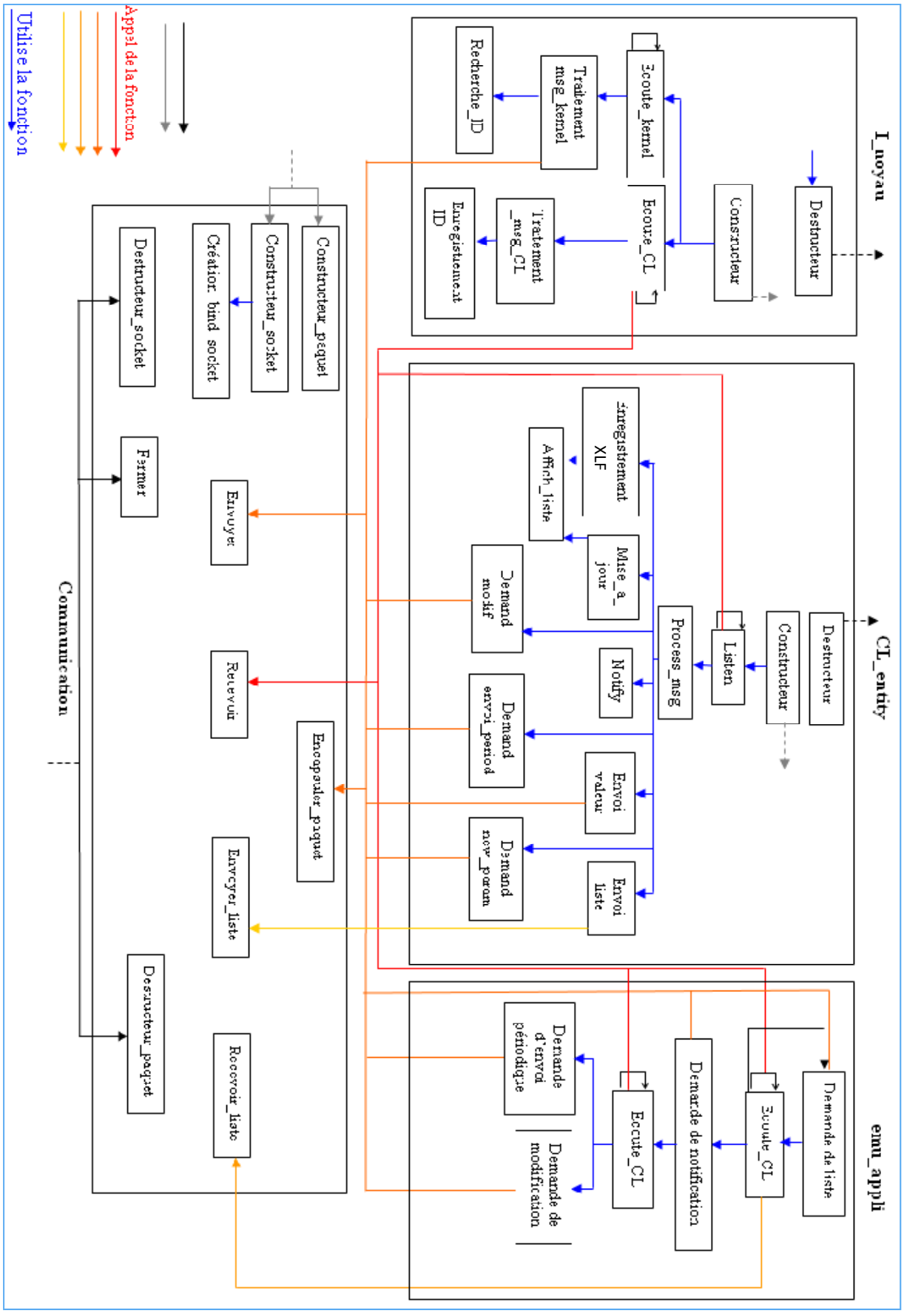


Figure 55 : schéma des interactions de l'architecture cross-layer

4.5. Conclusion et perspectives

Nous avons présenté dans ce chapitre la conception d'une architecture cross-layer pour les systèmes de communication actuel en couches. Elle utilise une entité intermédiaire CL_entity qui établit et gère des communications cross-layer de données locales entre les applications et les couches basses des systèmes. Ceci permet de conserver une compatibilité avec l'architecture classique en couche. Une base de données locale permet de stocker les données à partager, ainsi les différentes couches sont moins souvent sollicitées.

L'architecture et le déploiement proposés répondent non seulement aux problématiques étudiées, mais pourront aussi être réutilisés par la suite pour tout nouveau besoin d'interaction cross-layer. Nous projetons de faire évoluer cette architecture. L'objectif serait de pouvoir traiter les informations obtenues, c'est-à-dire les utiliser afin de créer de nouvelles valeurs plus élaborées (ex : SSR : Self-Selective Routing). Les paramètres élaborés sont, par exemple, des statistiques calculées pouvant être requises par la couche physique afin d'ajuster la puissance de transmission radio, ou par la couche réseau, afin de trouver des routes sans congestion ni collision.

$$STR = \frac{\text{SuccessfulTransmissions}}{\text{AttemptedTransmissions}}$$

STR (Successful. Transmission Rate) : indique la qualité des transmissions en donnant le rapport des transmissions réussies sur celles demandées.

$$CCR = \frac{\text{SuccessfulAttemptstoFindaClearChannel}}{\text{TotalAttemptstoFindaClearChannel}}$$

CCR (clear channel rate) : indique les possibilités de trouver le canal libre en donnant le rapport des tests réussis sur le nombre de tests.

De plus, nous souhaitons mettre en place une communication « cross-layer » horizontale : le système pourrait avoir une option de vue globale. C'est-à-dire qu'à chaque fois qu'un paquet est envoyé par un nœud, il lui ajoute une information locale, le nœud la recevant la récupère et la met dans une liste d'information globale du CL_entity. Ceci ne sera pas développé dans cette contribution, mais l'architecture est conçue pour en laisser la possibilité en perspective.

Ainsi, le CL_entity contiendrait trois listes :

- les informations locales brutes
- les informations locales avancées
- les informations globales

Cependant, il est important de citer les problématiques que cette extension peut poser : découvertes des CLME distants, droits de lire/écrire certaines données, risque de surcharge de sig dans le réseau, etc.

5. Architecture Orientée Web Services

5.1. Introduction

Dans la section 3 concernant les communications cross-layer, nous avons évoqué l'utilisation du service *QoS_Parameters*. Ce service fournit la spécification des besoins en QdS des applications multimédia, en termes de bande-passante, délai de bout en bout, gigue, taux de perte. Nous avons vu que cette spécification est utilisée par plusieurs mécanismes de QdS tels que le contrôle d'admission, le contrôle de débit, l'anticipation de requêtes de ressources. Afin que tout autre mécanisme puisse profiter également de ce service, il est nécessaire de le rendre visible et accessible dans le système de communication.

Il en va de même pour l'architecture cross-layer précédemment décrite. Cette architecture propose un service fournissant des paramètres mesurés et calculés (c.f. tableau métriques possibles) au sein de différents nœuds du réseau permettant ainsi l'optimisation de la communication. La problématique reste toujours la même : rendre ce service visible et accessible au reste du système de communication. En effet, nous pouvons nous retrouver dans un scénario d'interconnexion avec un autre segment réseau qui n'a pas connaissance de ces informations.

Afin de répondre à ce besoin, nous proposons dans ce chapitre l'instanciation d'une architecture orientée web services, permettant la publication, la découverte et l'invocation de ces services.

5.2. Présentation

Initialement, les services Web ont été introduits pour le déploiement d'applications de commerce électronique sur Internet. A l'heure actuelle, ils sont également utilisés dans d'autres contextes, comme par exemple le partage de ressources dans des environnements à large échelle. La popularité des services Web et leur utilisation dans ces contextes variés s'expliquent par le fait qu'ils peuvent opérer dans des environnements distribués particulièrement hétérogènes aussi bien du point de vue des plates-formes logicielles déployées que des modèles de programmation utilisés.

L'architecture Orientée Service (AOS [79]) proposée ici est un modèle d'architecture pour l'exécution d'applications logicielles réparties. Basée sur un modèle d'interaction applicative, cette architecture permet de mettre en œuvre des services avec une forte cohérence interne par l'utilisation de :

- un format d'échange universel (le plus souvent XML) ;
- un faible couplage par l'utilisation d'une couche d'interface interopérable.

Elle permet de décomposer une fonctionnalité en un ensemble de services, fournies par des composants et de décrire finement le schéma d'interaction entre ces services. Cet approche offre différents avantages tels que l'indépendance par rapport aux différentes plates-formes de communication utilisées (logicielles et matérielles), la réutilisabilité, la robustesse, etc.

La notion de ***Web service*** correspond à une application connectée à un réseau, capable d'interagir avec d'autres applications en utilisant des protocoles d'échange standardisés du W3C (World Wide Web Consurtium). Ainsi, les Web services peuvent opérer dans des environnements distribués particulièrement hétérogènes.

Les architectures orientées web services opèrent dans un environnement d'applications réparties. Dans un souci d'interopérabilité, le modèle des architectures orientée web services est organisé en couches (c.f. Figure 56).

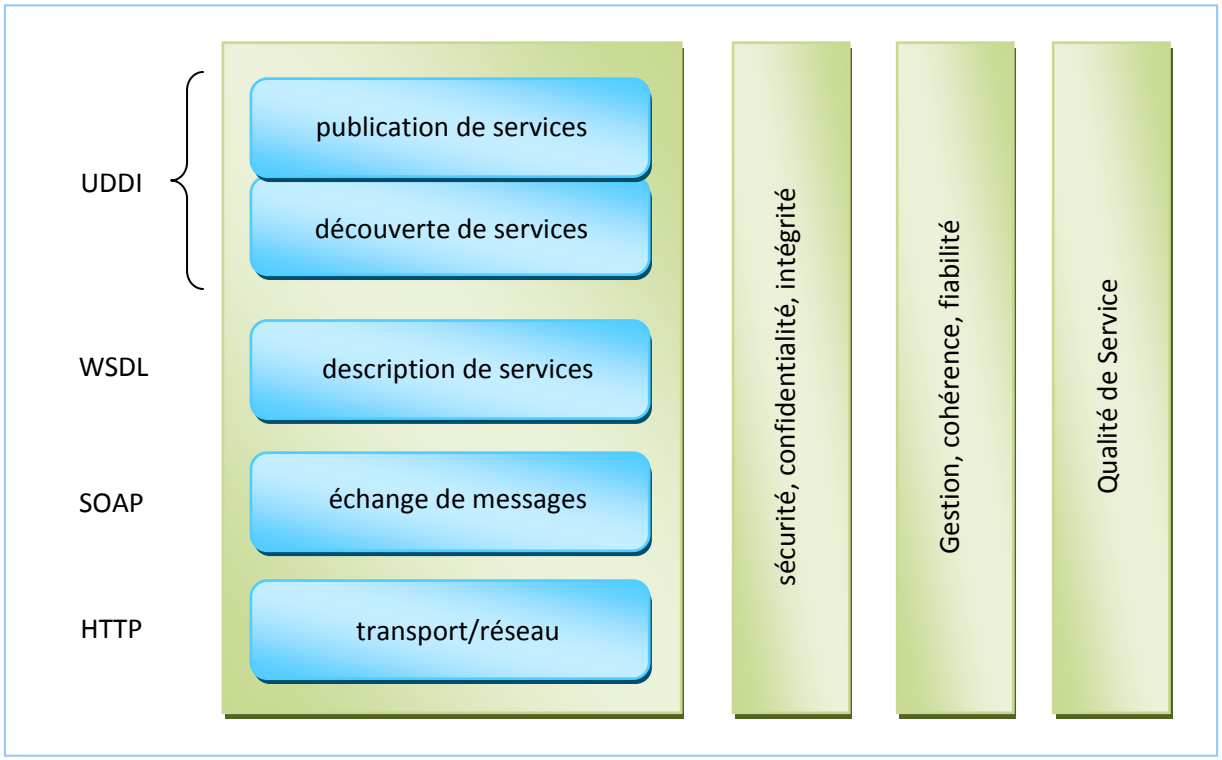


Figure 56 : structure en couche des web services

En dehors de l'architecture web services principale, une infrastructure permet d'assurer la sécurité (l'authentification, la confidentialité, l'intégrité des messages, …), la fiabilité des échanges (la cohérence, la durabilité, …), et la qualité de service (garantie, …).

Ces architectures orientées web services offrent un certain nombre d'avantages tels que :

- Les composants existants de l'architecture sont facilement réutilisables ;
- Il est possible de partager des applications entre des plates-formes et des environnements variés ;
- Le caractère évolutif permet l'ajout et la suppression de services afin de répondre aux nouveaux besoins fonctionnels d'applications ;
- Le mode « stateless » améliore la tolérance aux fautes : une défaillance (logicielle ou matérielle) n'entraine pas de perte d'information.

5.3. Fonctionnement

L'architecture orientée web services utilisée dans ce chapitre est présentée en Figure 57.

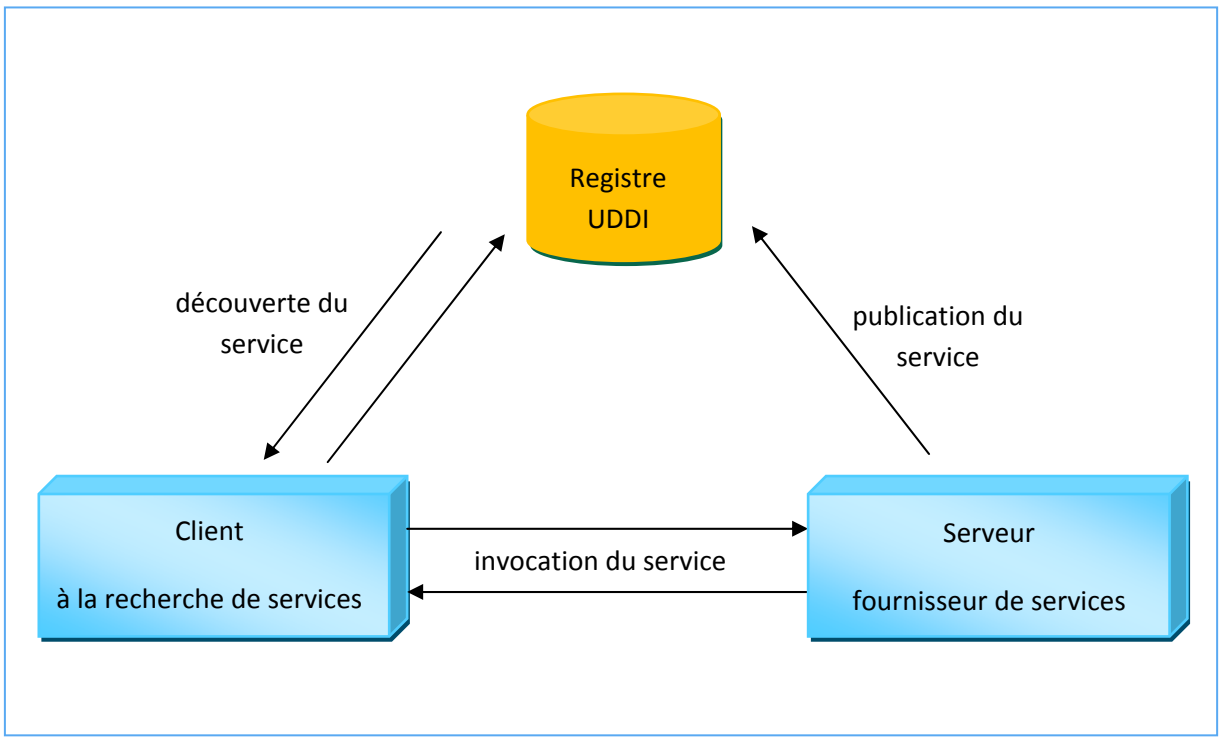


Figure 57 : Schéma d'interaction entre application cliente, serveur, et annuaire

Son fonctionnement est composé de plusieurs phases : la définition du service par le fournisseur, la publication du service par le fournisseur, la découverte du service par le client et l'invocation du service par le client auprès du fournisseur.

5.3.1. La définition du service : WSDL

Afin que le fournisseur de service puisse proposer son service, il est nécessaire de le définir. Nous utilisons le langage WSDL (Web Services Description Language) [80] pour décrire les interfaces fonctionnelles des services Web. Il est composé principalement de deux parties : abstraite et concrète. La partie abstraite définit une description fonctionnelle du service. La partie concrète contient les informations relatives au format de messages et au déploiement du service.

La partie abstraite d'un service est composée d'un ensemble d'opérations, regroupées pour former une interface définie par l'élément et caractérisé par un Type de Port. Chaque opération est caractérisée par ses paramètres d'entrée et de sortie, définie par l'élément message. La partie concrète de la spécification WSDL définit le format d'échange des paramètres des opérations et le point d'accès à partir duquel l'interface d'un service peut être appelée. Le point d'accès d'un service est défini par son URI qui désigne l'adresse du point d'accès, le chemin d'accès au service et son nom. Les parties concrètes et abstraites sont liées par une liaison définie par l'entité « Binding ».

La description d'un service comporte deux parties : l'interface et l'implémentation du service. L'interface constitue la partie réutilisable : elle contient la définition abstraite du service qui peut être instanciée et référencée par différentes définitions d'implémentations de service.

La représentation du service est effectuée avec le langage XML. XML « eXtensible Markup Language » [81] est un langage assurant la représentation de données et des documents structurés.

De ce fait, XML est alors flexible et extensible, et est devenu rapidement le standard d'échange de données sur le web.

La structure d'un document XML est souvent représentée graphiquement par une structure arborescente.

5.3.2. La publication de service : registre UDDI

Le fournisseur de service, après la programmation d'un service, publie dans le registre UDDI (Universal Description, Discovery, and Integration) [82] les services qu'il souhaite proposer avec également toutes les caractéristiques de celui-ci : le coût, les différentes garanties (sécurité, qualité de service, gestion, …). L'objectif principal du registre UDDI est de faciliter la recherche et la localisation des services web. UDDI définit des structures de données et généralement des interfaces APIs qui contiennent principalement deux types de fonctions. Les fonctions de publication pour publier les descriptions des services dans le registre (éventuellement la modification, ou la suppression des informations de l'annuaire). Les fonctions de recherche et d'interrogation pour interroger le registre afin de chercher des descriptions publiées des données techniques et administratives sur les services web et leurs fournisseurs. Comme les APIs de UDDI sont aussi spécifiées en WSDL avec une attache SOAP (décrit en section suivante), le registre peut être invoqué comme un service Web (en conséquence, ses caractéristiques peuvent être décrites dans le même registre, comme un autre service). Les requêtes et les réponses sont envoyées sous forme de message SOAP.

L'annuaire de service organise l'information sur les services en trois catégories :

- Les pages blanches : comportent les informations sur les fournisseurs de service (nom, coordonnées, description d'activités, etc.
- Les pages jaunes : la description au format WSDL des services déployés par le fournisseur de service.
- Les pages vertes : fournissent des informations sur le service fournis.
- L'annuaire est accessible de deux manières : via un navigateur WEB qui dialogue avec une application Web dédiée (interface spécifique à l'annuaire accédé), ou bien par un programme qui utilise une API définie par la spécification.

Dans la pratique, les annuaires UDDI sont encore très peu déployés et assez complexe d'usage. Leur utilisation n'est pas indispensable au fonctionnement des Web services. En effet, lorsque l'annuaire UDDI n'est pas spécifié, la description de service peut être publiée avec les moyens propres du fournisseur de service. Le cas le plus simple est un fournisseur qui envoie la description du service qu'il propose soit directement soit par une adresse URL où se trouve le fichier WSDL correspondant au service. Dans ce cas, le fournisseur ne peut pas modifier la localisation du WSDL sans perturber les clients (si le fournisseur modifie la localisation du WSDL, il doit mettre au courant tous ses clients).

5.3.3. Découverte du service

Un client à la recherche d'un service lance la recherche de celui-ci sur l'annuaire UDDI. Cette recherche est basée sur des critères de sélection concernant la qualité de service (coût, degré de

confiance, temps de réponse, niveau de sécurité, etc.). L'annuaire lui renvoie une réponse (plusieurs services susceptibles de répondre aux besoins du client) conforme aux critères spécifiés dans la requête. Le client sélectionne le meilleur service, et tente de récupérer sa description au format WSDL par le biais d'UDDI.

5.3.4. Invocation du service : SOAP

Une fois que le client a connaissance du service et de son fournisseur, une communication est établie entre client/serveur (requête/réponse) par le biais du protocole SOAP (Simple Object Access Protocol) [83]. SOAP est un protocole qui gère l'échange de messages avec les services Web, et en particulier les appels à des procédures distantes (RPC : Remote Procedure Call) induits par l'invocation d'opérations fournies par des services Web. Il assure donc l'interaction entre les services en transportant les paquets de données sous forme de texte structuré XML au sein d'une architecture distribuée orientée objet en utilisant le protocole HTTP. Le protocole SOAP est indépendant du protocole de transport.

Dans l'invocation d'objet distant à l'aide du protocole SOAP, le client SOAP crée un document XML qui contient les informations pour invoquer le service ; celui-ci est incorporé dans une enveloppe SOAP avant d'être transmis sous forme de requête POST HTTP (via une connexion HTTP standard vers le serveur). Le serveur SOAP reçoit le message, analyse sa cohérence et le dirige vers l'objet concerné. L'objet effectue le traitement et renvoie le résultat au serveur SOAP qui l'encapsule avant de le retourner au client. Le résultat est renvoyé au client au travers d'un document SOAP encapsulé dans un en-tête de réponse HTTP.

Pour notre cas d'application QdS, nous proposons de mettre en place un serveur appelé Media Type Repository (MTR), offrant comme service : **fournir les caractéristiques QdS des applications multimédia.**

Ces caractéristiques QdS sont obtenues à partir de sources diverses telles que l'IANA (Internet Assigned Numbers Authority) [84], ou encore l'ITU-T.

L'IANA est l'organisme responsable de la coordination globale des noms de domaines, des espaces d'adressage IP et AS (Autonomous System), et de bien d'autres ressources protocolaires Internet. Il gère également le format et l'encodage des corps de messages sur Internet MIME (Multipurpose Internet Mail Extensions) [85] [86]. A travers une série de standards RFC, il normalise les différents types de média audio et vidéo qui peuvent être utilisés sur Internet (ex. codec G.719 [87], codec H.261 [88]). Leurs caractéristiques y sont détaillées, et l'on retrouve en particulier le débit d'encodage de ces médias.

À partir du débit d'encodage du codec audio/vidéo, on peut déduire le débit moyen d'émission du flux multimédia : en ajoutant la taille des en-têtes des couches transport (RTP/UDP, DCCP, etc.), réseau (IP) et liaison (DVB-RCS) au débit d'encodage du média (données utiles).

On se base également sur les recommandations de l'organisme ITU [3] pour retrouver les seuils recommandés des métriques QdS de délai de bout en bout, de gigue, ainsi que le taux de perte.

Le serveur MTR stocke pour chaque type (audio, vidéo) et format (codec, ex. G711) de média les métriques QdS nécessaires dans le réseau satellite. Chaque entrée dans la base MTR contient trois parties pour l'utilisateur :

- La méthode permettant d'invoquer le service QoS_services est : qos_parametres
- Les paramètres d'entrée sont : le type d'application et le codec (sous forme de chaînes de caractères)
- Les sorties : les caractéristiques de QoS correspondants au type d'application et au codec en entrée :
    - peakBitRate en (kbps)
    - maxJitter(en ms)
    - maxLoss (en %)
    - maxDelay (en ms)

5.3.5. Conclusion

Nous avons présenté ici une architecture orientée web services qui permet à la fois :

- la définition du service par le fournisseur, à travers la description des interfaces fonctionnelles du service, en utilisant le langage WSDL ;
- la publication du service par le fournisseur dans un registre UDDI ; cette publication facilite la recherche et la localisation de services ;
- la découverte du service par le client ;
- et l'invocation du service par le client auprès du fournisseur.

Cette architecture permet au final de rendre les différents services visibles et accessibles au système de communication. C'est donc une approche compatible avec les architectures IMS () où l'on est censé pouvoir interconnecter une grande variété de réseaux.

5.4. Conclusion des contributions

Les trois précédents chapitres de ce mémoire ont présentés les contributions réalisées durant les travaux de cette thèse.

Dans la première partie des contributions, nous avons proposé trois solutions basées sur une approche cross-layer à déployer dans un réseau satellite DVB-S2/RCS. Ces solutions permettent à la fois de : (i) satisfaire les besoins en QdS des applications multimédia interactives sur un réseau satellite ; (ii) optimiser les ressources de la voie montante de ce réseau satellite par l'utilisation d'allocation dynamique.

Dans la seconde partie des contributions, nous avons spécifié et développé une architecture dédiée aux communications cross-layer pour les réseaux sans fil. Cette architecture offre un cadre de communication inter-couche mais également le stockage des données à partager. Cette approche permet de garder une compatibilité avec l'architecture de communication initiale tout en favorisant une évolution continue de l'architecture cross-layer.

Dans la dernière partie des contributions, nous avons proposé et développé une architecture orientée web services permettant la publication, la découverte et l'invocation de tous les services proposés par les contributions précédentes. Cette architecture met ainsi à disposition l'ensemble des services disponibles sur le réseau.

Le prochain chapitre présentera alors l'évaluation des différentes contributions proposées.

6. Évaluations

Nous proposons dans ce chapitre d'évaluer chacune des contributions précédemment présentées. Pour chacune des contributions, nous décrirons le contexte d'expérimentation utilisé, puis les scénarii d'évaluation, et enfin présenterons et analyserons les résultats obtenus.

Ne disposant pas d'un environnement satellite DVB-S2/RCS réel, les contributions basées sur les communications cross-layer ainsi que l'architecture orientée web services sont évaluées sur une plate-forme d'émulation satellite (c.f. section suivante). Cependant, les expérimentations concernant l'architecture cross-layer sont menées dans un environnement wifi 802.11 réel.

Nous allons donc commencer par décrire le contexte expérimental lié à l'environnement satellite.

6.1. Contexte expérimental

Tout comme pour l'état de l'art de la Qualité de Service, nous présentons, à travers les différentes couches des systèmes de communication, le cadre d'expérimentation pour les tests effectués sur les réseaux satellite : de la couche physique jusqu'à la couche application.

6.1.1. Couches Physique-MAC: plate-forme d'émulation satellite

L'un des objectifs du projet européen SatSix [42] fût la définition et la conception d'une plate-forme d'émulation d'un système satellite DVB-S2/RCS, appelée « Platine ». Cette plate-forme permit l'implémentation et l'évaluation de l'architecture QdS proposée dans le projet.

Dans cette plate-forme d'émulation, chaque élément du segment satellite (ST, GW et NCC) est émulé par un ordinateur distinct, sous Linux, déployé sur un réseau Ethernet 100 Mbps. La pile protocolaire (couche physique, liaison et réseau) est implémentée dans l'environnement Margouilla [89]. Basé sur une licence LGPL (Lesser General Public License), cette environnement offre un cadre de développement en langage C++ et d'exécution, indépendant du système d'exploitation. Un gestionnaire d'évènement complet est défini, comprenant l'échange de messages, threads, timers, évènements Socket et traces. Il permet ainsi de spécifier et développer complètement un protocole avant d'en évaluer les performances.

Ainsi, à partir de cet environnement Margouilla, les blocs suivants ont été implémentés :

- « **IP QoS** » : implémente les mécanismes de différentiation et de conditionnement de trafic au niveau IP ;
- « **IP Dedicated** » : effectue la segmentation et le réassemblage des données ;
- « **DAMA** » : implémente les algorithmes de gestion de ressources satellites (requête/réponse) au niveau MAC ;
- « **DVB-S2/RCS** » : construit les trames DVB-S2 et DVB-RCS avec les cellules ATM provenant de la couche IP Dedicated sous le contrôle du bloc DAMA ;
- « **Satellite Carrier** » responsable de l'émulation des différentes porteuses, du délai et des erreurs du lien satellite.

En ce qui concerne le dernier, l'émulation du lien satellite a été développée dans le cadre du projet IST Brahms [90] et permet de prendre en compte de multiples paramètres. Un ordinateur est dédié SE (Satellite Emulator). Ce dernier émule plusieurs systèmes satellites en permettant la configuration du délai, de la gigue, du profil des pertes, du type de satellite et du type de connectivité. Les travaux

de thèse de [91] donnent de plus amples renseignements sur cette plate-forme d'émulation systèmes satellite.

Pour nos expérimentations, nous nous plaçons dans le cadre de réseaux d'accès Internet par satellite, par conséquent le GW est couplé avec le NCC. L'émulation satellite choisie est la suivante : un satellite géostationnaire régénératif DVB-S2/RCS monospot, la liaison montante en Ka MF-TDMA et la descendante en Ku TDM.

La supertrame est composée de 10 trames, chacune de durée 50 ms. Les cellules ATM sont de 53 octets. Les requêtes de ressources sont émises en début de chaque supertrame, soit toutes les 500ms. Il en va de même pour l'émission du plan d'allocation TBTP. Le débit de transmission maximum du ST est fixé à 1024 kbps. Le mécanisme d'allocation du ST utilise les requêtes dynamiques de ressources RBDC.

Le délai de propagation du lien satellite est fixé à 250ms avec une gigue égale à ±1ms et le profil d'erreur est typique de conditions météorologiques favorables, ce qui signifie qu'il n'y a pas de pertes de paquets dues à des erreurs de transmissions (grâce à l'apport des formes d'ondes adaptatives S2/RCS).

Nous supposons qu'en dehors du réseau satellite, une liaison filaire est utilisée pour transporter les données (c.f.Figure 58) :

- entre le premier utilisateur jusqu'au terminal satellite (ex. réseau ethernet 10/100Mbps) ;
- entre la Gateway du réseau satellite jusqu'au second utilisateur (ex. réseau xDSL 20Mbps).

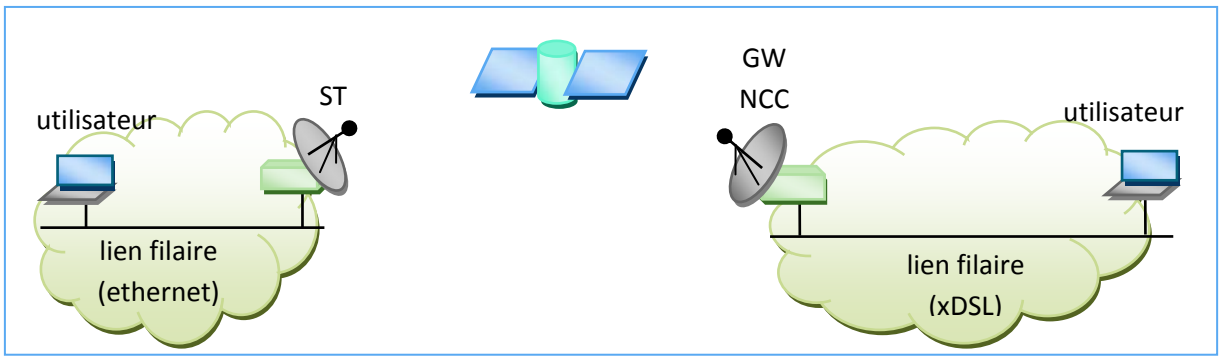


Figure 58 : composition réseau satellite et lien filaire

On suppose que ces liens filaires sont suffisamment surdimensionnés en capacité pour négliger leur temps de transfert par rapport au lien du segment satellite. Par conséquent, notre scénario peut s'abstraire à un utilisateur derrière un ST, et l'autre utilisateur derrière le Gateway.

6.1.2. Couche IP : DiffServ

La plate-forme Platine dispose du bloc « IP QoS » offrant au terminal satellite les fonctionnalités de routeur de bordure DiffServ (classification, différentiation et conditionnement de trafic IP). Selon les besoins, nous déciderons d'activer ou pas ces fonctionnalités IP DiffServ. Les protocoles IPv4 et IPv6 sont également à disposition.

6.1.3. Couche transport : DCCP, UDP

Pour les applications multimédia interactives, nous avons évoqué précédemment que le protocole de transport DCCP serait le plus approprié pour transporter les données utiles (c.f. section 1.3.4).

Cependant, les implémentations actuelles de DCCP ne sont pas complètement fonctionnelles pour être utilisées dans ces expérimentations. Nous proposons alors d'utiliser le protocole UDP.

6.1.4. Couches session-application : SIP, outils de mesure

Dans le sous chapitre 1.5, nous avons vu que les fonctionnalités de la couche session sont réparties entre la couche transport (ex. protocole TCP) et la couche application (ex. protocole SIP). Les implémentations de la pile SIP sont en général directement intégrées aux applications multimédia et proxys SIP. Nous proposons de mettre en place une plate-forme SIP locale pour les expériences. Elle est composée de clients SIP et de proxys SIP, avec chaque proxy gérant un domaine (ex. laas.fr).

Clients SIP

Pour les clients SIP, nous utilisons l'application « ***minisip*** » [92]. Cette application a été développée dans le cadre de thèses effectuées à la « Royal Institute of Technology » (KTH), à Stockholm en Suède. Elle permet d'effectuer des appels téléphoniques (VoIP), de la messagerie instantanée, ou encore de la visiophonie (c.f. Figure 59). Elle utilise deux librairies extérieures : OpenSSL pour le transport sécurisé et GladeMM pour l'interface graphique. Le code source de *minisip* est sous licence GNU LGPL (Lesser General Public License).

Figure 59 : application minisip sur PocketPC

Proxys SIP

Pour les proxys SIP, nous choisissons d'utiliser l'application « ***partysip*** » [93]. C'est une application modulaire dont les fonctionnalités sont gérées par des plugins (ajout/suppression). Elle peut être utilisée comme « SIP registrar » (serveur d'enregistrement SIP), « SIP redirect server » (serveur de redirection SIP) et « SIPproxy server » (proxy SIP). Partysip est entièrement développé en langage C, et dépend d'une unique librairie extérieure « oSIP » qui est une implémentation de bas niveau du protocole SIP. Le code source de *partysip* est également sous licence GNU LGPL.

La simplicité d'implémentation de ces deux applications permet une prise en main relativement aisée du code source, afin d'y porter les modifications nécessaires pour les expérimentations.

6.2. Contexte de mesure

Les contributions présentées dans cette thèse sont soumises à la fois à :

- des tests fonctionnels qui permettront de vérifier le bon fonctionnement du déploiement des contributions dans l'architecture de communication actuelle. Pour cela, nous utiliserons notre plate-forme multimédia SIP, avec les logiciels *minisip* et *partysip*.
- des tests de performance qui évaluent la conformité des solutions par rapport aux exigences des clients et des opérateurs réseau. Il s'agit par exemple, de la mesure du délai subi par les données, ou encore de la charge du lien retour du réseau satellite.

À ce jour, il n'existe pas d'applications multimédia intégrant des outils de mesure efficaces. La mesure idéale serait celle du délai perçu par l'utilisateur final, lorsque la donnée multimédia lui est présentée (visuellement et/ou auditivement). Pour cela, il serait nécessaire de modifier l'implémentation d'une application multimédia existante afin d'effectuer la mesure.

La plupart des outils de mesure, tels que « tcpdump » ou encore « ethereal » sont décorrélés de l'application elle-même. Elles permettent d'obtenir des mesures au niveau réseau.

Sachant que les applications multimédia actuelles n'implémentent pas de mécanismes leur permettant d'adapter dynamiquement leur débit d'émission en fonction de l'état du réseau, leur comportement peut être reproduit assez facilement. Afin de se rapprocher le plus possible de ce comportement, et par conséquent de la mesure applicative, nous pouvons utiliser des outils de rejeu de trafic.

Le groupe de recherche « Video Traces Research Group » de l'Université d'Arizona (Arizona State University) met à disposition plusieurs fichiers de traces de communications multimédia telles que la vidéo à la demande, ou encore la visioconférence, sous différents formats [94]. En intégrant ces fichiers de traces dans un outil de rejeu de trafic, il est ainsi possible de reproduire le comportement des applications multimédia.

Pour nos expériences, nous proposons l'utilisation d'un outil de capture, rejeu et mesure, développé au LAAS-CNRS, appelé « FL3 » (Floc : Flow Capture, Flore : Flow Replay, Flan : Flow Analyzer) [91].

6.2.1. Outils de mesure FL3

L'utilisation de la série d'outils FL3 se décompose en plusieurs phases :

- La première phase consiste à capturer, au niveau réseau, le trafic d'une communication entre des applications réelles. Cette phase doit s'effectuer dans les meilleures conditions réseau possibles (réseau surdimensionné), afin que la capture soit le plus proche possible du comportement nominal de l'application.
- Puis dans un second temps, les applications sont remplacées dans le contexte d'expérimentation par les outils de rejeu. Ces derniers vont reproduire les différents flux de la communication capturée, avec des paquets contenant les informations nécessaires de mesure (ex. estampille, taille). L'outil de rejeu nécessite une synchronisation temporelle entre les nœuds de rejeu concernés. Nous utilisons le protocole NTP (Network Time Protocol [95]). Initialement il permet de synchroniser les horloges des systèmes informatiques à travers le réseau Internet, dont la latence est variable.

- Ainsi, la dernière phase permet d'analyser les communications rejouées afin d'en tirer des métriques telles que le délai de bout en bout, la gigue, le taux de perte, le débit de transmission.

6.2.2. Outils de traçage Gnuplot

Afin de tracer des courbes pour ces différentes métriques, nous utilisons l'outil ***gnuplot*** [96]. C'est un utilitaire libre de traçage de courbes de fonctions et de données. Il a été initialement conçu pour les étudiants et scientifiques qui souhaitaient visualiser leurs fonctions et données mathématiques. Il permet de tracer des courbes en 2D, 3D, avec des points, lignes, boites, surfaces, contours, champs vectoriels.

Pour certains tracés, il inclut des options d'interpolation et d'approximation de données (**smooth option**). Par exemple, l'option ***smooth bezier*** effectue une approximation des données avec une courbe Bezier de degré n (le nombre de points de données). Dans certains cas, nous utiliserons cette option afin d'éviter d'obtenir des courbes surchargées de points, illisibles.

6.3. Amélioration de l'allocation dynamique en début de session multimédia

Dans cette section, nous allons présenter l'évaluation de la contribution concernant l'allocation dynamique de ressources dans un système satellite lorsqu'une session démarre.

6.3.1. Évaluation de performance du cas de référence

Nous proposons d'observer les performances de l'allocation dynamique et statique dans le réseau satellite pour un flux de type CBR (Constant Bit Rate). L'algorithme d'allocation dynamique utilisé est simple : la quantité de ressources à requérir est égale à la quantité de données contenue dans le tampon d'émission du ST. Pour cela, nous lançons un générateur de trafic, à 100kbit/s, dont le flux traverse la voie retour du système satellite. L'ensemble des ressources du réseau satellite sont disponibles, à savoir, chaque requête de ressources est satisfaite.

La Figure 60 permet de dégager un premier constat sur la latence du mécanisme d'allocation dynamique de ressources, lorsque ce dernier ne possède aucune amélioration particulière. Elle montre l'évaluation de l'allocation statique qui utilise les requêtes CRA (courbe rouge (carré)), et dynamique avec les requêtes RBDC (courbe verte (rond)). Nous mesurons le délai de transmission des premiers paquets, sur une période de 2 secondes. Les paquets sont émis quelques millisecondes après le début de l'expérience.

Nous observons que le temps de transmission des premiers paquets soumis à l'allocation statique est très proche du délai de propagation du lien retour satellite (250ms à 280ms). Il y a donc très peu d'attente dans la file d'émission du ST pour l'allocation statique.

En ce qui concerne l'allocation dynamique, les premiers paquets sont soumis à un délai de transmission de 1500ms (au moins 1200ms), ce qui correspond à :

- calcul de la requête au sein du ST (30ms) ;
- 250ms pour la traversée de la requête sur le lien satellite (temps de propagation) ;
- traitement de l'allocation au sein du NCC (30ms) ;
- 250ms pour la traversée de la réponse d'allocation, TBTP ;
- traitement de l'allocation reçue au sein du ST (30ms) ;

À ces délais, il faut rajouter un délai d'attente supplémentaire des données dans le ST avant l'envoi de la prochaine requête d'allocation (ce processus n'étant pas synchronisée avec l'arrivée des données) : variable entre 0 et 1 seconde.

Sachant que les requêtes de ressources sont satisfaites, le délai de transmission des paquets diminue et atteignent un niveau optimal (280ms) au bout de 1500ms. Cela correspond à environ un cycle de requête/allocation de ressources. Il est important de noter que le délai maximal recommandé pour les applications multimédia que nous utiliserons est de 400ms pour un fonctionnement correct [3].

Rappelons que les ressources du réseau satellite sont entièrement disponibles durant les expériences. En effet, cette configuration permet déjà de dégager les problématiques qui nous intéressent.

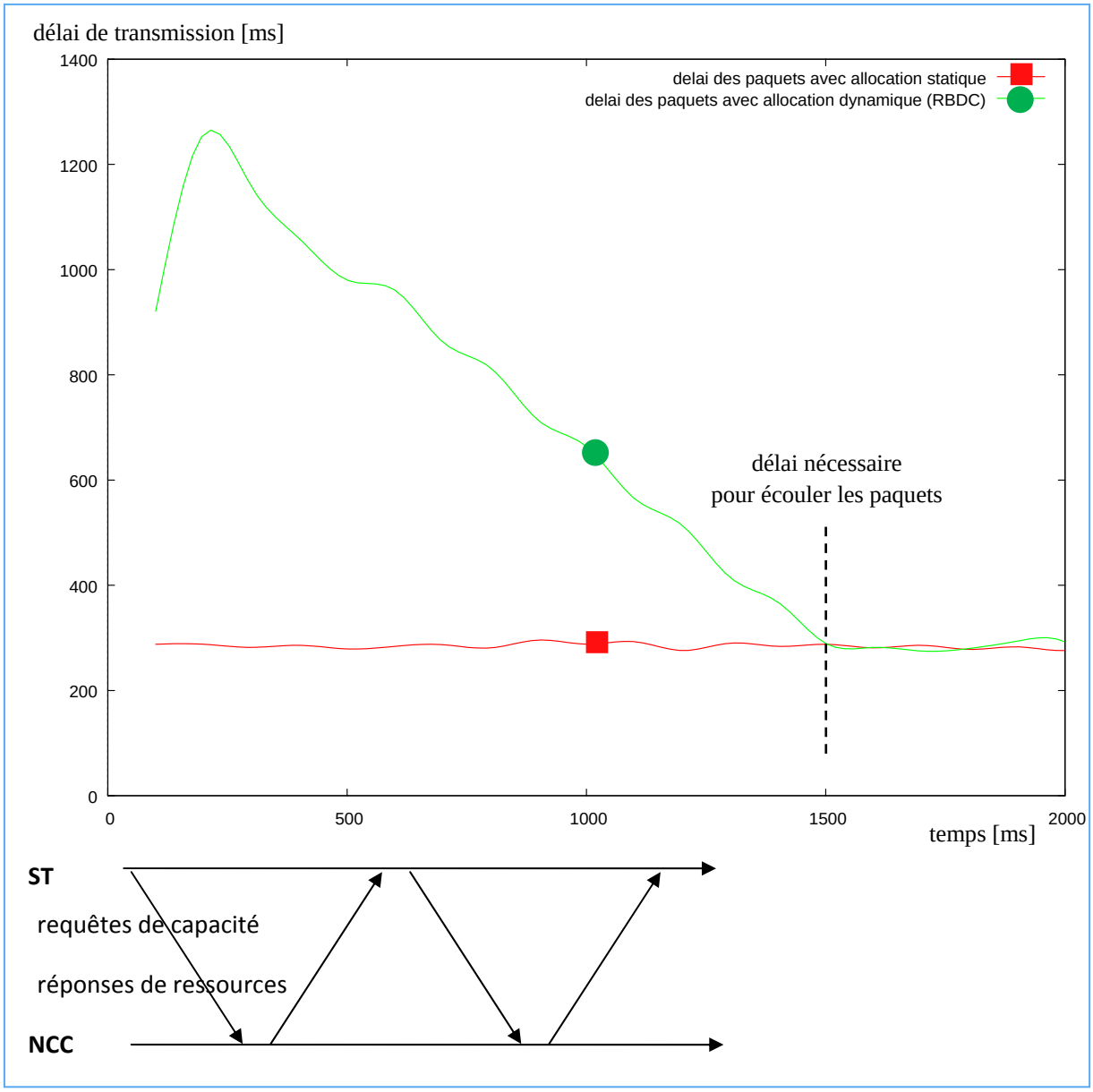


Figure 60 : délai d'attente subi par les données avec allocation dynamique sans amélioration

6.3.2. Scénarii avec amélioration

Face à ce constat, nous proposons maintenant d'évaluer l'amélioration apportée par l'interaction cross-layer au sein du NCC proposée dans la contribution. Elle permet d'effectuer la requête de ressources anticipée pour le ST concerné par le flux de données.

Le proxy SIP et la base de données MTR sont installés du côté NCC dans le réseau satellite.

Pour le proxy SIP, le code source est modifié pour y rajouter le mécanisme de requête de ressources inspiré de celui d'un client DAMA. L'interface vers le NCC est créée à l'aide d'un socket réseau permettant la transmission de la requête de ressources.

Au niveau de l'émulateur satellite Platine, le bloc émulant la couche du serveur DAMA du NCC est modifié afin de prendre en compte la requête provenant du proxy SIP. L'interface vers ce dernier

utilise également un socket réseau, pour transmettre l'acquittement au proxy SIP. Nous évaluons plusieurs scénarii :

scénario 1. le proxy SIP modifié effectue la requête anticipée de ressources, et le NCC ne transmet pas d'acquittement au proxy SIP (c.f. Figure 61). Par conséquent le proxy SIP libère le message SIP OK après avoir émis la requête cross-layer. Or, sachant que la signalisation SIP est indépendante de la période d'émission du plan d'allocation qui est en générale de 500ms, deux cas peuvent survenir :

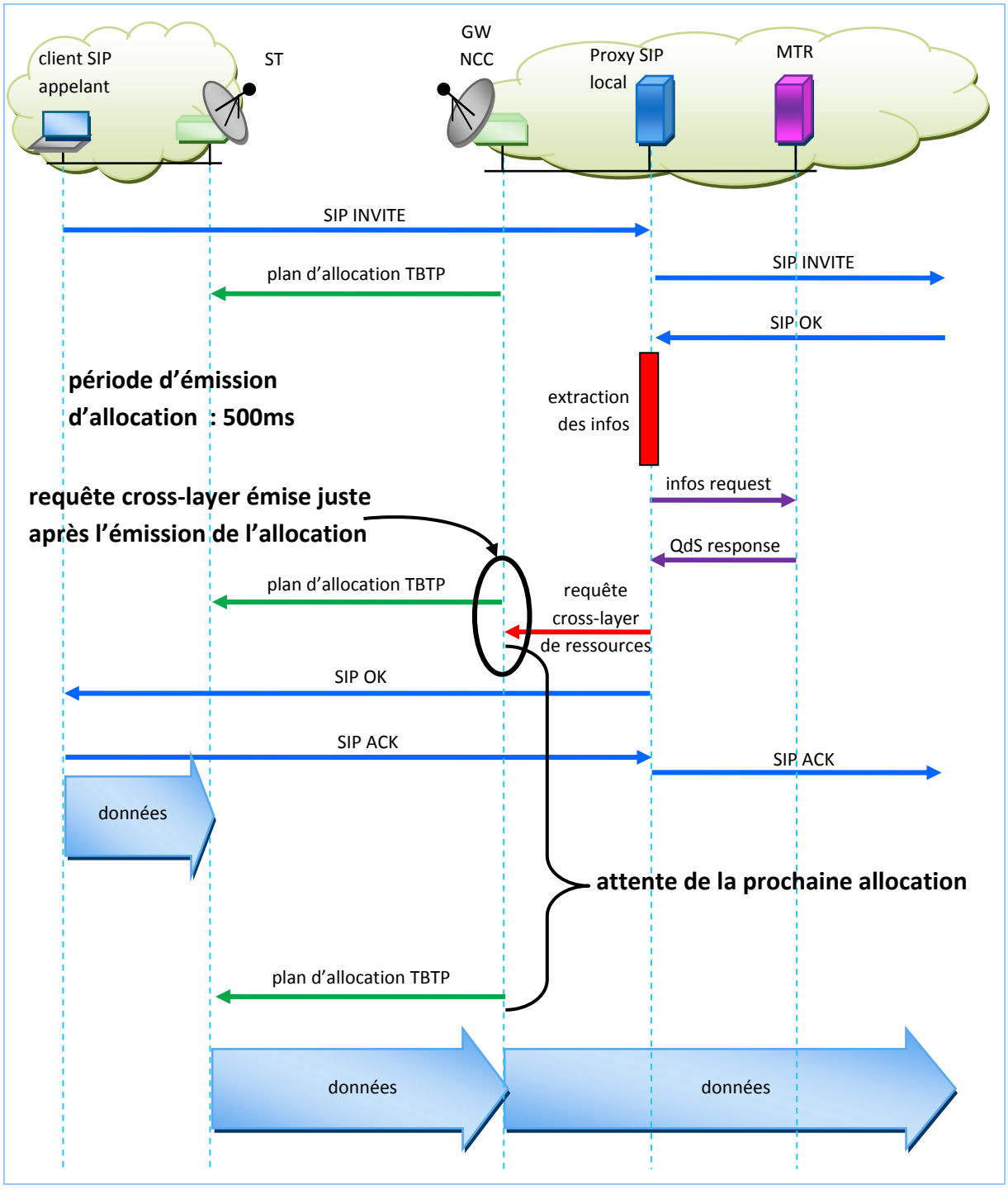


Figure 61 : scénario de requête cross-layer avec période d'émission allocation à 500ms

1.1. la requête cross-layer anticipée de ressources est émise juste après l'émission du plan d'allocation de ressources. Notre requête cross-layer est donc prise en compte au prochain calcul d'allocation de ressources par le NCC, soit 500ms plus tard. C'est le pire cas.

1.2. la requête cross-layer est émise juste avant le calcul d'allocation de ressources. Notre requête est alors prise en compte et ne subit pas la période d'émission du plan d'allocation.

scénario 2. Ici, la fréquence d'émission du plan d'allocation de ressources est réduite à 50ms. Par conséquent le pire cas rajoutera 50ms au délai total de transmission des données multimédia (c.f. Figure 62). Cependant une signalisation trop fréquente du plan d'allocation a tendance à surcharger la voie aller du réseau satellite.

scénario 3. La fréquence d'émission du plan d'allocation revenue à 500ms, mais cette fois-ci, nous utilisons notre mécanisme cross-layer d'acquittement : le NCC signale le proxy SIP l'envoi du plan d'allocation (c.f.Figure 63). Ainsi quelque soit la période d'émission du plan d'allocation, le proxy SIP sera toujours en mesure de transmettre le message SIP OK au même moment de l'envoi de l'allocation de ressources au ST concerné.

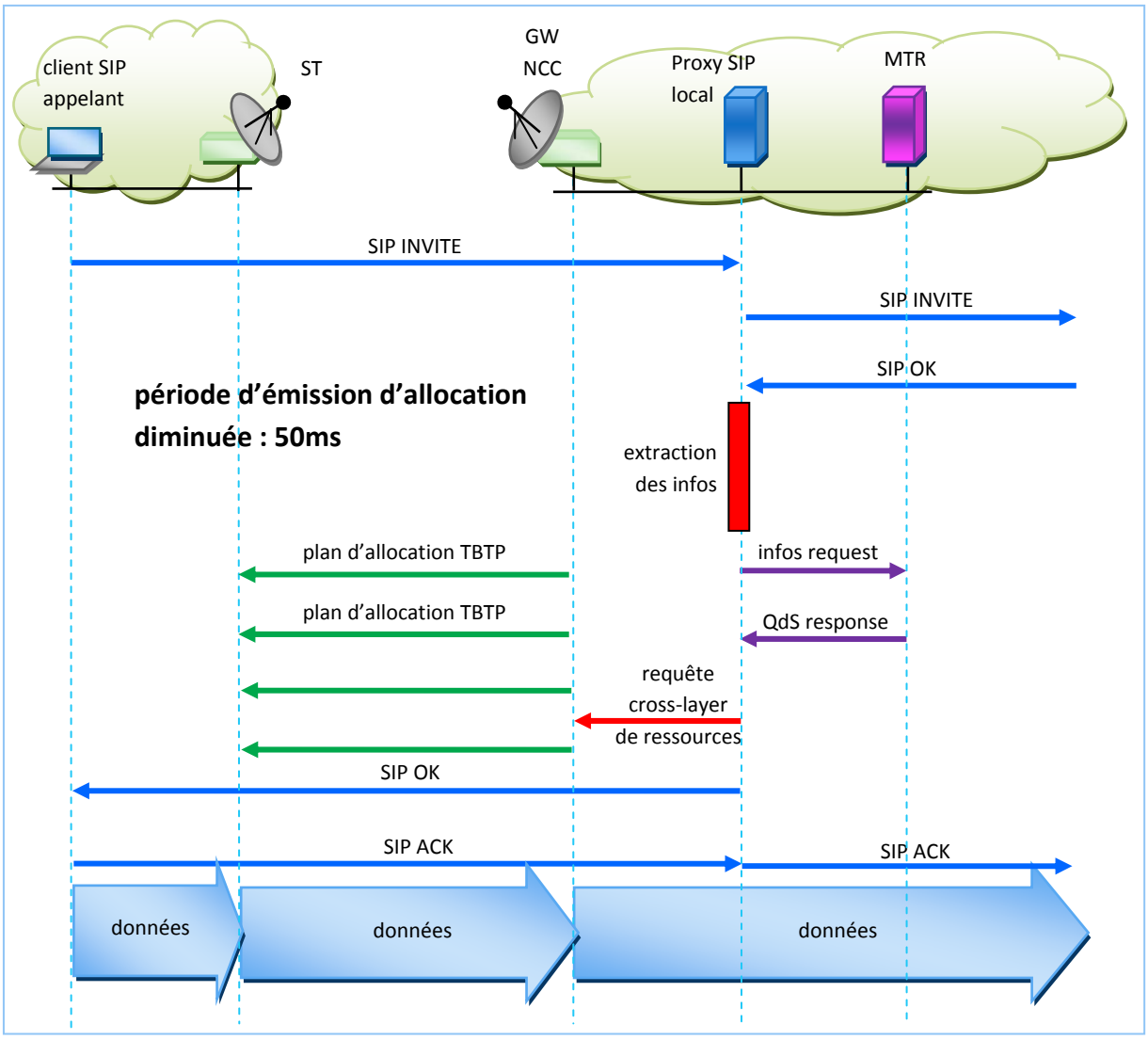


Figure 62 : scénario de requête cross-layer avec période d'émission plan d'allocation à 50 ms

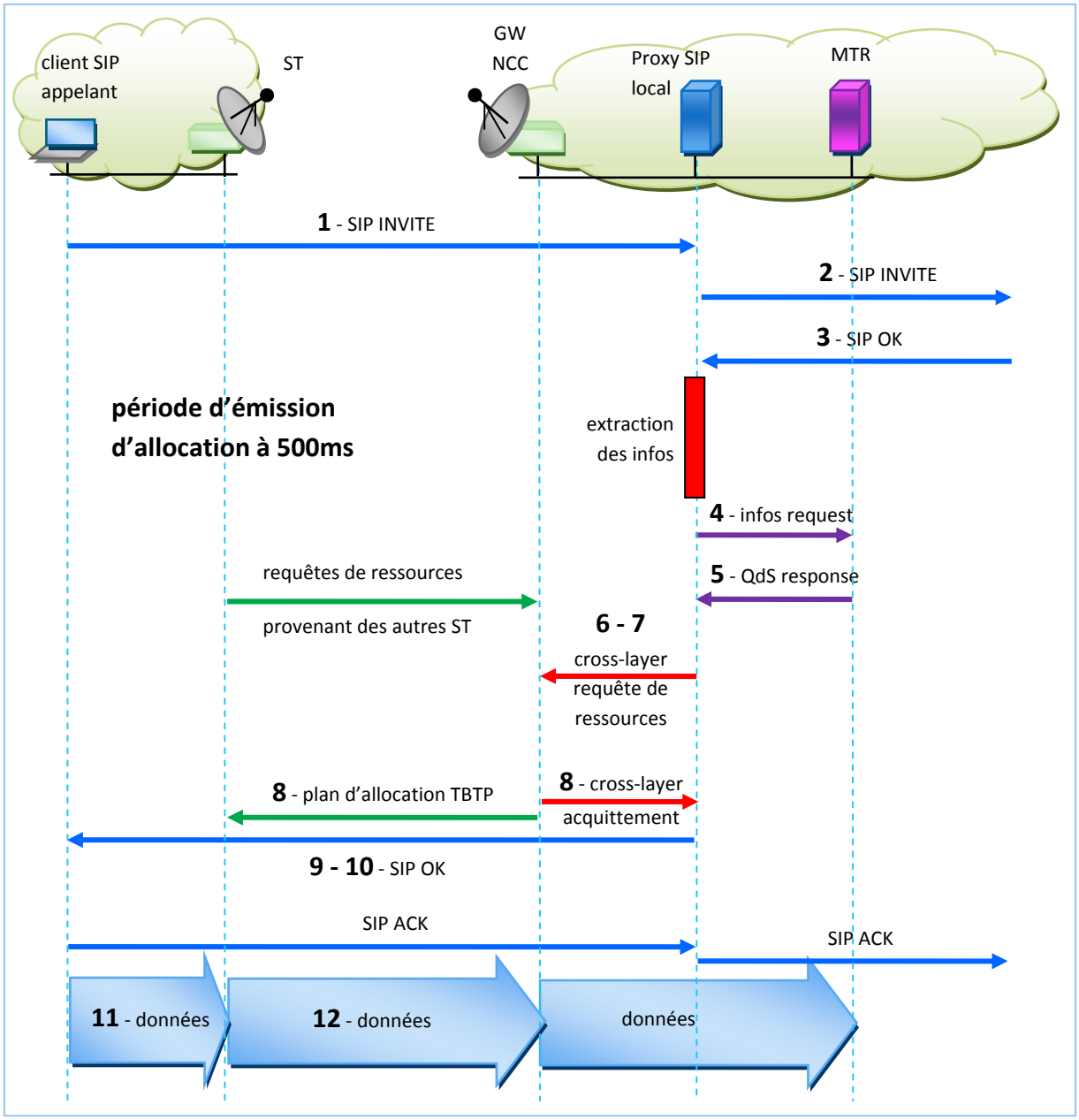


Figure 63 : scénario de requête cross-layer avec signalisation d'acquittement cross-layer

6.3.3. Validation fonctionnelle

Nous validons tout d'abord l'aspect fonctionnel de la communication cross-layer. Nous vérifions le bon établissement de la session multimédia SIP avec la mise en place des interactions cross-layer (requête de ressources + signal d'acquittement).

L'objectif de l'acquittement cross-layer (NCC vers proxy SIP) est de retarder l'émission du message SIP OK jusqu'à l'émission du plan d'allocation par le NCC. Nous observons que, dans le pire cas, le message SIP OK est retenu par le proxy SIP durant 500ms. Ce délai se manifeste du côté client SIP par une durée de tonalité d'appel plus longue. Dans certains cas, ce délai peut paraitre long aux clients SIP. Ces derniers, pour fiabiliser l'établissement de la session, retransmettent les messages de session SIP INVITE et SIP OK. Ces retransmissions de messages sont alors gérées par le proxy SIP

modifié en les ignorant. Ceci permet d'éviter de retransmettre de nouvelles requêtes cross-layer de ressources au NCC.

6.3.4. Évaluation de performance de la contribution

En ce qui concerne les tests de performance, nous mesurons le délai moyen de transmission subi par les premières données multimédia au niveau applicatif. Nous utilisons l'outil FL3 pour cette validation. Le délai mesuré est alors celui entre l'émission des données par l'application émettrice et leur réception par l'application réceptrice. Les quatre scénarios précédents sont évalués et analysés, avec le cas de référence qui représente le délai subi sans amélioration.

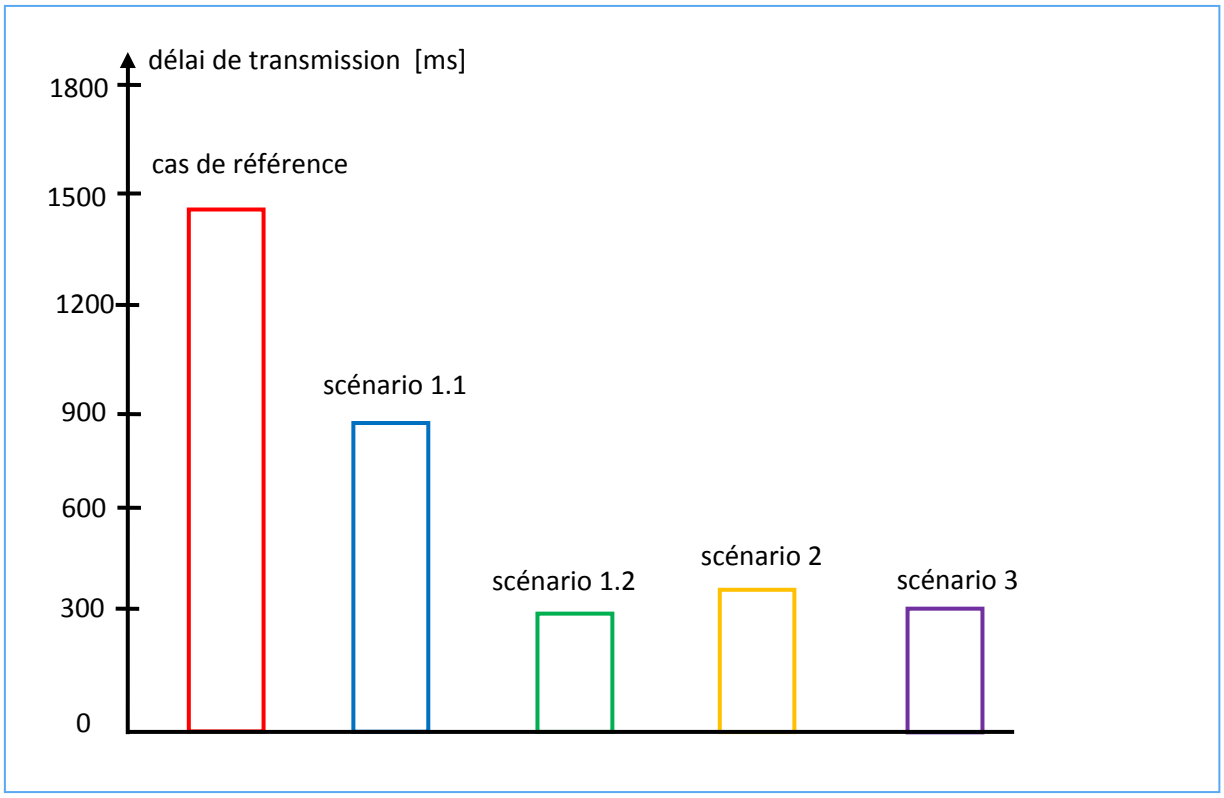


Figure 64 : délai de transmission des données multimédia

Pour le cas de référence, nous observons un délai de transmission de 1450ms ce qui correspond à 500ms (mise en attente de la prochaine requête) + 500ms (cycle requête/allocation) + 250ms (transmission des données). Ceci est le pire cas, dans la mesure où les données arrivent dans la file d'émission du ST juste après que ce dernier ait émis une requête de ressources. Le client DAMA du ST ne prend donc pas en compte ces données multimédia dans sa requête. Il faut donc attendre la prochaine période d'émission de requête. Puis le cycle de requête/allocation, et enfin la transmission des données.

Pour le scénario 1.1, cross-layer sans acquittement, avec une période d'émission du plan d'allocation de 500ms, le délai moyen de transmission est de 850ms, qui correspond à 500ms pour l'attente de la prochaine période d'émission du plan d'allocation, + 250ms (transmission des données). Dans le scénario 1.2, le délai est de 250ms, car notre requête a pu être prise en compte juste avant l'émission du plan d'allocation.

Dans le scénario 2, cross-layer sans acquittement, avec une période d'émission du plan d'allocation de 60ms, le pire cas est observé, avec un délai moyen de transmission proche de 300ms : 50ms (période d'émission du plan d'allocation) + 250ms (transmission des données).

Dans le scénario 3, avec acquittement du NCC et une période d'émission du plan d'allocation de 500ms, nous observons, dans tous les cas, un délai de transmission des premières données multimédia de 250ms. Ce mécanisme permet de supprimer toute attente des données dans la file d'émission du ST, et ceci, quelques soit les périodes d'émission des requêtes et des allocations de ressources.

6.3.5. Conclusion

Les résultats présentés montrent l'intérêt de communication cross-layer au démarrage d'une session multimédia au sein du système satellite DVB-S2/RCS. Une communication cross-layer à double sens est nécessaire :

- Dans le sens couche session (proxy SIP) vers la couche liaison (client DAMA) afin d'effectuer la requête anticipée de ressources sans avoir à subir la latence due à la traversée du lien satellite ;
- Dans le sens inverse afin de synchroniser l'obtention effective des ressources par le ST et le démarrage de la session multimédia.

Cette communication cross-layer réduit significativement le délai de traversée du lien retour satellite à 250ms pour les premières trames de données multimédia.

6.4. Amélioration de l'allocation dynamique en cours de session multimédia

Cette seconde évaluation porte toujours sur le mécanisme d'allocation dynamique, mais cette fois-ci durant la communication multimédia, sur la voie retour du réseau satellite.

Le mécanisme d'allocation dynamique de ressources du ST utilise un algorithme d'anticipation de ressources afin de répondre aux futurs besoins du flux multimédia en régime stationnaire. Un facteur de pondération « α » permet d'ajuster la quantité de ressources anticipée à requérir auprès du NCC. Ce facteur α est à la base fixé par l'opérateur réseau. Notre contribution consiste à rendre dynamique ce facteur α en fonction de l'évolution du trafic multimédia. Il est calculé à partir d'informations de session fournies par notre mécanisme de communication cross-layer au sein du ST. Nous proposons d'évaluer l'amélioration apportée par ce facteur d'anticipation α.

6.4.1. Scénario d'expérimentation

Pour cette expérimentation, nous installons le proxy SIP du côté ST, dans le réseau local (c.f. Figure 65). Le code source du proxy SIP est modifié pour intégrer l'une des interfaces de la communication cross-layer.

L'autre interface est intégrée au code source du client DAMA du ST. L'algorithme du client DAMA est également modifié afin d'y intégrer le calcul du facteur d'anticipation dynamique nécessaire pour le calcul des requêtes de ressources auprès du serveur DAMA du NCC.

Nous plaçons également du côté du ST, la base de données MTR qui fournit la spécification des besoins en QdS de la communication multimédia (ex. débit, délai max). Ainsi, cette topologie évite la latence de la traversée du lien satellite (requête/réponse MTR + transmission cross-layer).

Les clients SIP établissent une communication multimédia audioconférence en utilisant le codec G.711 avec un débit de transmission de 64 kbit/s.

6.4.2. Validation fonctionnelle

Nous vérifions tout d'abord le bon fonctionnement de la communication cross-layer entre le proxy SIP et le mécanisme client DAMA du ST (niveau MAC).

Comme décrit dans la contribution (c.f. Figure 65), à la réception du message SIP OK, le proxy SIP modifié extrait les informations de codec de session et les utilise pour interroger la base de données MTR. Ce dernier répond avec le débit de transmission associé au codec.

Le proxy SIP transmet cette information au client DAMA du ST par le biais de la communication cross-layer proposé dans la contribution.

Tout comme dans la contribution précédente, les éventuelles retransmissions de messages d'initialisation de session (SIP INVITE et SIP OK) des clients SIP sont gérées par le proxy SIP modifié, afin de ne pas être pris en compte.

Le client DAMA utilise l'information du débit de transmission du codec attendu afin de calculer dynamiquement le facteur d'anticipation α utilisé dans la formule de requêtes dynamiques de ressources.

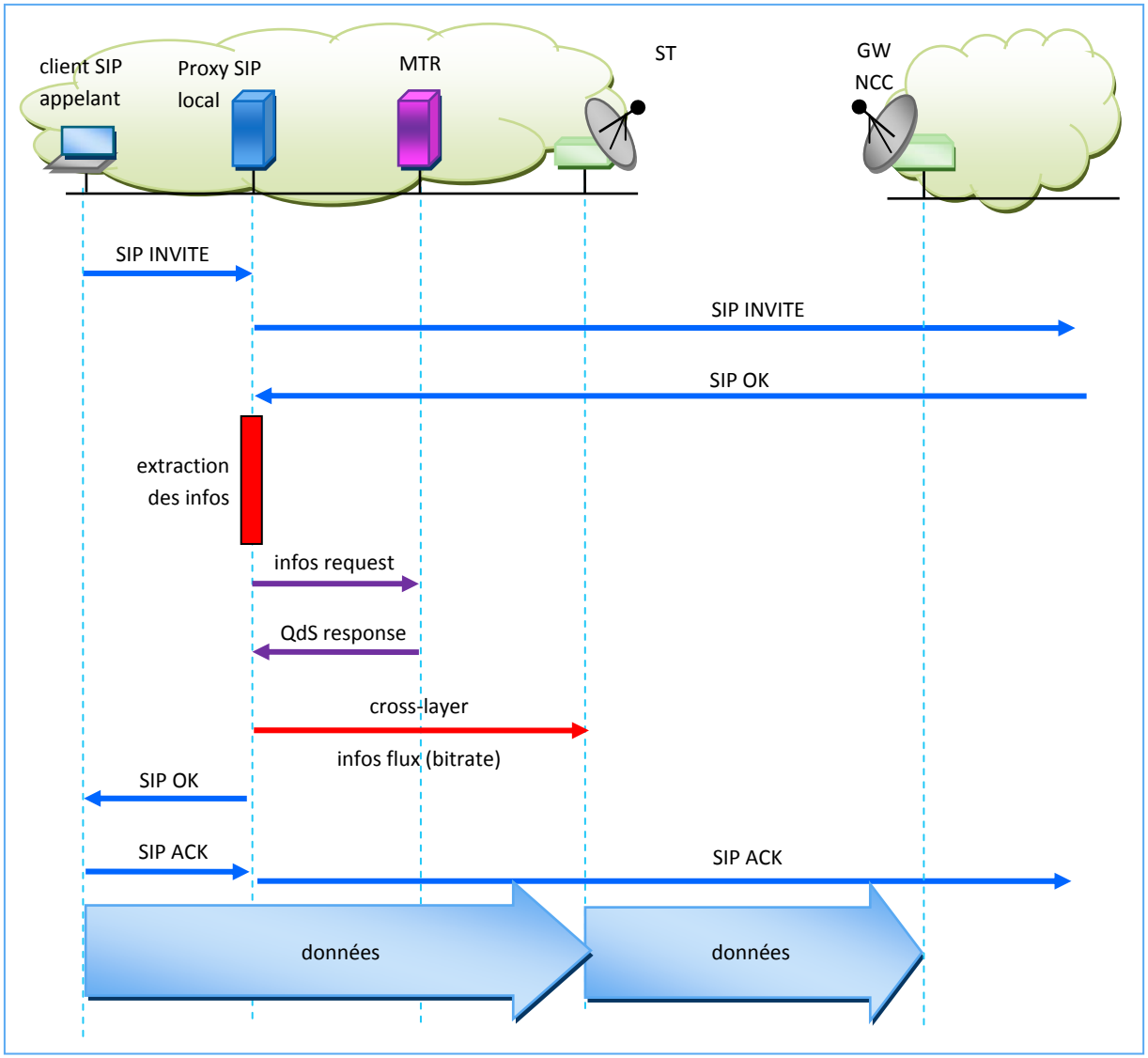


Figure 65 : scénario de transmission d'information cross-layer fourni au ST

6.4.3. Évaluation de performance du cas de référence

Nous proposons d'évaluer les performances du mécanisme d'allocation dynamique en fonction du facteur d'anticipation α. Cette évaluation s'articule autour de deux métriques :

- le délai de transmission des données multimédia. Ce délai est mesuré entre l'émission des données multimédia par l'émetteur rejoueur de trafic du côté du ST et leur réception par le récepteur du côté GW/NCC.
- le taux d'utilisation du lien retour du satellite. Ce taux correspond au rapport entre la quantité de ressources demandée par le client DAMA du ST à travers les requêtes dynamiques RBDC, et la quantité de ressources effectivement consommée pour la transmission des données sur la voie retour satellite. Il est mesuré directement au sein du ST. Afin que cette mesure soit objective, il est nécessaire que les requêtes de ressources émises soient entièrement satisfaites. Pour cela, le lien retour du satellite est dimensionné de manière à respecter cette hypothèse.

Nous proposons d'abord, pour différentes valeurs fixes du facteur d'anticipation α, d'observer le taux moyen d'utilisation du lien retour satellite en fonction du délai moyen de transmissions des données multimédia (c.f. Figure 66).

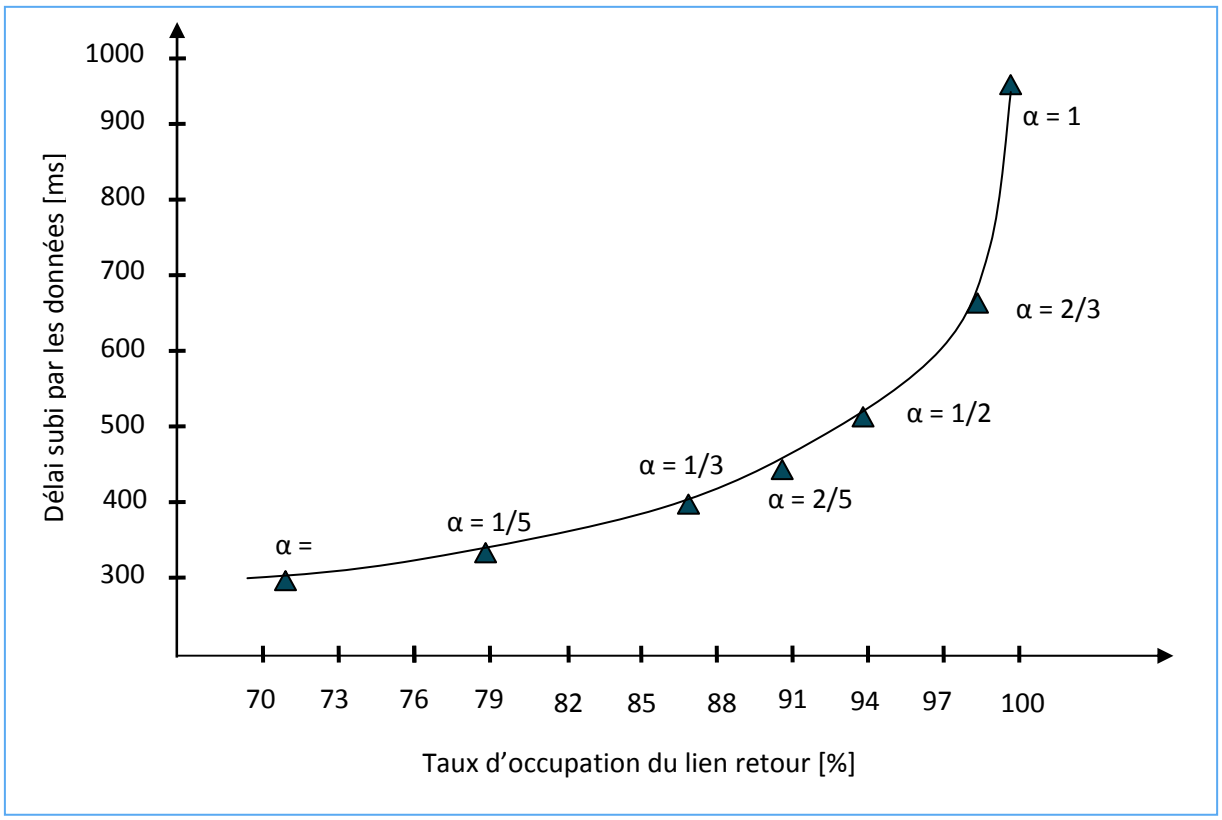


Figure 66 : Taux d'occupation en fonction du délai de bout en bout

Nous voyons, qu'avec un facteur d'anticipation α=0, nous sommes en mesure d'obtenir un délai moyen inférieur à 300ms (50ms attente dans tampons de transmission + 250ms propagation lien satellite). Ceci répond bien aux besoins des applications multimédia interactives. En effet, avec un tel facteur d'anticipation α=0, l'anticipation de ressources est à son maximum, par conséquent, le ST requiert et reçoit une quantité importante de ressources pour transmettre les données multimédia sur le lien retour. Ces données subissent donc très peu d'attente dans les tampons de transmission du ST. Cependant, les ressources demandées/reçues ne sont pas entièrement consommées. Les ressources non utilisées sont perdues. Cette valeur du facteur d'anticipation α=0 fait donc chuter le taux d'utilisation du lien retour à 72%.

À l'inverse, avec un facteur α=1, nous obtenons un taux d'utilisation du lien retour satellite à 100%. Une telle valeur du facteur d'anticipation représente une anticipation nulle, la quantité de ressources demandée est égale à la quantité de données arrivées dans les tampons de transmission du ST, à l'instant t. Par conséquent, la quantité de ressources demandée est entièrement consommée par les données en attente dans les tampons de transmission du ST. Cependant durant l'attente des ressources, d'autres données arrivent dans les tampons et sont donc obligées d'attendre le prochain cycle requête/allocation afin de pouvoir quitter les tampons de transmissions du ST. D'où un délai moyen de transmission de 1050ms (200ms d'attente dans les tampons de transmission + 500ms cycle requête/allocation + 250ms de propagation lien satellite).

Ce cas de référence confirme le dilemme entre :

- minimisation du délai moyen de transmission pour les applications multimédia de l'utilisateur ;
- et maximisation du taux d'occupation du lien retour satellite pour l'opérateur réseau.

6.4.4. Évaluation de performance de la contribution

Avec le même contexte d'expérimentation (flux audioconférence avec codec G.711 à 64kbit/s), nous proposons maintenant d'analyser les performances du mécanisme d'allocation de ressources avec un facteur d'anticipation α calculé dynamiquement grâce aux informations fournies par l'interaction cross-layer mise en place dans la contribution. Nous comparons les performances obtenues avec celles de trois valeurs statiques du cas de référence : α=0 ; α=0.5 ; α=1.

Pour cette analyse, les deux métriques, délai et taux d'utilisation, sont présentées séparément. Nous commençons avec l'évolution du délai de transmission des données sur la voie retour en fonction du temps (c.f. Figure 67).

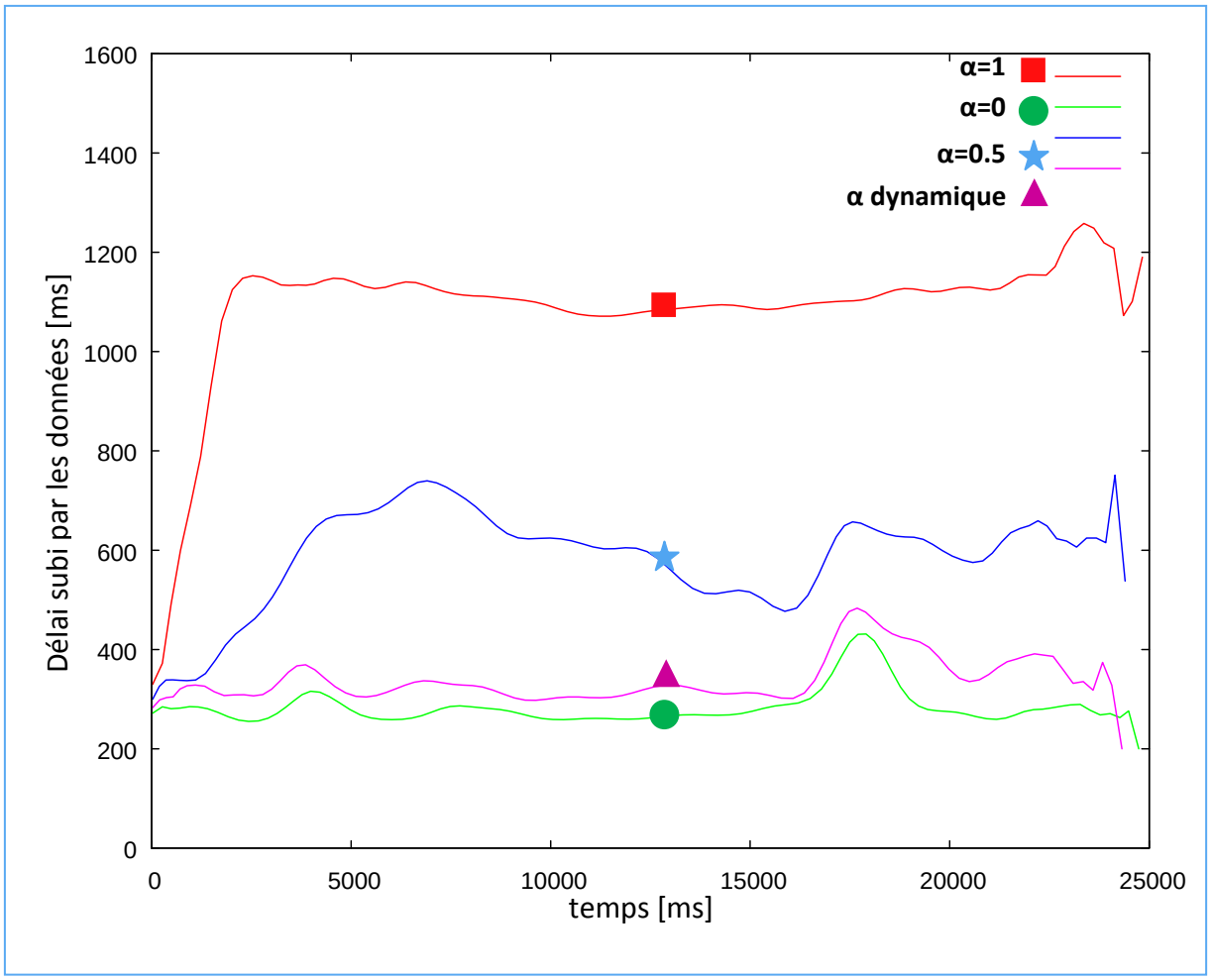


Figure 67 : délai de transmission avec alpha statique et dynamique

Sur la Figure 67, on remarque pour le début de l'expérimentation, quelque soit la valeur du facteur α (statique ou dynamique), que le délai est minimal (300ms). En effet, les ressources nécessaires pour la transmission des premières données multimédia sont gérées par la contribution précédente, à

l'aide de requêtes cross-layer de ressources anticipées au sein du NCC. Puis, selon la valeur de α, le délai évolue différemment.

Avec un facteur statique α=0 (courbe verte (rond)), le délai varie autour de 300ms, soit le délai minimal possible à obtenir. Ceci est bien sûr dû à une anticipation maximale de ressources pour écouler les tampons de transmission du ST.

À l'inverse, avec un facteur statique α=1 (courbe rouge (carré)), le délai augmente progressivement jusqu'à varier autour d'une seconde. Le manque d'anticipation entraîne l'augmentation du délai de transmission des données sur la voie retour.

Pour un facteur α=0.5 (courbe bleue (étoile)), le délai de transmission se dégrade également, pour se situer entre les deux cas précédents du facteur α statique. Il varie autour de 500ms.

Avec le facteur d'anticipation α calculé dynamiquement durant l'expérience (courbe violet (triangle)), nous voyons que le mécanisme s'adapte selon le trafic multimédia arrivant dans le ST afin de requérir suffisamment de ressources auprès du NCC pour limiter l'attente des données dans les tampons de transmission. Cela permet d'obtenir un délai moyen de transmission proche du celui du meilleur cas (α=0), qui varie autour de 320ms.

Nous présentons maintenant l'évolution du taux d'utilisation du lien retour en fonction du temps (c.f Figure 68).

Pour tous les cas de facteur d'anticipation α, statique et dynamique, le taux d'occupation du lien retour est à 100% au début de l'expérimentation. Cela montre bien que les ressources anticipées fournies par la contribution précédente ont été calculées précisément afin d'être toutes consommées lors de l'arrivée des premières données multimédia dans les tampons de transmission du ST.

En ce qui concerne la suite de l'expérience, dans le cas du facteur statique α=1 (courbe rouge (carré)), l'anticipation est minimale, par conséquent toutes les ressources sont consommées. On obtient un taux d'utilisation du lien retour de 100% tout le long de l'expérience.

Pour le facteur statique α=0 (courbe verte (rond)), le taux d'utilisation est au plus bas, soit 70%. L'anticipation est plus importante, ce qui entraîne des risques de sous utilisation des ressources demandées.

Avec le facteur α calculé dynamiquement (courbe violet (triangle)), la quantité de ressources à anticiper est calculée à chaque requête de ressources, soit toutes les 500ms, en fonction de l'état du trafic. Cette mise à jour régulière du facteur d'anticipation α permet de l'adapter en fonction du trafic arrivant afin de requérir une quantité suffisamment précise de ressources auprès du NCC. Au final, on obtient un taux d'utilisation proche du meilleur cas (α=1), qui varie autour de 97%.

Remarquons que durant l'expérience, à l'instant 17000ms, il y a une légère augmentation de délai de transmission pour l'ensemble des cas de figure. Ceci est causé par un mécanisme de détection d'activité vocale (VAD : Voice Activity Detection) utilisé par les codecs récents. Ce mécanisme distingue la voix du bruit de fond, et permet ainsi de diminuer le débit de transmission en absence de voix. Cependant, à la reprise de l'activité vocale, le débit augmente, et cette augmentation n'est pas

prédictible par le mécanisme d'anticipation de ressources. Cela entraine donc l'augmentation du délai de transmission mais également l'augmentation du taux d'occupation du lien retour satellite.

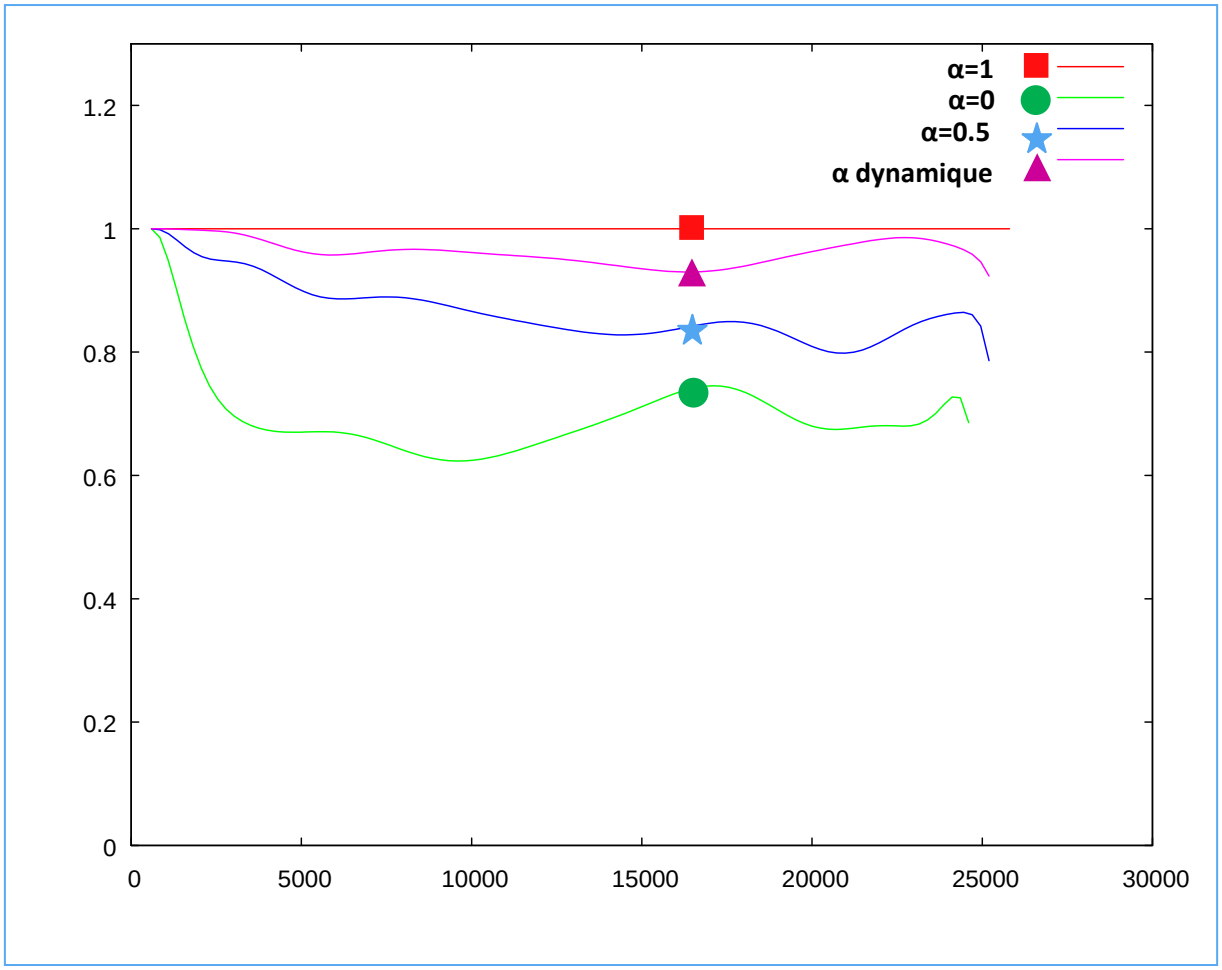


Figure 68 : taux d'occupation du lien retour avec alpha statique et dynamique

6.4.5. Conclusion

Ces résultats nous montrent que le calcul dynamique du facteur d'anticipation α durant la communication multimédia permet de fournir un compromis entre délai de transmission et taux d'utilisation du lien retour satellite.

Dans le cas du taux d'utilisation des ressources de la voie retour satellite, le calcul dynamique du facteur d'anticipation α permet de s'approcher des valeurs du meilleur cas, c.à.d. α=1, en ajustant la quantité de ressources demandée sans en gaspiller.

Alors que dans le cas du délai de transmission des données multimédia sur la voie retour, le calcul dynamique de α se rapproche des valeurs du meilleur cas, c.à.d. α=0, en requérant une quantité suffisante de ressources pour anticiper les futurs besoins d'émission, évitant ainsi les attentes dans les tampons de transmission du ST.

Ce calcul dynamique du facteur d'anticipation α est finalement possible grâce à l'interaction cross-layer mise en place dans la contribution, entre la couche session (proxy SIP) et la couche liaison (client DAMA).

6.5. Contrôle d'admission et de débit pour session multimédia

Dans ces expérimentations, nous voulons évaluer l'impact de la contribution concernant le contrôle de débit pour les sessions multimédia dans le réseau d'accès satellite DVB-S2/RCS. Cette contribution permet d'adapter le trafic multimédia en fonction de la charge du réseau satellite. Elle est proposée sous deux variantes, selon l'environnement réseau dans lequel elle est utilisée :

- En environnement contrôlé, où la charge réseau, les ressources disponibles et les flux admis sont connus. Dans ce cas, notre contribution se base sur les informations de session, tel que dans SIP, afin de déclencher le changement de codec des sessions multimédia.
- En environnement non contrôlé, où la charge réseau et sa nature, ainsi que les ressources disponibles sont très variables (ex. classe Best Effort). Dans ce cas, nous utilisons une communication cross-layer, de la couche liaison (serveur DAMA) vers le plan de contrôle (mécanisme de contrôle d'admission).

Nous présentons ici l'évaluation de la variante concernant l'environnement contrôlé.

6.5.1. Scénario d'expérimentation

Il est important de noter que les performances de cette approche dépendent directement du nombre de codecs disponibles pour la communication multimédia concernée. En effet, avec un ensemble de codecs fournissant un large panel de débits utilisables durant la communication, nous sommes en mesure d'effectuer des adaptations précises de débits aux conditions de charge du réseau.

Initialement, l'application client SIP *minisip* utilisée pour nos expérimentations, met à disposition uniquement le codec G.711, possédant un débit d'encodage moyen de 64kbit/s. Nous proposons d'intégrer un deuxième codec, G.721 avec un débit d'encodage de 32kbit/s. Ces deux codecs audio proposés par l'ITU-T, utilisent la même fréquence d'échantillonnage de 8KHz. Cette configuration nous offre ainsi la possibilité de faire varier la charge de transmission d'une communication multimédia entre deux valeurs de débit : 64kbit/s et 32kbit/s. Nous intégrons également le mécanisme permettant de gérer la réception de message SIP-ReINVITE lors du changement de codecs.

En ce qui concerne le proxy SIP *partysip*, il est installé du côté GW/NCC. Son code source est également modifié afin d'intégrer le mécanisme permettant de déclencher le changement de codecs auprès des clients SIP *minisip*.

Nous intégrons alors au sein du NCC un mécanisme de contrôle d'admission avec les règles précédemment évoquées dans la contribution.

Pour le scénario d'expérimentation, plusieurs sessions multimédia d'audioconférence sont lancées à différents instants dans le réseau satellite. Chacune de ces sessions dispose des deux codecs (G.711 et G.721) pour leur communication multimédia. Au niveau ST, les flux multimédia à transmettre sur la voie retour sont dirigés vers une classe de trafic dédiée. Nous limitons la capacité réseau de cette classe à 200kbit/s.

Pour l'expérimentation, nous lançons 5 sessions d'audioconférence à travers le réseau satellite :

- à l'instant t=0sec (début de l'expérimentation), 3 flux audioconférence démarrent ;
- à l'instant t=30sec, nous introduisons un 4e flux audioconférence dans le réseau satellite ;
- et à l'instant t=60sec, nous introduisons un 5e et dernier flux audioconférence.

6.5.2. Mesures et résultats

La Figure 69 montre l'impact du mécanisme de changement sur l'évolution des flux multimédia. La Figure est divisée en 6 parties, 5 représentants un flux multimédia, et la sixième montre la charge totale de la classe de trafic.

À l'instant t=0sec, les 3 premiers flux d'audioconférence souhaitent démarrer. À ce stade, l'ensemble des ressources réseau (200 kbit/s) sont disponibles. Par conséquent, nous voyons que le contrôle d'admission accepte ces 3 flux avec utilisation du codec G.711 à 64 kbit/s. Jusqu'à l'instant t=30sec, la charge totale de la classe de service dédiée s'élève à 192 kbit/s.

À l'instant t=30sec, le 4e flux audioconférence souhaite démarrer. Or, l'admission d'un quatrième flux avec le codec G.711 à 64kbit/s ou encore G.721 à 32kbit/s entraîne le dépassement de la capacité réseau de la classe de service dédiée. En effet le système ne peut assurer un support de QdS à ce quatrième flux sans dégrader le service fourni aux 3 flux déjà admis.

Afin de tenter d'admettre le nouveau flux, le contrôle d'admission recalcule la charge actuelle de la classe de trafic. Il aboutit à un changement de codec de l'un des 3 flux déjà admis, et l'admission du 4e flux avec le codec G.721 à 32kbit/s. Sachant que seul l'un des 3 flux admis nécessite d'être renégocié, le contrôle d'admission choisit arbitrairement le premier flux. Il fournit cette information au proxy SIP qui lui déclenche le changement de codec (G.711 vers G.721) pour la 1ère communication qui voit son débit chuter à 32 kbit/s.

Puis le proxy SIP continue l'établissement de la 4e session audioconférence avec pour codec autorisé durant la communication, le G.721.

De t=30sec à t=60sec, la charge totale de la classe est de 32+32+64+64=192kbit/s.

À t=60sec, la 5e session multimédia souhaite démarrer dans le réseau. La démarche précédente est également appliquée sur l'un des 2 flux déjà admis utilisant le codec G.711. Le flux représenté par la courbe est choisi pour le changement de codec. Cette session voit son codec en cours changé (de G.711 vers G.721), par conséquent son débit est également diminué (de 64 kbit/s à 32kbit/s).

Au final, nous avons 5 flux audioconférence admis dans la classe alors qu'initialement, 3 flux multimédia remplissaient la charge totale de la classe. Grâce à notre mécanisme, nous avons 4 flux multimédia qui utilisent le codec G721 à 32kbit/s et le 5e flux qui utilise le codec G711 à 64kbit/s.

La dernière courbe montre l'évolution de la charge totale de la classe de trafic dédiée pour l'expérimentation. Comme on peut le voir, le mécanisme proposé permet une évolution relativement stable de la charge réseau malgré les multiples admissions de flux multimédia durant l'expérience. Il offre ainsi une utilisation optimale des ressources réseau du lien retour satellite pour la transmission des données multimédia. En ce qui concerne le délai de transmission, le délai subi par les flux multimédia avec l'intégration du mécanisme de changement de codec est le même présenté dans les résultats de la contribution précédente concernant les mécanismes d'anticipation de ressources pour le lien retour satellite.

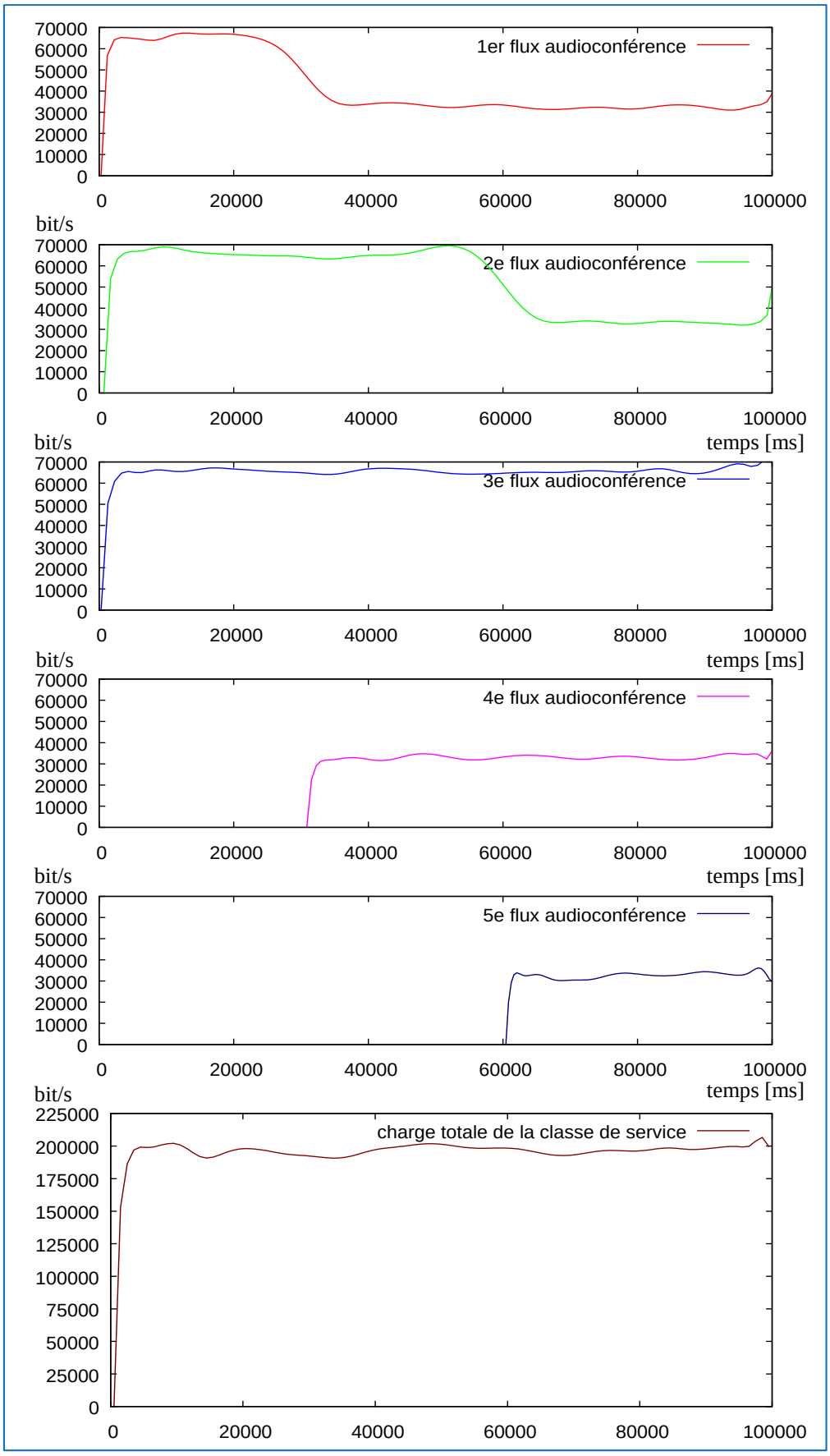


Figure 69 : évolution des flux multimédia avec contrôle de débit

6.5.3. Conclusion

Les expérimentations menées nous ont montré que le mécanisme de changement de codec permet d'adapter le débit de transmission des sessions multimédia en fonction des ressources disponibles. Ceci offre ainsi une utilisation optimale de lien retour satellite.

Deux points méritent d'être précisés :

- Lors de conditions réseau très variables, où la quantité de ressources disponibles change fréquemment, les sessions multimédia admises ont tendance à changer aussi fréquemment de codec pour s'adapter aux conditions réseau. Or un changement de codec trop fréquent risque de perturber la communication multimédia. Il est donc important de limiter la fréquence de changement de codec pour les sessions multimédia.
- Lors du changement de codec, le choix de la session multimédia parmi celles qui sont en cours, à renégocier, est effectué arbitrairement par l'algorithme proposé. Ce dernier choisit la plus ancienne session enregistrée. Il serait préférable d'effectuer ce choix selon des critères plus pertinents, pouvant être négociés en amont, durant la phase de mis en place du contrat de qualité de service avec le client (ex. SLA).

6.6. Architecture de communication Cross-Layer

Dans le chapitre 4, nous avons défini une architecture de communication cross-layer pour les réseaux sans fil, l'objectif de cette architecture étant de mettre à disposition des données présentes à différents niveaux de la pile protocolaire. L'ensemble des fonctionnalités fournies par cette architecture sont testées. Cependant, dans cette partie, nous allons présenter les tests fonctionnels de deux des fonctionnalités : l'enregistrement et la mise à jour de données (NF).

Concernant, les tests de performance, il s'agit plus précisément de la capacité des communications cross-layer à optimiser le système selon la situation et les besoins. Par conséquent, nous considérons que les résultats précédemment présentés sur les communications cross-layer représentent ces tests de performance.

6.6.1. Scénario d'expérimentation

La généricité de l'architecture proposée lui permet de fonctionner avec n'importe quel type de technologie réseau du nœud de communication (Ethernet, Wifi, Wimax, Satellite, etc.). Dans un souci d'intégration, nous choisissons de l'appliquer au réseau Wifi 802.11 pour les expérimentations.

Nous mettons en place une plate-forme wifi 802.11 : un ordinateur portable possédant une interface wifi est connecté (sans fil) à un point d'accès wifi.

À partir de cette plate-forme, nous installons notre architecture cross-layer au sein de l'ordinateur portable et proposons de mettre à disposition plusieurs données :

- au niveau socket : un compteur d'utilisation des sockets netlink ;
- au niveau transport : la longueur des buffers et les flags utilisés dans le protocole TCP ;
- au niveau physique : la qualité du signal, le SNR, et la proportion de bruit sur le driver de carte Wifi.

La couche liaison est celle de l'interface réseau sans fil 802.11, dont le chipset est Intel Pro WLAN 3945. Ce dernier fonctionne sous le système Linux 2.15.20 et utilise le pilote réseau iwlwifi-1.2.25.

Nous modifions le code source de ce pilote réseau afin de mettre à disposition les trois paramètres concernant le signal sans fil.

Afin vérifier le bon fonctionnement de l'architecture, nous proposons de visualiser des messages d'état, introduits dans les 3 parties de l'architecture :

Pour les parties I_noyau et CL_entity, étant au niveau utilisateur, leurs messages sont affichés directement dans leur console de lancement respective. Pour la troisième partie présente dans l'espace noyau, nous utilisons l'interface du protocole « syslog » [97] communément déployé sur les systèmes Unix/Linux. Ce protocole client/serveur est initialement conçu pour la gestion de systèmes d'ordinateur et le contrôle de sécurité. Il permet de transmettre des messages d'alerte et/ou d'information de bas niveau, au sein d'un même ordinateur ou encore à travers un réseau IP. Notre entité dans l'espace noyau se comporte en client et envoie ses messages au serveur syslog qui lui, les affiche à l'écran.

6.6.2. Validation fonctionnelle

6.6.2.1. Étape d'enregistrement des NF (données)

Lorsque l'ordinateur portable démarre, les différentes données NF concernées présentes au niveau noyau émettent leurs demandes d'enregistrement au I_noyau. La Figure 70 montre deux des trois demandes d'enregistrement : le NF NOISE_EMILIE pour le paramètre de bruit du canal et le NF SIGNAL_EMILIE pour la qualité du signal. Le champ suivant est la valeur du paramètre à l'enregistrement.

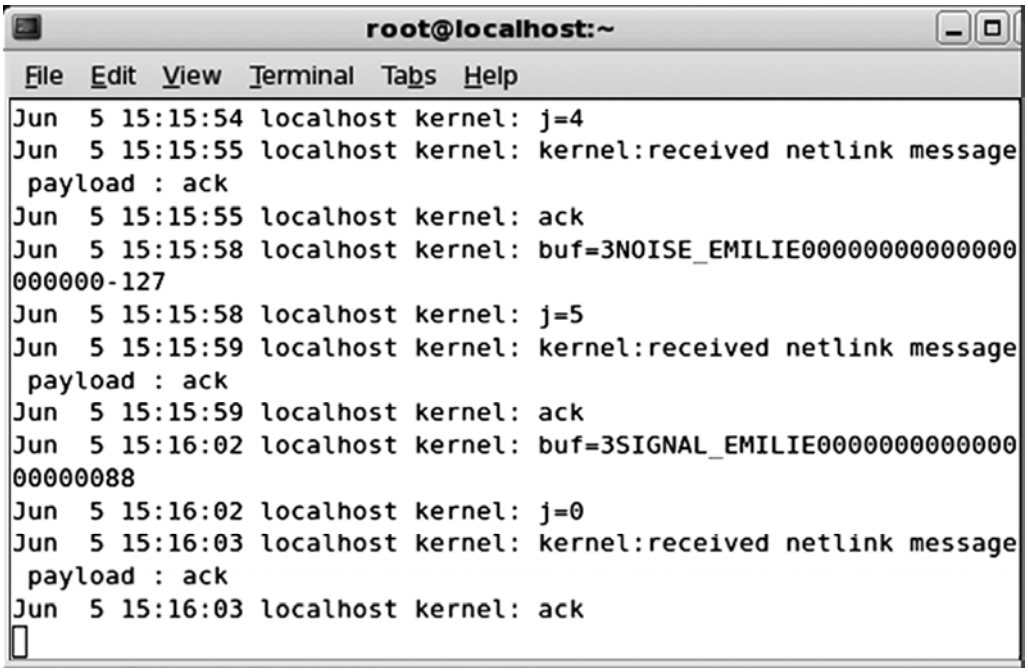


Figure 70 : : messages d'enregistrement des données au sein du noyau

Au sein de l'entité I_noyau (c.f. Figure 71), la fonction « process_msg_kernel » s'active afin de recevoir et traiter le message d'enregistrement provenant du noyau. Sur la Figure 71, nous voyons le cas du NF qualité du signal. Cette demande est ensuite encapsulée puis envoyée au CL_entity. Pour l'encapsulation : on retrouve le type du message, 3, qui correspond bien à une demande d'enregistrement de NF ; l'id, 105, l'identifiant de la communication ; le NFname, SIGNAL_EMILIE ; le champ data, 88, étant la valeur du signal lors de l'enregistrement (88%) ; et le hostname, hecate, qui est le nom de l'ordinateur portable.

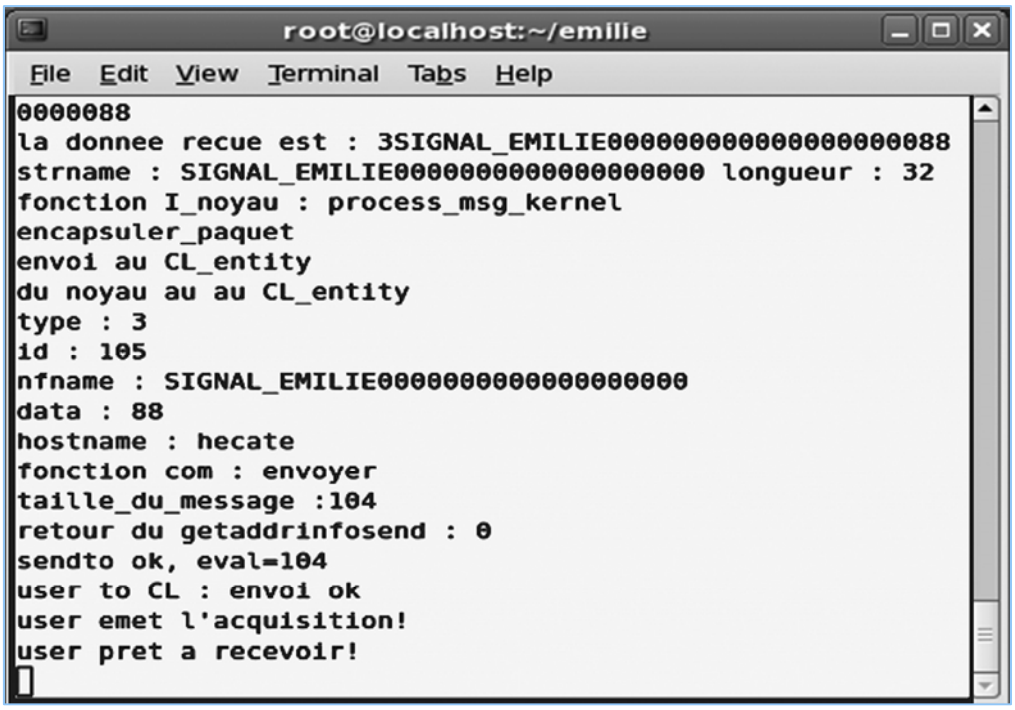


Figure 71 : messages d'enregistrement des données au sein du I_noyau

Le CL_entity quant à lui, lance la méthode « ProcessMsg » afin de recevoir et centraliser l'ensemble des demandes d'enregistrement des NF provenant du I_noyau (c.f. Figure 72). Au niveau de l'interprétation de l'affichage du CL_entity, on retrouve les champs précédemment décrits pour l'entité I_noyau. Ici, les 6 données NF sont enregistrées auprès du CL_entity.

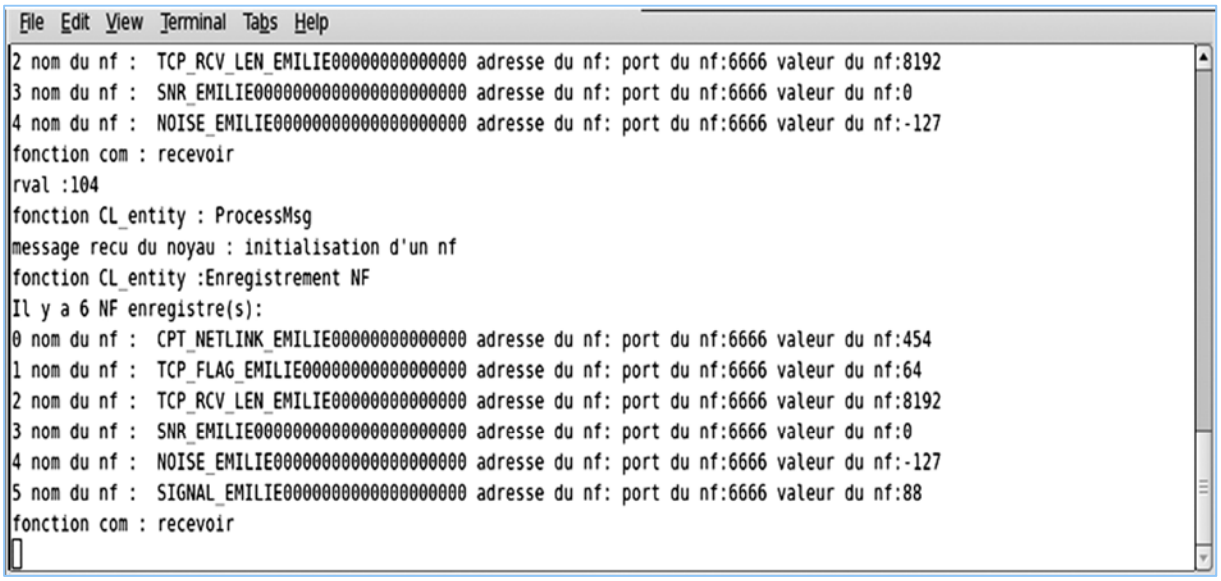


Figure 72 : messages d'enregistrement des données au sein du CL_entity

On remarquera que le bruit est initialisé à -127dBm à l'enregistrement. Cela veut dire que l'interface réseau wifi de l'ordinateur portable n'a pas pu l'estimer et que la qualité du signal n'est pas extraite du SNR (signal (dBm)-noise (dBm)).

6.6.2.2. Étape de mise à jour des NF

Afin de vérifier la mise à jour correcte des NF enregistrés, nous choisissons de faire évoluer la qualité du signal perçue par l'interface wifi. Pour cela, nous éloignons (géographiquement) l'ordinateur portable (donc l'interface réseau) du point d'accès wifi.

Comme le montre la Figure 73, au fur et à mesure de l'éloignement, le noyau envoie la mise à jour de la qualité du signal au CL_entity (par l'intermédiaire du I_noyau). Nous voyons ainsi décroitre la valeur du NF NF de 90%, 62% jusqu'à 37%.

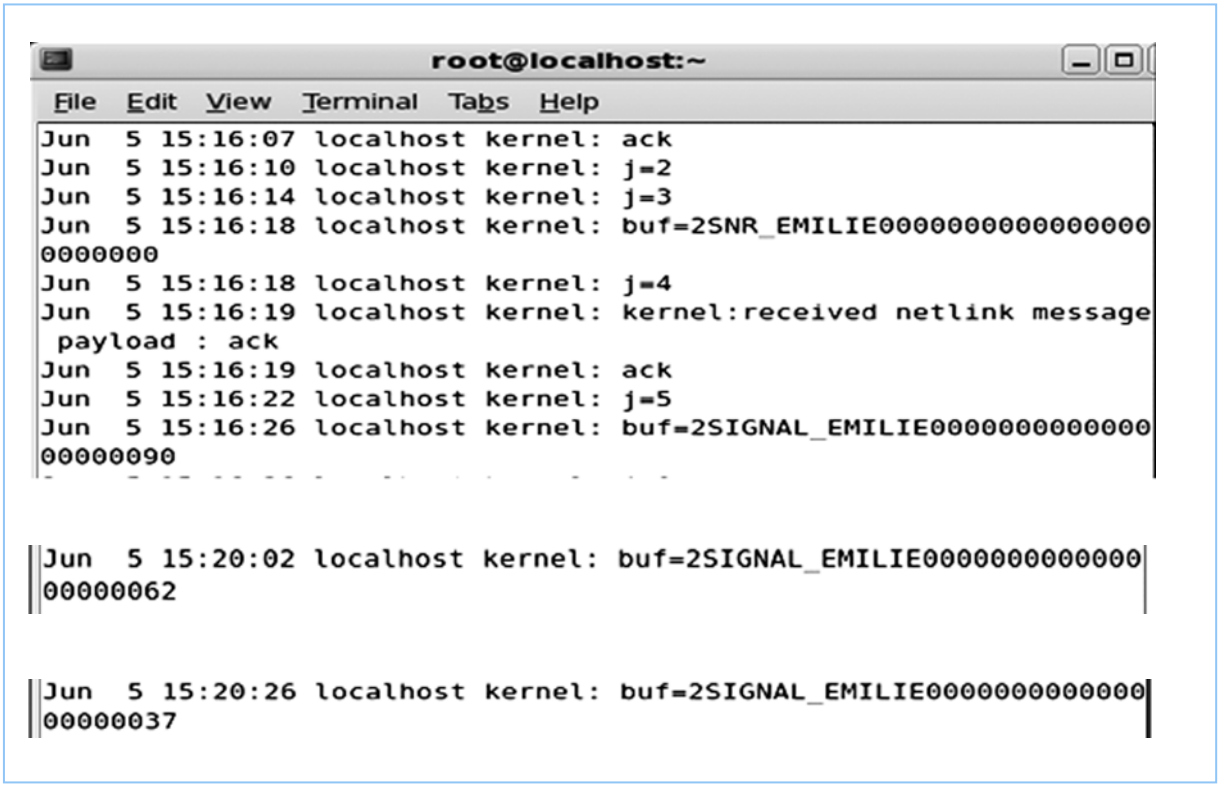


Figure 73 : messages de mise à jour de la qualité du signal au sein du noyau

6.6.3. Conclusion

Ces résultats ont permis de justifier le bon fonctionnement de l'architecture cross-layer proposée dans ces travaux de thèse. Cette architecture permet d'enregistrer des paramètres puis de les mettre à disposition en recevant régulièrement les mises à jour des valeurs. À partir de ces mises à jour reçues, la couche applicative est en mesure d'effectuer des opérations appropriées, telles que la recherche de nouvelles cellules wifi, ou encore la fermeture anticipée de connexions au niveau application, ou encore la gestion des connexions au niveau transport, tel que dans TCP.

6.7. Architecture orientée Web Services : caractérisation des besoins QdS

Dans cette section, nous allons présenter l'évaluation de la contribution concernant la base de données MTR (Media Type Repository). Cette architecture a pour objectif la publication, la découverte et l'invocation de services par le client auprès du fournisseur du service. Dans nos travaux de thèse, elle est instanciée par la base de données MTR (Media Type Repository). Elle est essentiellement utilisée pour fournir la spécification des besoins en QdS des applications multimédia aux proxys SIP. Ces derniers utilisent ces informations pour quantifier les ressources nécessaires pour la réservation auprès du réseau satellite. Nous proposons ici d'évaluer les performances de notre architecture orientée web services dans un environnement satellite DVB-S2/RCs.

6.7.1. Scénarii d'expérimentation

L'architecture MTR peut être déployée essentiellement de deux manières différentes dans le réseau satellite :

- De manière centralisée (c.f. Figure 74) : le serveur MTR est placé à l'entrée du réseau, du côté GW/NCC. Ainsi, une seule base de données MTR est nécessaire pour offrir une disponibilité totale à tous les clients du réseau satellite. De plus cette topologie permet une maintenance et mise à jour aisées de la base de données. Cependant, comme on peut s'y attendre, l'accès au serveur MTR subira le temps de propagation du lien satellitaire.

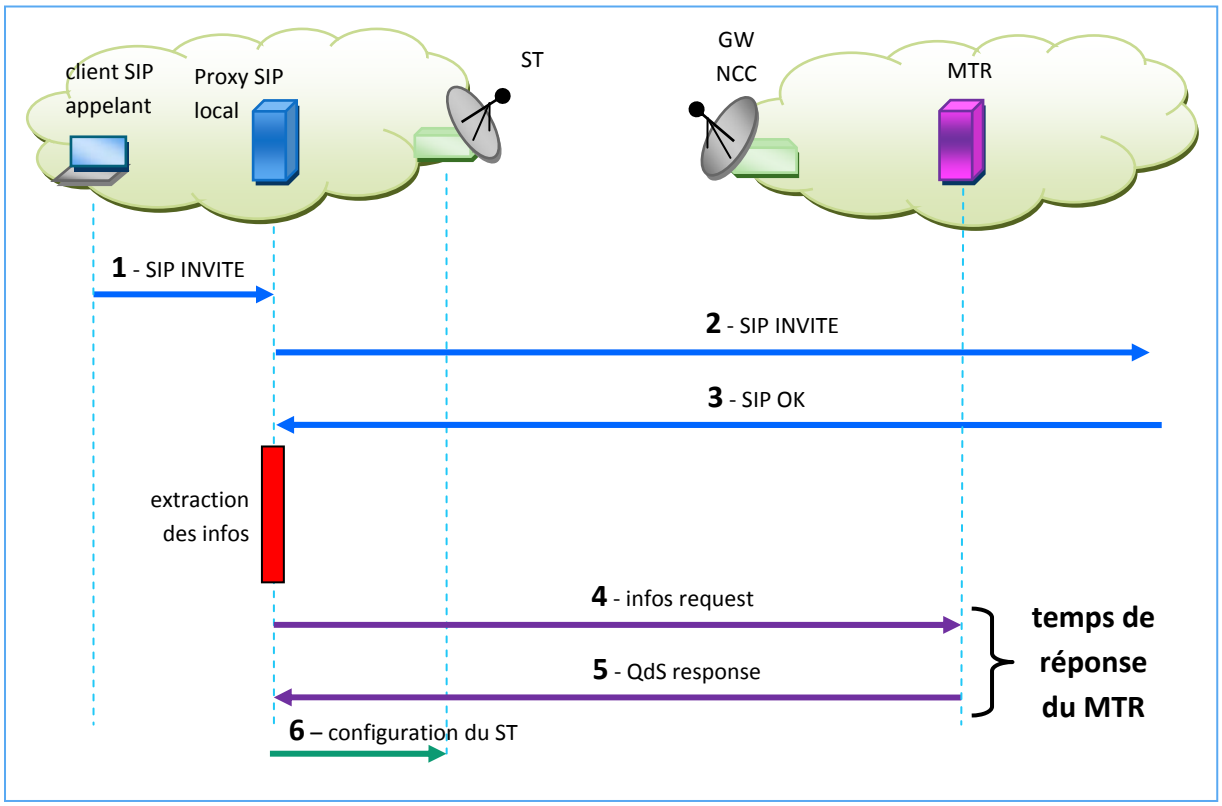


Figure 74 : scénario d'expérimentation avec serveur MTR côté GW/NCC (centralisé)

- De manière distribuée (c.f. Figure 75) : le serveur MTR est placé dans chaque réseau utilisateur, derrière le ST. Cette topologie permet un temps de réponse faible et une autonomie vis-à-vis du reste du réseau satellite.

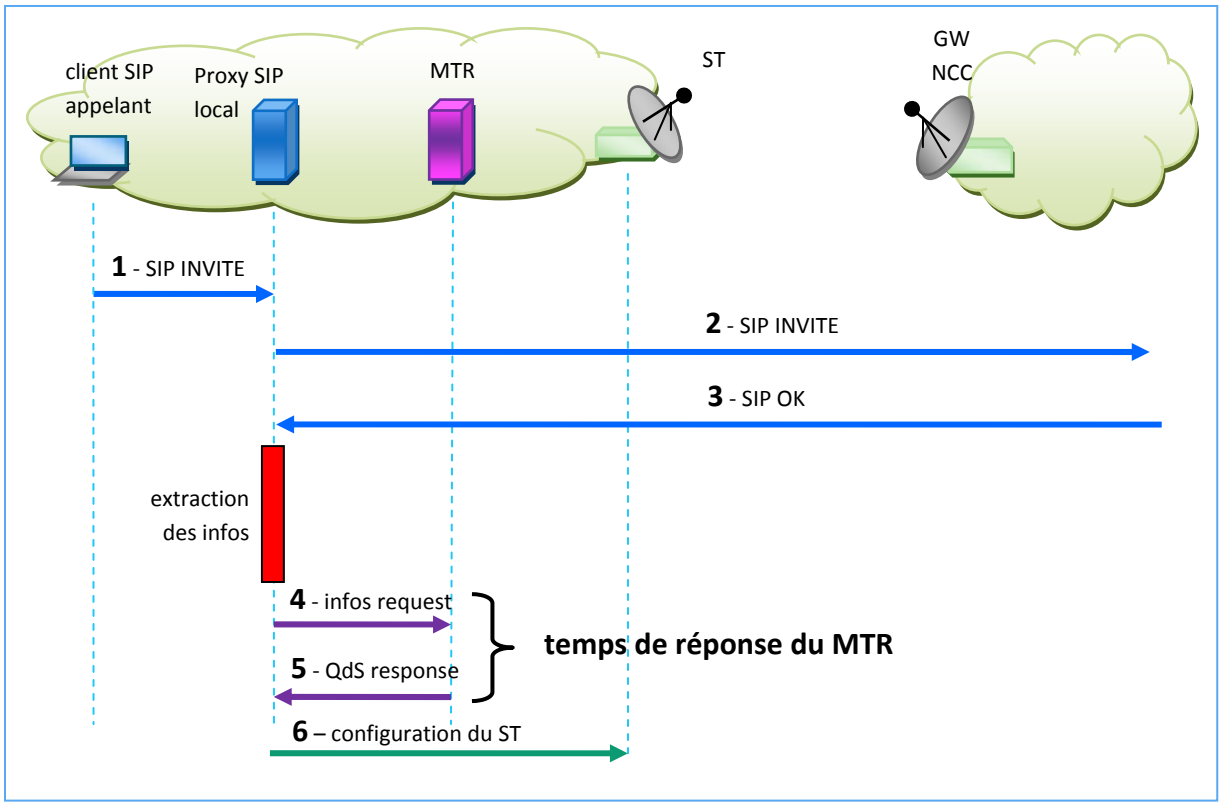


Figure 75 : scénario d'expérimentation avec serveur MTR côté ST (distribué)

Nous proposons ici d'évaluer les performances de l'architecture MTR en termes de temps de réponse pour les deux topologies précédemment présentées.

Dans un scénario d'établissement de session multimédia SIP, nous utilisons le MTR en tant que base de données fournissant les spécifications en QdS des applications multimédia durant l'établissement d'une session SIP.

Le proxy SIP extrait du message SIP OK les caractéristiques de la session à venir : le type de média (ex. audio, vidéo) et le format du média (ex. nom du codec : G.711, G.721, GSM). Puis il est chargé d'interroger le serveur MTR, afin que ce dernier lui fournisse les spécifications en QdS pour la communication. Pour cela on intègre au proxy SIP une interface de communication avec le serveur MTR pour effectuer ces requêtes et recevoir les réponses.

6.7.2. Mesures et résultats

Nous mesurons le temps de réponse (temps d'aller/retour) pour l'interrogation du serveur MTR dans les deux scénarii.

La Table 4 montre les résultats de mesure du temps de réponse dans les 2 scénarii.

Table 4 : temps de réponse dans les scénarii d'expériemntation

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

Comme on peut le voir, dans le scénario centralisé, nous obtenons un temps de réponse moyen de 1229ms. Ce temps de réponse correspond :

- Temps de mise en place de la connexion au serveur MTR. En effet, le protocole utilisé pour communiquer les informations est basé sur le protocole HTTP, qui lui-même utilise le protocole de transport TCP. Par conséquent, la poignée de main à 3 voies de TCP (three-way handshake) provoque 3 traversées du lien satellite, soit 250 ms x 3 = 750 ms.
- Temps d'aller-retour des données, qui entraine deux traversées du lien satellite, soit 250 ms x 2 = 500 ms.

Nous obtenons bien au final 1250 ms théorique de temps de réponse dans la topologie centralisée.

Pour le scénario distribué, le temps de réponse moyen obtenu est de 17.75 ms, qui est largement inférieur à celui obtenu dans le scénario centralisé. En effet, les échanges s'effectuent sur un réseau filaire de type Ethernet par conséquent le temps de propagation est négligeable.

Le temps de réponse élevé obtenu pour le scénario centralisé aura une influence plus ou moins importance selon le contexte d'utilisation du serveur MTR. En effet, si nous reprenons la contribution 3.4, cette latence d'accès au serveur MTR n'a pas d'incidence néfaste sur le service fourni à l'utilisateur final, mais cette latence prolonge la tonalité d'appel pour les clients SIP durant l'établissement de la session multimédia.

6.7.3. Conclusion

Nous avons présenté ici l'évaluation de performances de l'architecture orientée web services, en termes de temps d'accès au serveur MTR. Selon la topologie utilisée (centralisée ou distribuée), l'accès au serveur varie entre une dizaine de millisecondes et 1 seconde. Cette latence élevée est essentiellement due à l'utilisation du protocole de transport TCP pour fiabiliser les échanges entre client serveur MTR.

En cas de besoin de réactivité du système, une topologie distribuée sera adoptée. Alors qu'en cas de besoin de maintenance et mise à jour de la base de donnée du serveur MTR, une topologie centralisée sera plus appropriée.

Le problème de la latence d'accès due à l'utilisation du protocole TCP pour l'échange d'informations pourrait être résolu par l'utilisation du protocole de transport UDP. [98] propose l'utilisation de TCP et UDP comme protocoles de transport du protocole HTTP. La spécification [99] préconise l'utilisation de UDP pour transporter les données SOAP.

7. Conclusion générale et Perspectives

7.1. Bilan

Les nouvelles applications multimédia fournissent aujourd'hui des environnements de communication et collaboration complets à la fois pour l'utilisateur grand public, comme pour les entreprises. La notion de « multimédia » permet ainsi, au sein d'une même session, l'utilisation de la voix (ex. VoIP), la vidéo (ex. audioconférence ou Vidéo à la Demande), le partage d'application et bien d'autres médias.

Les récentes technologies réseau sans fil permettent aux utilisateurs les plus itinérants de bénéficier du service minimum d'accès Internet (ex. navigation web, transfert de fichiers). Il s'agit en l'occurrence des technologies réseau wifi (802.11e), réseau wimax (802.16) et réseau satellite (DVB-S2/RCS).

Cependant les déploiements de nouveaux services multimédia sur ces technologies réseau sans fil sont confrontés à de nombreux problèmes. Ces services nécessitent un niveau de qualité minimum afin d'être satisfaisants.

Nous avons dans un premier temps défini ce niveau de qualité de services pour différents types d'applications, en termes de métriques de performance : délai de bout en bout, gigue et taux de perte maximaux, débit de transmission minimal. Cette classification a montré que les applications multimédia interactives nécessitent un support de qualité de service adapté pour satisfaire les contraintes temporelles.

À partir de ces besoins en QdS, nous avons effectué un état de l'art des mécanismes et architectures QdS actuellement disponibles pour le réseau internet. Nous constatons que diverses fonctionnalités QdS sont disponibles à chaque niveau de la pile protocolaire du système de communication Internet.

Cependant la conception du réseau Internet est basée sur le paradigme du système de communication en couche hiérarchique, à savoir qu'une couche est optimisée pour fournir un service spécifique uniquement à la couche supérieure et ainsi de couche en couche.

Nous avons montré que le déploiement d'une telle pile sur les réseaux à technologies sans fil, particulièrement sur réseaux d'accès par satellite, afin d'assurer un service de qualité aux applications multimédia interactives, pose divers problèmes. Les propriétés des différentes couches de la pile protocolaire possèdent des interdépendances substantielles. Il est donc nécessaire d'établir, à travers les différentes couches, des communications fournissant des informations de contrôle pertinentes, ceci afin de résoudre le conflit entre optimisation de l'utilisation des ressources réseau et support de QdS pour les utilisateurs. Nous avons alors présenté un état de l'art du concept « Cross-Layer » et des architectures existantes. Une communication cross-layer nécessite la création de nouvelles interfaces dans l'implémentation des couches concernées. Ces communications cross-layer peuvent être entre couches adjacentes ou non-adjacentes.

À partir de cet état de l'art, nous avons proposé et implémenté un certain nombre de communications cross-layer afin de répondre à plusieurs problématiques liées au déploiement d'applications multimédia sur les réseaux d'accès par satellite. Ces problématiques concernent plus précisément l'utilisation de l'allocation dynamique pour le transfert des flux multimédia.

La première contribution établit une communication cross-layer entre le plan de session (proxy SIP) et la couche liaison (MAC) au sein du NCC (élément gérant les ressources du réseau). À l'établissement d'une session multimédia à travers le réseau satellite, le proxy SIP, à l'aide d'une base de données (MTR), identifie les caractéristiques QdS (débit de transmission) du futur flux multimédia, puis effectue une requête dynamique de ressources anticipée pour la transmission des premiers paquets du flux multimédia sur la voie retour. Ainsi lorsque le flux multimédia démarre, les premiers paquets multimédia arrivent dans les tampons de transmission du ST. Ce dernier est en mesure de transmettre ces paquets sur la voie retour satellite sans les soumettre au délai du cycle requête/allocation.

La seconde contribution concerne toujours l'utilisation de l'allocation dynamique pour le transfert des données multimédia, mais cette fois-ci, durant la transmission du flux multimédia. Un algorithme d'anticipation de ressources est utilisé par le ST pour effectuer les requêtes de ressources. Afin d'affiner le calcul des ressources requises, nous avons proposé de créer une communication cross-layer au sein du ST, du plan de session (proxy SIP) vers la couche liaison (MAC) du ST. À partir des messages d'établissement de session, le proxy SIP identifie les caractéristiques QdS du flux à venir et fournit le débit de transmission au client DAMA de la couche liaison du ST. Ce dernier prend alors en compte cette information pour préciser le calcul de requêtes de ressources anticipées. Cela permet ainsi d'améliorer les performances du service rendu aux flux multimédia en limitant le délai d'attente des paquets dans la file d'émission du ST, et optimise l'utilisation des ressources du lien retour satellite.

La troisième contribution propose un ensemble de mécanismes et algorithmes QdS afin d'optimiser l'admission et le contrôle des flux multimédia dans le réseau satellite. En se basant sur les informations de session, nous proposons de modifier les caractéristiques des sessions multimédia à venir et en cours, pour optimiser l'utilisation des ressources de la voie retour, et ainsi être en mesure d'accepter davantage de connexions multimédia dans le réseau satellite. Pour cela, nous utilisons une méthode de changement de codec, permettant de faire varier le débit de transmission des communications en cours, et par conséquent la charge actuelle du réseau. Grâce à une méthode de spoofing, le changement de codecs est initié par le proxy SIP pour les clients multimédia, qui eux sont inconscients de la QdS sous-jacente. Dans le cas où la quantité de ressources disponibles varie de manière imprévisible, nous établissons une communication cross-layer de la couche liaison (MAC) du ST vers le plan de session (proxy SIP), qui permet de déclencher le changement de codecs auprès des clients.

Afin d'intégrer toutes ces communications cross-layer proposées, nous avons conçu et développé une architecture à communication cross-layer. Cette architecture dédiée permet de conserver le fonctionnement normal de la pile protocolaire Internet, car les communications cross-layer sont gérées par une entité intermédiaire. Cette dernière permet de garder une compatibilité avec l'architecture classique en couche. Cela nous permet donc de maintenir tous les avantages inhérents à une architecture modulaire en couches isolées, tels que la robustesse ou la facilité d'évolutivité. De plus, en centralisant les informations cross-layer à partager, les différentes couches sont bien moins sollicitées.

La dernière contribution est une base de données MTR (Media Type Repository) orientée web services. Elle permet à la fois (i) au fournisseur de services de définir ses services, à travers la

description des interfaces fonctionnelles des services, (ii) de publier ces services dans un annuaire de type registre UDDI, facilitant la recherche et la localisation de services, (iii) et les clients de découvrir (iiii) et invoquer ces services auprès du fournisseur. Dans le cas des communications cross-layer précédemment présentées, elle permet de répondre aux requêtes de besoin en QdS des applications multimédia. Cette architecture met ainsi à disposition l'ensemble des services disponibles dans le réseau satellite.

7.2. Perspectives

L'état d'avancement de ces travaux de thèse laisse ouvert plusieurs perspectives.

Les communications cross-layer proposées tendent à répondre aux besoins en QdS des applications multimédia interactives de type audioconférence et visioconférence, sur les réseaux satellite. Pour les applications plus complexes contenant plusieurs médias, à la fois audio, vidéo, messagerie instantanée, tableau blanc, elles nécessitent une synchronisation entre ces différents médias qui reste problématique. Nous pouvons citer l'environnement de collaboration Platine [6] qui intègre et utilise l'ensemble des différents média à l'intérieur d'une même session.

Il en va de même pour les technologies réseau sans fil. Ces travaux de thèse ont été appliqués aux réseaux d'accès par satellite DVB-S2/RCS. Les autres technologies réseau sans fil tel que le wifi et le wimax possèdent des caractéristiques communes au satellite. Cependant certaines propriétés sont différentes selon les technologies. Une application de nos contributions nécessiteraient de prendre en compte les autres paramètres des technologies sans fil wifi et wimax.

Notre architecture à communication cross-layer dédiée offre la possibilité, à partir des données cross-layer, de construire des paramètres plus élaborés. Il s'agit par exemple des statistiques calculées pouvant être requises par la couche physique afin d'ajuster la puissance de transmission radio dans un réseau wifi, ou par la couche réseau, afin de trouver des routes sans congestion ni collision.

$STR = \frac{SuccessfulTransmissions}{AttemptedTransmissions}$ (Successful Transmission Rate), qui indique la qualité des transmissions en donnant le rapport des transmissions réussies sur celles attendues.

$CCR = \frac{\mathrm{SuccessfulAttemptstoFindaClearChannel}}{\mathrm{TotalAttemptstoFindaClearChannel}}$ (Clear Channel Rate), qui indique les possibilités de trouver le canal libre en donnant le rapport des tests réussis sur le nombre de tests.

L'élaboration de tels paramètres nécessite également d'être approfondie.

Toujours concernant l'architecture à communications cross-layer, les données mises à disposition sont uniquement visibles dans la pile protocolaire locale. Il y a la possibilité d'avoir une vue globale, qui fonctionnerait approximativement de la même manière que l'architecture CrossTalk [63]. À savoir chaque fois qu'un paquet est transmis à un nœud, la tiers entité cross-layer y ajoute une information locale. Le nœud recevant ce paquet, récupère la donnée distante et l'enregistre dans une liste d'information globale du CL_entity. Ce mécanisme de vue globale pour l'architecture cross-layer est également une piste à développer.

Nous prévoyons d'intégrer l'ensemble des contributions proposées dans ces travaux de thèse, dans une architecture globale à QdS hétérogène telle qu'EuQoS [50] [51]. En effet, nos travaux se positionnent au niveau réseau d'accès. La Qualité de Service, pour qu'elle soit complètement fonctionnelle, nécessite une portée de bout en bout comme celle proposée dans le système EuQoS.

8. Bibliographie

[1] Jain B. N., Agrawala A. K., *Open Systems Interconnection: Its Architecture and Protocols.* s.l. : Elsevier, 1990.

[2] R., Braden., *Requirements for Internet Hosts - Communication Layers, RFC 1122.* s.l. : IETF Network Working Group, Octobre 1989.

[3] ITU-T., *TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS - Quality of service and performance - End-user multimedia QoS categories, Recommandation G.1010.* Novembre 2001.

[4] R., STEINMETZ., "Human Perception of Jitter and Media Synchronisation." s.l. : IEEE-JSAC, Janvier 1996, Issue 1, Vol. 14.

[5] B., SHNEIDERMAN., "Response Time and Display Rate in Human Performance with Computers." s.l. : ACM-Computing Surveys, Septembre 1984, Issue 3, Vol. 16.

[6] D.RAYMOND, V.BAUDIN, K.KANENISHI., *Distant e-learning using synchronous collaborative environment "Platine".* Miami, USA : IEEE Sixth International Symposium on Multimedia Software Engineering, Décembre 2004.

[7] 802.11, IEEE Std., *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* Mars 2007.

[8] S. Mangold, S. Choi, G. R. Hiertz., *Analysis of IEEE 802.11e for QoS Support in Wireless Lans.* s.l. : IEEE Wireless Communications, Décembre 2003.

[9] Clark D., Shenker S., L. Zhang., *Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanisms.* Baltimore, MD : Proc. SIGCOMM '92, Aout 1992.

[10] S., Floyd., *Issues in Flexible Resource Management for Datagram Networks.* s.l. : Workshop on Very High Speed Networks, Mars 1992.

[11] Jamin S, Shenker S., *An Admission Control Algorithm for Predictive Real-Time Service.* San Diego, CA : International Workshop on Network and Operating System Support for Digital Audio and Video, Novembre 1992.

[12] C., Partridge., *A Proposed Flow Specification, RFC 1363.* s.l. : IETF Network Working Group, Juillet 1992.

[13] Shenker S., Clark D., *A Service Model for the Integrated Services Internet.* s.l. : Work in Progress, Octobre 1993.

[14] Zhang L., Deering S., *RSVP: A New Resource ReSerVation Protocol.* s.l. : IEEE Network, 1993.

[15] Braden R., Clark D., Shenker S., *Integrated Services in the Internet Architecture: an Overview, RFC 1633.* s.l. : IETF Network Working Group, Juiller 1994.

[16] D., Clark., *The Design Philosophy of the DARPA Internet Protocols.* s.l. : ACM SIG-COMM, Aout 1988.

[17] Blake S., Black D., Carlson M., *An Architecture for Differentiated Services, RFC 2475.* s.l. : IETF Network Working Group, Décembre 1998.

[18] Jacobson V., Poduri K., *An Expedited Forwarding PHB, RFC 2598.* s.l. : IETF Network Working Group, Juin 1999.

[19] Heinanen J., Baker F., Weiss W., *Assured Forwarding PHB Group, RFC 2597.* s.l. : IETF Network Working Group, Juin 1999.

[20] Information Sciences Institute, University of Southern California., *Transmission Control Protocol, RFC 793.* s.l. : IETF Network Working Group, Septembre 1981.

[21] Allman M., Paxson V., Stevens W., *TCP Congestion Control, RFC 2581.* s.l. : IETF Network Working Group, Avril 1999.

[22] Postel, J., *User Datagram Protocol, RFC 768.* s.l. : IETF Network Working Group, Aout 1980.

[23] L-A. Larzon, M. Degermark, S. Pink., *The Lightweight User Datagram Protocol (UDP-Lite), RFC 3828.* s.l. : IETF Network Working Group, Juillet 2004.

[24] , *AMR Speech Codec - General description, TS 26.071 version 5.0.0.* s.l. : 3rd Generation Partnership Project, Juin 2002.

[25] *AMR Speech Codec - Frame Structure, TS 26.101 version 5.0.0.* s.l. : 3rd Generation Partnership Project, Juin 2002.

[26] ITU-T., *Video Coding for Low Bit Rate Communication, Recommendation H.263* . Janvier 2005.

[27] —. *Advanced Video Coding for Generic Audiovisual Services, Recommendation H.264.* Mars 2009.

[28] ISO/IEC., *Information Technology Coding of Audio-Visual Objects, International Standard 1446 (MPEG).* Janvier 2000.

[29] E. Kohler, M. Handley, S. Floyd., *Datagram Congestion Control Protocol (DCCP), 4340.* s.l. : IETF Network Working Group, Mars 2006.

[30] Floyd S., Kohler E., *Profile for DCCP Congestion Control ID 4: the Small-Packet Variant of TFRC Congestion Control.* s.l. : fnivor.

[31] K. Ramakrishnan, S. Floyd, D. Black., *The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168.* s.l. : IETF Network Working Group, Septembre 2001.

[32] Lochin E., Dairaine L., Jourjon G., *Guaranteed TCP Friendly Rate Control (gTFRC) for DiffServ/AF Network.* s.l. : IETF Internet Draft, Aout 2006.

[33] Nivor, F., *Experimental Study of DCCP for Multimedia Applications.* Toulouse, France : Conference on emerging Networking EXperiments and Technologies, CoNEXT 2005, Octobre 2005.

[34] E., Exposito., *Spécification et mise en oeuvre d'un protocole de tranport orienté qualité de service pour les applications multimédia.* Toulouse : Rapport LAAS-CNRS, Décembre 2003.

[35] ITU-T., *Packet-based multimedia communications Systems, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS.* s.l. : Recommandation H.323, Juin 2006.

[36] Rosenberg J., Schulzrinne H., Camarillo G., *SIP: Session Initiation Protocol, RFC 3261.* s.l. : IETF Network Working Group, Juin 2002.

[37] H. Schulzrinne, S. Casner, R. Frederick., *RTP: A Transport Protocol for Real-Time Applications, RFC 3550.* s.l. : IETF Network Working Group, Juillet 2003.

[38] Camarillo G., Marshall W., *Integration of Resource Management and SIP, RFC 3312.* s.l. : IETF Network Worling Group, Octobre 2002.

[39] D. Durham, J. Boyle, R. Cohen., *The COPS (Common Open Policy Service) Protocol, RFC 2748.* s.l. : IETF Network Working Group, Janvier 2000.

[40] Veltri L., Salsano S., *SIP Extensions for QoS support.* s.l. : IETF Internet Draft, Avril 2003.

[41] R. Hancock, G. Karagiannis, J. Loughney., *Next Steps in Signaling (NSIS): Framework, RFC 4080.* s.l. : Network Working Group, Juin 2005.

[42] Alcatel Alenia Space France-Espagne, Telespazio, LAAS-CNRS, et al., *Satellite-based communications systems within IPv6 networks.* s.l. : IST Project, 2008.

[43] Deering S., Fenner W., *Multicast Listener Discovery for IPv6, RFC 2710.* s.l. : IETF Network Working Group, Octobre 1999.

[44] Dierks T., Rescorla E., *The Transport Layer Securoty (TLS) Protocol, RFC 5246.* s.l. : IETF Network Working Group, Aout 2008.

[45] Rescorla E., Modadugu N., *Datagram Transport Layer Security, RFC 4347.* s.l. : IETF Network Working Group, Avril 2006.

[46] Duquerroy L., Josset S., *SatIPSec: An Optimized Solution for Securing Multicast and Unicast Satellite Transmissions.* Californie, USA : ICSSC, May 2004.

[47] Harney H., Meth U., Colegrove A., *GSAKMP: Group Secure Association Key Management Protocol, RFC 4535.* s.l. : IETF Network Working Group, Juin 2006.

[48] Wallner D., Harder E., Agee R., *Key Management for Multicast: Issues and Architectures, RFC 2627.* s.l. : IETF Network Working Group, Juin 1999.

[49] Johnson D., Perkins C., Arkko J., *Mobility Support in IPv6, RFC 3775.* s.l. : IETF Network Working Group, Juin 2004.

[50] Dugeon O., Morris D., Monteiro E., *End to End Quality of Service over Heterogeneous Networks (EuQoS).* s.l. : Proceedings of Network Control and Engineering for QoS, Security and Mobility, 2005.

[51] T. Braun, M. Diaz, J. E. Gabeiras., *End-to-End Quality of Service Over Heterogeneous Networks.* s.l. : Springer, Aout 2008.

[52] Racaru, S. F., *Conception et validation d'une architecture de signalisation pour la garantie de qualité de service dans l'Internet multi-domaine, multi-technologie et multi-service.* s.l. : Rapport de thèse LAAS-CNRS, Octobre 2008.

[53] Farrel A., Vasseur J. P., *A Path Computation Element (PCE)-based Architecture, RFC 4655.* s.l. : IETF Network Working Group, Aout 2006.

[54] Projet EuQoS, Delivrables., *Definition of Business, Communication and QoS models - Intermediate, D1.1.1.*

[55] —. *Business models and system design specification, D1.1.3.*

[56] E. Rosen, A. Viswanathan, R. Callon., *Multiprotocol Label Switching Architecture, RFC 3031.* s.l. : IETF Network Working Group, 2001.

[57] Ramakrishnan K., Floyd S., Black D., *The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168.* s.l. : IETF Network Working Group, Septembre 2001.

[58] Srivastava V., Motani M., *Cross-Layer Design: A Survey and the Road Ahead.* s.l. : IEEE Communications Magazine, Décembre 2005.

[59] G., Wu., *Interactions between TCP and RLP in Wireless Internet.* Rio de Janeiro, Brésil : Proceedings of IEEE GLOBECOMP9, Décembre 1999.

[60] Sudame P., Badrinath B. R., "On Providing Support for Protocol Adaptation in Mobile Wireless Networks." s.l. : Mobile Networks and Applications, Janvier 2001, Issue 1, Vol. 6.

[61] Q. Wang, M.A. Abu-Rgheff., "Cross-layer Signalling for Next-Generation Wireless Systems." s.l. : IEEE WCNC, Mars 2003, Vol. 2.

[62] W. Berrayana, H. Youssef, S. Lohier., *Proposition of a cross-layer architecture model for the support of QoS in ad-hoc networks.* Lisboa, Portugal : CoNEXT, 2006.

[63] Winter, R., *CrossTalk: A Data Dissemination-Based Crosslayer Architecture or Mobile Ad Hoc Networks.* s.l. : Proceedings ASWN, Juin 2005.

[64] V. T. Raisinghani, S. Iyer., *ECLAIR: An efficient cross layer architecture for wireless protocol stacks.* s.l. : WWC, 2004.

[65] S. V. Adve, A. F. Harris, C. J. Hughes., *The Illinois GRACE Project: Global Resource Adaptation through CoopEration.* s.l. : Proceedings of the Workshop on SHAMAN, Juin 2002.

[66] K. Chen, S. H. Shan, K. Nahrstedt., "Cross- Layer Design for Data Accessibility in Mobile Ad Hoc Networks." s.l. : Wireless Personal Communications, Avril 2002, Issue 1, Vol. 21.

[67] Project, SatSix., "Cross-Layer Interaction Model, Delivrable 2000-4." 2008.

[68] F. Nivor, M. Gineste, C. Baudoin, P. Berthou, T. Gayraud., *Cross-layer anticipation of resource allocation for multimedia applications based on SIP signaling over DVB-RCS Satellite System.* Budapest, Hongrie : IP Networking over Next-generation Satellite Systems Workshop, INNSS 2007, Juillet 2007.

[69] F. Nivor, P. Berthou, S. Abdellatif, T. Gayraud., *Amélioration de l'Allocation Dynamique de Ressource dans un Système Satellite DVB-S/RCS.* Tozeur, Tunisie : Colloque Francophone sur l'Ingénierie des Protocoles, CFIP 2006, Novembre 2006.

[70] F. Nivor, P. Berthou, S. Abdelatif, T. Gayraud., *Optimization of a Dynamic Resource Allocation in DVB-S/RCS Satellite Networks.* Rome, Italie : International Congess ANIPLA 2006, Novembre 2006.

[71] Z. Jiang, Y Li., *A Predictive Demand Assignment Multiple Access Protocol for Broadband Satellite Networks Supporting Internet Applications.* New York, USA : IEEE ICC, Avril 2002.

[72] Lee, K .D., "An Efficient Real-Time Method for Improving Intrinsic Delay of Capacity Allocation in Interactive GEO Satellite Networks." s.l. : IEEE Transaction on Vehicular Technology, Mars 2004, Issue 2, Vol. 53.

[73] L. Chisci, R. Fantacci, T. Pecorella., "Predictive Bandwidth Control for GEO Satellite Networks." s.l. : IEEE ICC, Juin 2004, Vol. 7.

[74] L. Chisci, R. Fantacci, T. Pecorella., "Multi-terminal dynamic bandwidth allocation in GEO Satellite Networks." s.l. : IEEE VTC, Mai 2004, Vol. 5.

[75] F. Delli Priscoli, A. Pietrabissa., "Design of a bandwidth-on-demand protocol for satellite networks modeled as timedelay systems." s.l. : Automatica, Mai 2004, Issue 5, Vol. 40.

[76] F. Nivor, M. Gineste, M. Diaz., *Cross-Layer Rate Control for VoIP Applications using Codec Switching over DVB-S2/RCS.* San Diego, USA : International Communication Satellite System Conference, ICSSC 2008, Juin 2008.

[77] M. Gineste, F. Nivor., *SIP-based Resource Allocation for Interactive Multimedia Applications over DVB-S2/RCS Satellite System.* St Petersbourg, Russie : International Conference on Telecommunications, ICT 2008, Juin 2008.

[78] J. Salim, H. Khosravi, A. Kleen., *Linux Netlink as an IP Services Protocol, RFC 3549.* s.l. : IETF Network Working Group, Juillet 2003.

[79] D. Booth, H. Haas, F. McCabe., *Web Services Architecture.* s.l. : W3C Working Group, Février 2004.

[80] E. Christensen, F. Curbera, G. Meredith., *Web Services Description Language (WSDL) 1.1.* s.l. : W3C, Mars 2001.

[81] T. Bray, J. Paoli, C. M. Sperberg-McQueen., *Extensible Markup Language (XML) 1.0.* s.l. : W3C, Novembre 2008.

[82] L. Clement, A. Hately, T. Rogers., *UDDI Version 3.0.2 : Specification.* s.l. : OASIS Standard, Octobre 2004.

[83] D. Box, D. Ehnebuske, G. Kakivaya., *Simple Object Access Protocol (SOAP) 1.1.* s.l. : W3C, Mai 2000.

[84] , Internet Assigned Numbers Authority. *http://www.iana.org/.* [Online]

[85] N. Freed, N. Borenstein., *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, RFC 2045.* s.l. : IETF Network Working Group, Novembre 1996.

[86] —. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, RFC 2046.* s.l. : IETF Network Working Group, Novembre 1996.

[87] M. Westerlund, I. Johansson., *RTP Payload Format for G.719, RFC 5404.* s.l. : IETF Network Working Group, Janvier 2009.

[88] Even, R., *RTP Payload Format for H.261 Video Streams, RFC 4587.* s.l. : IETF Network Working Group, Aout 2006.

[89] CQ-Software., *Margouilla Runtime.* 2002.

[90] F. Delli Priscoli, T.Inzerilli, V.Morsella., *Broadband Access for High-Speed Multimedia via Satellite (Brahms).* s.l. : IST project.

[91] Alphand, O., *Architecture à qualité de service pour systèmes satellites DVB-S/RCS dans un contexte NGN.* s.l. : Rapport LAAS N°05672, Décembre 2005.

[92] Royal Institute of Technology (KTH, Stockholm)., *Minisip : a SIP User Agent .* s.l. : http://www.minisip.org/.

[93] WellX Telecom, Antisip SARL, A. MOIZARD., *Partysip : a SIP proxy server.* 2003.

[94] P. Seeling, F. Fitzek, M. Reisslein., *Video Traces for Network Performance Evaluation.* s.l. : Springer, Novembre 2006.

[95] D. L. Mills., *Network Time Protocol (Version 3) Specification, Implementation and Analysis, RFC 1305.* s.l. : IETF Network Working Group, Mars 1992.

[96] T. Williams, C. Kelley., *Gnoplot: An Interactive Plotting Program.* s.l. : http://www.gnuplot.info, 2004.

[97] Lonvick, C., *The BSD syslog Protocol, RFC 3164.* s.l. : IETF ?Network Working Group, Aout 2001.

[98] S. Cohen, M. Benger, R. Peleg., *HTTP over UDP, Final Report.* s.l. : Spring, 2000.

[99] T. Nixon, A. Regnier., *SOAP over UDP version 1.1.* s.l. : Ram Jeyaraman, Janvier 2009.

[100] I. Melhus, T. Gayraud, F. Arnal, L. Fan, F. Nivor, M. Gineste, A. Pietrabissa., *SATSIX Cross-layer Architecture.* Toulouse, France : International Workshop on. Satellite and Space Communications, Octobre 2008.

Bibliographie de l'auteur

[30] Floyd S., Kohler E., *Profile for DCCP Congestion Control ID 4: the Small-Packet Variant of TFRC Congestion Control.* s.l. : fnivor.

[33] Nivor, F., *Experimental Study of DCCP for Multimedia Applications.* Toulouse, France : Conference on emerging Networking EXperiments and Technologies, CoNEXT 2005, Octobre 2005.

[68] F. Nivor, M. Gineste, C. Baudoin, P. Berthou, T. Gayraud., *Cross-layer anticipation of resource allocation for multimedia applications based on SIP signaling over DVB-RCS Satellite System.* Budapest, Hongrie : IP Networking over Next-generation Satellite Systems Workshop, INNSS 2007, Juillet 2007.

[69] F. Nivor, P. Berthou, S. Abdellatif, T. Gayraud., *Amélioration de l'Allocation Dynamique de Ressource dans un Système Satellite DVB-S/RCS.* Tozeur, Tunisie : Colloque Francophone sur l'Ingénierie des Protocoles, CFIP 2006, Novembre 2006.

[70] F. Nivor, P. Berthou, S. Abdelatif, T. Gayraud., *Optimization of a Dynamic Resource Allocation in DVB-S/RCS Satellite Networks.* Rome, Italie : International Congess ANIPLA 2006, Novembre 2006.

[76] F. Nivor, M. Gineste, M. Diaz., *Cross-Layer Rate Control for VoIP Applications using Codec Switching over DVB-S2/RCS.* San Diego, USA : International Communication Satellite System Conference, ICSSC 2008, Juin 2008.

[77] M. Gineste, F. Nivor., *SIP-based Resource Allocation for Interactive Multimedia Applications over DVB-S2/RCS Satellite System.* St Petersbourg, Russie : International Conference on Telecommunications, ICT 2008, Juin 2008.

[100] I. Melhus, T. Gayraud, F. Arnal, L. Fan, F. Nivor, M. Gineste, A. Pietrabissa., *SATSIX Cross-layer Architecture.* Toulouse, France : International Workshop on. Satellite and Space Communications, Octobre 2008.

TITLE

Communication Architecture for Interactive Multimedia Applications in Wireless Networks

SUMMARY

Those thesis works are done in the wireless access networks context, and especially in satellite networks. They present some issues concerning the deployment of interactive multimedia applications which require low end to end delays and some Quality of Service (QoS) from the communication system.

We propose in this thesis to use session signaling information from SIP-based multimedia applications in order to adjust the system communication parameters. This is done by using a “cross-layer” approach to significantly increase the system reactivity.

In order to deploy the proposed contributions in a real communication system, a cross-layer architecture is proposed and developed.

The proposed solutions are evaluated in emulated and real wireless platforms.