



HAL
open science

Conception et évaluation d'un système local au terminal pour une gestion unifiée de la Qualité de Service.

Guillaume Lathoumetie

► **To cite this version:**

Guillaume Lathoumetie. Conception et évaluation d'un système local au terminal pour une gestion unifiée de la Qualité de Service.. Congrès EDSYS 2011, May 2011, TOULOUSE, France. 6p. hal-00594029

HAL Id: hal-00594029

<https://hal.science/hal-00594029>

Submitted on 18 May 2011

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

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

Conception et évaluation d'un système local au terminal pour une gestion unifiée de la Qualité de Service.

Guillaume Lathoumetie (guillaume.lathoumetie@laas.fr)

LAAS-CNRS, Toulouse

Université Paul Sabatier, Toulouse

Thèse encadrée par : Thierry Gayraud (LAAS-CNRS groupe OLC) et Pierre Saucourt-Harmel (Wi Telecoms)

Résumé *Actuellement, de plus en plus de terminaux sont capables de communiquer en utilisant plusieurs standards de réseaux sans fil, parfois de manière simultanée. La gestion de ces différents protocoles et le partage des ressources qui leurs sont accordées est un axe d'étude qui nécessite d'appréhender plusieurs thèmes variés. Ils font partie intégrante des innovations des télécommunications. Dans cette optique, le premier concept étudié a été la «radio logicielle» ou SDR¹, car nous avons besoin de sa flexibilité nouvelle pour développer des prototypes de réseaux sans fil. Le second axe de la thèse est plus centré sur la problématique intrinsèque de la gestion de Qualité de Service, à travers l'étude des différents mécanismes de gestion de files d'attente et d'ordonnement qui sont implémentés dans les couches protocolaires, et leur évaluation expérimentale.*

A. INTRODUCTION ET PROBLÉMATIQUE

Les terminaux de nouvelle génération embarquent un grand nombre de protocoles de communication sans fil, nombre voué à croître avec l'apparition de nouveaux standards répondant à des besoins précis tels que le WiMAX 802.16 [12] pour l'accès internet sans fil longue distance ou le NFC pour les communications en champ proche (paiement bancaire etc. [8]). Il est donc nécessaire d'améliorer leur gestion afin de tirer le meilleur parti de chacun d'entre eux et assurer la meilleure expérience possible à l'utilisateur. Cela passe par l'élaboration d'algorithmes et de mécanismes qui permettent d'assurer une meilleure qualité de service (QoS) au sein du terminal. Ces algorithmes peuvent aussi bien reposer sur des échanges inter-couches que sur des protocoles de QoS existants.

Après avoir effectué un état de l'art des différentes méthodes qui permettent d'assurer la qualité de service au sein des protocoles actuels, nous nous sommes aperçu d'un facteur intéressant : la plupart des solutions proposées concernent des couches protocolaires de niveau réseau (IP, ATM niveau 3 etc. voir [4] et [5]) ainsi que les couches supérieures comme TCP et plus particulièrement ses évolutions SCTP [13] et MPTCP [7]. Ces solutions permettent aussi bien d'assurer de la QoS de bout en bout de la communication que de manière locale au terminal et connaissent une forte croissance.

En partant de ce constat, nous avons choisi d'étudier les différentes manières d'assurer la QoS à un niveau plus bas des piles protocolaires, au sein de la couche d'accès ou couche MAC². Le fait de fournir de la QoS au niveau de la couche accès limite la connaissance de l'architecture complète du réseau tel que cela est possible à des niveaux supérieurs. Toutefois cette limite est négligeable puisque nous nous concentrons sur la QoS locale au système qui opérera la plupart du temps entre un terminal et un point d'accès qui lui, possède la vue du réseau nécessaire. Ce choix présente plusieurs avantages : le premier est que ce niveau a moins été étudié pour les problèmes liés à la QoS, cela est en partie dû à l'architecture des périphériques de communication qui embarquent des couches MAC non reconfigurables (micro-programme et matériel spécifiques) contrairement aux couches supérieures qui sont elles logicielles. Ce fait a donc compliqué les expérimentations sur ces couches mais, à présent, les couches MAC sont plus complexes et nécessitent une implémentation logicielle pour assurer les différents mécanismes. Ainsi, ces couches deviennent abordables à des fins de recherche. Le second avantage d'effectuer la gestion de la QoS au niveau MAC vient du principe d'empilage des couches protocolaires. Si le service assuré par la couche la plus basse possible est le meilleur, cette qualité se propagera naturellement aux couches supérieures. C'est un facteur important, qui ne dispense cependant pas de continuer la gestion de QoS dans les couches supé-

¹Software Defined Radio, désigne l'ensemble des outils permettant d'effectuer de la radio logicielle.

²Medium Access Control, couche 2 du modèle OSI assurant la gestion de l'accès au médium de communication.

rieures, afin d'assurer les paramètres de transmission qui leur sont propres et bénéficier des particularités de chacune d'entre elles. Le dernier avantage est la proximité de la couche MAC vis à vis de la couche physique. La couche MAC, sans posséder la vue logique du réseau que la couche réseau possède, est proche des paramètres qui influent directement sur la qualité de la transmission comme le débit, le taux de perte et l'état du médium. Ces paramètres ont un rôle prépondérant sur la qualité de service fournie à l'utilisateur d'un périphérique et à sa gestion locale. Ce document présente les différents axes suivis lors de la première année de thèse, la première partie est consacrée à la gestion de profils de service et les mécanismes qui en découlent. La seconde partie présente une architecture de SDR basée sur l'utilisation de l'USRP³ et GNURADIO et se terminera par une conclusion et les recherches futures qui découlent de ces premiers travaux.

B. LA GESTION DE PROFILS DE SERVICE

L'évolution des réseaux de communications, depuis plusieurs décennies, passe par un besoin grandissant en qualité de service offerte à l'utilisateur. L'un des premiers facteurs qui permet l'amélioration de certains paramètres de services, tels que le débit, la gigue et le délai, est l'amélioration de la méthode avec laquelle les données sont gérées au sein des couches protocolaires. Dans les implémentations actuelles des divers protocoles de la couche d'accès, ainsi qu'aux autres niveaux, la méthode la plus répandue pour organiser ces données en transit est l'utilisation de files d'attente. Au début, l'organisation de ces files d'attente était la plus simple, basée sur l'utilisation de FIFO (First In First Out) mais, pour assurer une meilleure qualité de service, la notion de gestion de files d'attente est apparue. Cependant, lorsque le protocole utilise plusieurs classes de service pour ses flux, il est au préalable nécessaire de les classer pour les orienter vers les bonnes files d'attente. Ce mécanisme simple fera l'objet de la première partie de ce chapitre.

La gestion de files d'attente quant à elle, est en charge d'assurer l'acheminement, le retardement ou même l'élimination des paquets stockés dans une seule et même file. Les différents mécanismes de gestion de files seront donc présentés dans la seconde partie de ce chapitre. Cependant, afin de répondre

plus précisément aux besoins de différenciation des services offerts à l'utilisateur du périphérique, plusieurs types de files ont été insérés au sein d'une même couche protocolaire. Ainsi chaque file peut correspondre à un type de service (ou un flux de service) particulier avec ses contraintes propres sur les paramètres cités précédemment. Afin d'assurer la répartition des flux de données entre les différentes files, il est nécessaire de suivre une politique d'ordonnancement spécifique. Les mécanismes d'ordonnancement, et le choix d'un type par rapport à un autre, sont des sujets très étudiés ces derniers temps, avec l'apparition de la technologie WiMAX 802.16 qui laisse le choix au développeur de la politique d'ordonnancement à suivre. La troisième partie de ce chapitre présentera donc les différentes politiques d'ordonnancement de files et les mécanismes associés.

Les mécanismes de classification de flux, de gestion de file et d'ordonnancement sont génériques et s'appliquent à tous les types de données pouvant transiter dans les dites files. Cependant, comme l'étude porte sur la couche d'accès, le terme employé pour représenter l'unité de donnée traitée dans les files sera trame. La figure 1 présente une vue générale de leurs interactions.

1. Classification des flux

Les mécanismes de classification de flux sont souvent très simples et peuvent être indépendants ou liés au type d'ordonnancement qui suit. Par exemple la couche MAC du protocole WiMAX 802.16 utilise ce type de mécanisme de manière indépendante. Dans ce protocole, le classificateur fait correspondre l'identifiant de la connexion par laquelle transite la trame, avec la classe de service qui lui correspond. Ainsi, le classificateur peut ensuite insérer la trame dans la file correspondante sans pour autant que le standard ne spécifie un ordonnancement spécifique. Cependant, une particularité s'applique aux flux temps réel qui, une fois identifiés, sont directement transférés sans attendre d'être ordonnancés avec les autres.

Le classificateur de trames du WiMAX 802.16 ajoute aussi une estampille temporelle à chaque trame, afin de faciliter le fonctionnement du gestionnaire de files pour une éventuelle élimination de cette trame si son temps d'attente dans la file dépasse un seuil défini. Les données utilisées par le mécanisme de classification peuvent être représen-



Figure 1 – Chaîne des mécanismes de gestion de profils de service

³Universal Software Radio Peripheral, matériel de SDR défini pour être le plus flexible possible.

tées sous la forme d'une table, qui fait correspondre l'identifiant de connexion et la classe de service à laquelle elle appartient. Cette méthode de représentation dépend des choix du développeur et ne représente en rien un devoir d'implémentation.

Il est important de savoir que la méthode de classification, bien qu'elle soit un mécanisme à part entière, dépend fortement de l'ordonnement qui suit. En effet, des choix de classification découlent leur interprétation en sortie de file par l'ordonneur, et donc si l'ordonneur ne sait pas interpréter la classification faite au préalable, les trames ne sont pas traitées telles qu'elles doivent l'être. Dans la suite du document, les classificateurs liés aux mécanismes d'ordonnement seront présentés avec ledit mécanisme.

2. Gestion de files

Les mécanismes de gestion de files d'attente sont un ensemble de méthodes qui évitent qu'un engorgement se produise au sein d'une file. Ces mécanismes peuvent s'appliquer dès qu'une file d'attente est utilisée pour stocker les trames en attente d'émission ou de réception, afin d'augmenter la qualité de service offerte à l'utilisateur du périphérique. Cependant, il existe plusieurs types de mécanismes qui sont plus ou moins prédictifs de ces congestions, ils sont détaillés dans la suite.

Un des mécanisme de gestion de files d'attente le plus répandu est Random Early Detection (RED [6]). Il est basé sur l'utilisation d'un paramètre nommé «threshold» qui est un intervalle de seuils dans lequel s'effectue le choix d'éliminer des trames de la file lors d'un risque de congestion. L'intervalle threshold [min, max] est défini par le développeur du mécanisme et limitera les bornes d'élimination de trames, ces valeurs correspondent à des tailles moyennes d'occupation de la file. Ainsi, pour chaque arrivée d'une trame à stocker dans la file, le mécanisme calcule la valeur moyenne d'occupation de la file actuelle et peut ainsi déterminer si la trame est rejetée ou non. La probabilité de rejet augmente à partir du seuil minimum, de manière linéaire jusqu'à arriver au seuil maximum, après lequel toutes les trames entrantes sont rejetées. La méthode de calcul de la probabilité de rejet de la trame peut varier suivant les besoins du développeur. Plus de détails sur les méthodes de calcul de la taille moyenne de la file et de la probabilité de rejet d'une trame sont disponibles dans [6]. Le développeur doit aussi décider de la valeur de la probabilité, calculée pour chaque trame, au-delà de laquelle elle est rejetée. La figure 2 résume la probabilité de rejet d'une trame, en fonction du remplissage moyen de la file.

Plusieurs évolutions de RED ont été développées, comme RIO (RED with In and Out) et WRED (Weighted RED) qui apportent principalement la différenciation de classes de flux afin d'avoir plusieurs seuils d'élimination de trames.

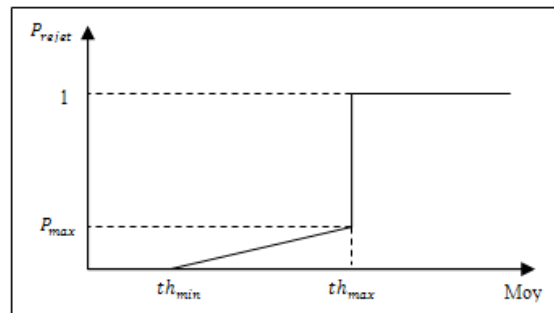


Figure 2 – Fonction de probabilité d'élimination de trames de RED

3. Ordonnement de files

Historiquement, l'ordonnement fut d'abord appliqué à une seule file d'attente. Cela correspondait à insérer une notion de temps sur les trames traitées à l'aide d'estampilles temporelles. Grâce à cette marque temporelle, le système pouvait déterminer le point d'insertion de la trame dans la file et ainsi traiter les trames en fonction. Mais, afin de différencier des classes de services au sein des protocoles, plusieurs files sont utilisées afin d'offrir une plus grande séparation des flux pour la QoS, le besoin de mécanismes plus évolués est apparu. Ces méthodes apportent plusieurs avantages par rapport à l'accumulation de plusieurs flux sur une même file : L'interdépendance des flux n'est plus présente, puisque chaque flux est lui-même représenté par une file et non pas cumulé dans une seule avec d'autres. Les délais de traitement peuvent aussi être améliorés puisque cette séparation permet de simplifier la méthode d'ordonnement.

La première méthode implémentée pour traiter plusieurs files de manière la plus équitable possible fut l'utilisation d'un mécanisme Round-Robin. Ainsi, chaque file était parcourue de manière séquentielle, afin d'en extraire la première trame en attente. Ce mécanisme apportait une première réponse au problème de gestion de files multiples, mais en pratique, les flux de données arrivent de manière stochastique et les trames sont de tailles différentes, ce qui fait que les files ne peuvent être traitées de manière égale ainsi. Le principe de fonctionnement de ces mécanismes est présenté en figure 3.

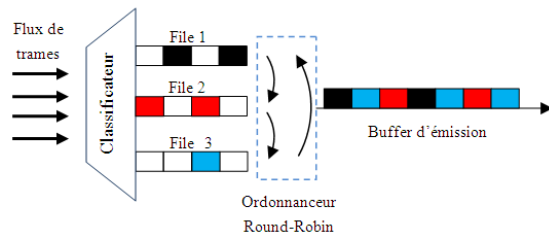


Figure 3 – Principe des mécanismes d'ordonnement de files

Suite à ce constat, des mécanismes d'ordonnement de files plus évolués ont été développés. Ainsi, plusieurs types d'algorithmes d'ordonnement ont émergé, et nous pouvons les classer en deux grands groupes :

Mécanisme	Avantages	Inconvénients
Random Early Detection	Définit des seuils d'élimination de trames au sein des files, afin d'éviter les engorgements.	Ne différencie pas les types de flux concurrents qui peuvent être mis en œuvre, et ne gère donc pas les priorités.
RED with In and Out	Affine le mécanisme RED en ajoutant un second type de service pris en compte.	Fonctionnement plus complexe que RED.
Weighted RED	Traite la différenciation de services sur plus de deux niveaux. Offre ainsi une plus grande finesse dans le traitement des différents flux et leur priorité.	Complexité de traitement accrue par rapport aux deux mécanismes précédents.
Priority Queuing	Insertion de la notion de priorité entre différentes files à ordonnancer. Très utile pour les flux à forts besoins temps réel.	Risque d'apparition de famine pour les flux moins prioritaires.
Fair Queuing	Équité de traitement théorique des différents flux.	En pratique, l'hétérogénéité des flux rend FQ moins apte à les traiter équitablement.
Weighted Fair Queuing	Améliore FQ en ajoutant un poids aux files, afin d'obtenir une priorisation de certains flux tout en garantissant une valeur de bande passante à chaque file.	La complexité de l'algorithme rend ce mécanisme très demandeur de ressources si le nombre de files utilisées est important.
Stochastic Fairness Queuing	Fonctionnement simple, peu nécessitant en ressources de calcul. Impact mémoire limité grâce au nombre de files fixé, contrairement à FQ.	Pas de garanties suffisantes pour les flux temps réel.
Earliest Deadline First	Préemptif, réponds aux besoins des flux temps réel.	Difficile à mettre en œuvre et fonctionnement peu fiable lors de charges importantes.
Class Based Queuing	Plus de précision pour la gestion des flux grâce à la différenciation de classes de service.	Délais de traitement importants pour les flux temps réel.
Start Time Fair Queuing	Améliore le principe de WFQ en diminuant sa complexité de traitement et améliore les délais, même pour des débits variables.	Manque de réactivité et délais trop longs pour des flux temps réel.
Self Clocked Fair Queuing	Évolution de WFQ, dont la méthode de calcul des temps a été simplifiée, afin de diminuer les traitements.	Délai maximum de transfert de trame encore important pour du temps réel, à cause des calculs mis en jeu.

Figure 4 – Récapitulatif des caractéristiques des mécanismes de gestion de file et d'ordonnancement

- Les mécanismes d'ordonnancement qui offrent une garantie de qualité de service sur un paramètre précis, soit le délai, soit le débit.
- Les mécanismes d'ordonnancement qui tentent de fournir un partage équitable des ressources entre les flux concurrents.

Un exemple de la première catégorie de mécanisme d'ordonnancement, et aussi l'un des plus simples, est l'algorithme Priority Queuing (PQ). Chaque file d'attente possède une priorité différente et l'ordonnanceur parcourt chaque file en privilégiant la file la plus prioritaire. Le mécanisme ne possède pas de limitations au niveau du nombre de niveaux de priorités, afin de couvrir le plus de besoins possibles au niveau de la qualité de service offerte à l'utilisateur du périphérique. Cependant, la complexité de l'algorithme de parcours des files et l'espace mémoire nécessaire augmente en fonction du nombre de classes désirées par le développeur. Chaque file fonctionne sur le principe de la FIFO basique, la priorisation effectuée ensuite par l'ordonnanceur suffit à assurer le fonctionnement du mécanisme.

Le premier mécanisme d'ordonnancement ayant cherché à assurer un traitement équitable des flux est l'algorithme Fair Queuing (FQ [1]). Comme le mécanisme Priority Queuing, le mécanisme Fair

Queuing est ancien et son fonctionnement basique. Son principe est d'avoir un ordonnancement de flux le plus équitable possible. Ainsi, le classificateur doit orienter chaque flux vers une file propre, mais aucune priorisation n'est faite. Une fois la classification effectuée, l'ordonnanceur parcourt les files en suivant l'algorithme Round Robin, afin d'être le plus équitable possible.

Les dernières évolutions des mécanismes d'ordonnancement cherchent à cumuler les forces des deux grands groupes cités précédemment, sans pour autant hériter de leurs faiblesses. Les algorithmes comme Weighted Fair Queuing (WFQ), Start Time Fair Queuing (STFQ) en sont des exemples représentatifs. Le tableau en figure 4 présente un récapitulatif des avantages et inconvénients des différents mécanismes de gestion de files et d'ordonnancement de files extraits des résultats de [11], [3], d'études supplémentaires et des définitions des différents mécanismes.

C. LA RADIO LOGICIELLE

Le nombre grandissant de protocoles de communication sans fil, embarqués dans les terminaux actuels, répond à la variété des besoins d'applications toujours plus précis et hétérogènes. Cette mixité fait apparaître un fort besoin de flexibilité, pour assu-

rer le basculement d'un standard de communication vers l'autre, tout en offrant la meilleure expérience à l'utilisateur. La radio logicielle consiste à répondre à ce besoin, en déplaçant la complexité de traitement habituellement matérielle, vers une complexité logicielle. Ainsi avec une architecture matérielle simple composée de convertisseurs numériques analogiques et numériques analogiques pilotés par un FPGA⁴ ou par un DSP⁵, et des entrées RF⁶ (frontaux) correspondant aux différents protocoles que l'on veut embarquer, il est possible de faire de la radio logicielle avec un framework logiciel spécifique. Il existe plusieurs types de frameworks disponibles. Celui que nous avons choisi pour les expérimentations se nomme GNU Radio [2]. C'est un logiciel libre qui est donc facile d'accès contrairement à des frameworks propriétaires ; il possède aussi l'avantage d'avoir une communauté qui assure le support et le développement de différentes solutions.

La plateforme matérielle, qui sert de base à ces expérimentations, se nomme USRP pour Universal Software Radio Peripheral. Elle implémente des fonctionnalités simples de conversion du signal reçu, à l'aide de quatre convertisseurs analogique/numérique à 64 MHz sur 12 bits et quatre convertisseurs numérique/analogique à 128 MHz sur 14 bits pilotés par un FPGA ; la réception de signaux se fait à l'aide de cartes filles adaptées à des bandes de fréquences différentes. L'USRP fonctionne couplée avec un PC relié par un bus USB2 (débit théorique de 480 Mb/s) qui viendra récupérer les données converties afin de les traiter avec le framework GNU Radio. La figure 5 présente l'architecture matérielle de l'USRP avec ses composants principaux.

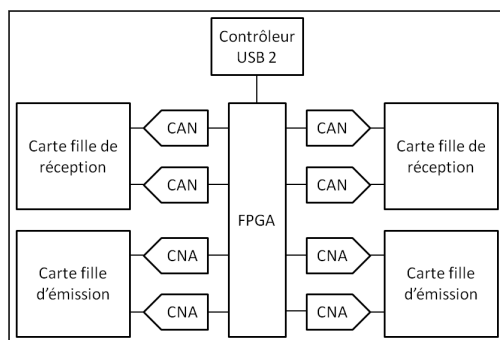


Figure 5 – Architecture matérielle de l'USRP

A cette carte mère viennent s'ajouter les cartes filles suivantes :

- La carte RFX900 est un émetteur/récepteur sur la bande 800-1000MHz, elle inclut un filtre sur la bande ISM 902-928 MHz installé pour éviter les perturbations de forts signaux hors bande. Cette carte a été choisie pour l'utilisation sur la bande de fréquence du GSM.
- La carte XCVR2450 est un émet-

teur/récepteur sur les bandes 2.4-2.5 GHz et 4.9-5.8 GHz son aspect double bande la rend très utile pour toutes les technologies 802.11.

Tout le matériel cité est présent en double, pour effectuer des communications indépendantes de solutions existantes. Ainsi, nous possédons deux USRP et quatre cartes-filles. Chaque USRP est couplée à un PC fonctionnant sous l'OS Ubuntu 10.10 avec le logiciel GNU Radio installé. Les expérimentations que nous avons menées ont principalement consisté à tester les capacités du couple GNU Radio/USRP, à l'aide de solutions déjà existantes telles que Airprobe, récepteur GSM permettant de visualiser les informations sur les canaux non cryptés, ou alors BBN 802.11, qui est une solution de couche MAC 802.11b qui ne fonctionne pas encore comme le précise le standard. Ces solutions sont encourageantes pour les possibilités qu'elles offrent, cependant plusieurs limites sont malheureusement apparues au fur et à mesure de nos expérimentations :

- La première est purement matérielle. En effet le FPGA de l'USRP n'est pas assez puissant quand les conditions de transferts atteignent des débits nécessitant une forte charge de calcul (problèmes avec le 802.11b à 54Mbps par exemple). Il est possible de modifier le microprogramme du FPGA pour que son fonctionnement corresponde aux besoins précis d'un standard, mais, dans ce cas-là, nous perdons la flexibilité de l'architecture.
- L'architecture même de GNU Radio, en particulier son scheduler, limite fortement les capacités d'implémentation d'une couche MAC entièrement effective. En effet, ce dernier traite les graphes de flux séquentiellement, ce qui rend la combinaison de fonctions d'émission et de réception au sein du même graphe assez complexe. Cependant, [10] propose une nouvelle méthode de conception pour pallier ce problème et affirme que le développement de certaines couches MAC basées sur celles du WiFi et du bluetooth a été fait.
- L'exécution de GNU Radio se fait dans l'espace utilisateur du système d'exploitation Ubuntu, et cela ne permet pas de bénéficier des meilleures performances possibles, en particulier à cause de l'utilisation du port USB2 limité par le contrôleur de l'USRP. Une solution à ce problème pourrait être de s'implanter directement dans le pilote USB d'Ubuntu, mais les besoins de GNU Radio en termes de puissance de calcul sont trop importants pour correspondre à ceux d'un pilote.

Après ces constatations, nous avons donc choisi de poursuivre les expérimentations sur des architec-

⁴Field Programmable Gate Array, type de circuit logique programmable.

⁵Digital Signal Processor, processeur spécialisé dans le traitement de signaux numériques.

⁶Radio Frequency, fréquences radios.

tures de test basées plutôt sur des outils de simulation de réseaux tels que NS ou OPNET.

D. CONCLUSION ET PERSPECTIVES

Les différentes méthodes de gestion et d'ordonnement de files étudiées nous ont permis de prendre conscience de l'étendue des possibilités offertes pour assurer une meilleure qualité de service au sein d'un terminal. Cependant, il est nécessaire d'avoir une vision globale du système afin d'éviter de dégrader les performances offertes. En effet, comme le précise [9], les interactions entre les différents mécanismes d'ordonnement des couches, en particulier entre la couche réseau et la couche liaison peuvent, suivant les cas, entraîner un surcoût de traitement des données (ajout de délai de transfert etc.) ou alors gréver fortement les performances de transfert de trafics prioritaires comme les trafics temps réel.

Il sera donc intéressant, par la suite, d'appréhender ce problème en mettant en oeuvre plusieurs expérimentations pour vérifier ces propos. L'enjeu primordial sera de paramétrer le plus finement possible les couches protocolaires, afin d'obtenir le meilleur compromis entre efficacité et impact sur les ressources du système.

Les mécanismes d'interactions cross-layer présentent aussi des aspects intéressants sur le partage des informations sur l'état des réseaux entre les différentes couches. Cet aspect, déjà étudié dans nos travaux préliminaires, pourra faire l'objet d'une investigation plus poussée pour mettre en oeuvre un mécanisme centralisé d'échange d'informations au sein du système. La création d'une entité centrale agissant sur le plan de contrôle entre les différentes couches (plus particulièrement, physique, liaison, réseau et

transport) peut représenter une idée intéressante à développer. Le plus difficile sera de sélectionner les informations pertinentes sans toutefois encombrer le fonctionnement global du système, cela peut passer par l'utilisation d'un framework de développement flexible de piles protocolaires tel que WiMPS, développé par l'entreprise Wi Telecoms, industriel partenaire de cette thèse CIFRE.

Une fois les meilleurs compromis déterminés, afin d'assurer la qualité de service sur une interface particulière, il restera à trouver des algorithmes pour déterminer quelle est l'interface la plus appropriée pour mettre en oeuvre la communication à l'instant précis. Un des principaux challenges de cet aspect est de prendre en compte l'évolutivité du système. En effet, suivant sa mobilité (micro-mobilité, macro-mobilité etc.), le système rencontre différents standards qui lui sont accessibles sur des durées différentes, et il est alors primordial de trouver un bon compromis sur le choix de basculement à faire pour éviter soit de trop basculer de standard soit de ne pas assez basculer. Ces algorithmes dépendent de plusieurs paramètres (comme le type de communication voulu : transfert de fichier, appel téléphonique etc.), qui influent sur le choix de l'interface la plus appropriée pour l'application. L'état des infrastructures alentours représente aussi une information importante pour effectuer le choix du basculement. Il est possible, par exemple, d'envisager un mécanisme d'échange d'informations entre le système et son environnement, afin de pouvoir sélectionner la base offrant la meilleure qualité de service, sans pour autant être saturée. Pour cela, la gestion des files d'attente représente un bon moyen d'avoir une vue précise de l'état d'encombrement des différentes bases. Cette partie s'inscrit dans le travail de recherche à venir dans la suite de la thèse.

RÉFÉRENCES

- [1] V. Bhargavan, Lu Songwu, and T. Nandagopal. Fair queuing in wireless networks : issues and approaches. *IEEE Personal Communications*, 1999.
- [2] Eric Blossom. *Exploring GNU Radio*, November 2004.
- [3] Y Cao and V.O.K Li. Scheduling algorithms in broadband wireless networks. *IEEE proceedings*, 2001.
- [4] Mohamad El Masri. *Contribution à la qualité de service dans les réseaux d'accès sans fil*. PhD thesis, Université Paul Sabatier, 2009.
- [5] P. Ferguson and G. Huston. *Quality of service : delivering QoS on the Internet and in corporate networks*. John Wiley and sons, 1999.
- [6] S Floyd and V Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1993.
- [7] A. Ford, C. Raiciu, and al. *TCP Extensions for Multipath Operation with Multiple Addresses draft-ietf-mptcp-multiaddressed-03*, 2011.
- [8] ISO. *ISO/IEC 18092 :2004 Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)*, 2004.
- [9] J. Manner, M. Kojo, and al. Exploitation of wireless link qos mechanisms in ip qos architectures. *Quality of service over next-generation data networks*, 2001.
- [10] G. Nychis, T. Hottelier, and al. Enabling mac protocol implementations on software-defined radios. *NSDI'09 Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, 2009.
- [11] M.G. Rashed and M Kabir. A comparative study of different queuing techniques in voip, video conferencing and file transfer. *Daffodil international university journal of science and technology*, January 2010.
- [12] IEEE Computer Society, the IEEE Microwave Theory, and Techniques Society. 802.16 – iee standard for local and metropolitan area networks – coexistence of fixed broadband wireless access systems, 2009.
- [13] R. Stewart. *Stream Control Transmission Protocol*, September 2007.