**MASTER EN SCIENCES INFORMATIQUES**

# Diversité dans la pile protocolaire TCP/IP

Thonard, Marc

*Award date:*
2012

*Awarding institution:*
Universite de Namur

Facultés Universitaires Notre-Dame de la Paix, Namur
Faculté d'informatique

Année Académique 2011-2012

# Diversité dans la pile protocolaire TCP/IP

Marc Thonard

**Mémoire présenté en vue de l'obtention du grade de master en Sciences Informatiques**

# Table des matières

# Liste des figures

# Liste des tableaux

## Liste des acronymes

**3G** 3 ème Génération
**4G** 4 ème Génération
**ACK** ACKnowledgment
**ASCII** American Standard Code for Information Interchange

**BER** Bit Error Rate

**CRC** Cyclic Redondancy Check

**dB** Decibel
**DoS** Denial of Service
**DSN** Data SubFlow Number
**DSM** Data Sequence Mapping

**EDPF** Earliest Delivery Path First
**EGC** Equal Gain Combining

**FTP** File Transfert Protocol

**Gb** Giga bit
**GB** Giga Byte
**GHz** Giga Hertz
**GM** Multiplexing Gain

**HDSPA** High Downlink Speed Packet Access
**HSPA** High Speed Packet Access
**HSUPA** High Speed Uplink Packet Access
**HTTP** HyperTex Transfer Protocol

**IEEE** Institute of Electrical and Electronics Engineers
**IP** Internet Protocol
**IPv4** Internet Protocol version 4
**IPv6** Internet Protocol version 6
**iSCSI** internet Smal Computer System Interface
**ISO** Internal Standard Organization

**LLC** Link Layer Control

**MAC** Media Access Control
**Mb** Mega bit
**MB** Mega Byte
**MIMO** Multiple Input Multiple Output
**MPTCP** Multi-Path Transmission Control Protocol
**MRC** Maximal Ratio Combining

**NAT** Network Address Translation

**QoS** Quality of Service

**RAM** Random Access Memory
**RF** Radio Frequency
**RFC** Requests For Comments
**RTT** Round-Trip Time

**SACK** Selective ACKnowledgment
**SAN** Storage Area Network
**SC** Selection Combining
**SCTP** Stream Control Transmission Protocol
**SM** Spatial Multiplexing
**SNR** Signal-to-Noise Ratio
**SSN** Subflow Sequence Number
**STC** Spatial-Time Coding
**SYN** SYNchronize

**TAS** Transmit Antenna Selection
**TCP** Transmission Control Protocol

**UDP** User Datagram Protocl
**UMTS** Universal Mobile Telecommunications System
**USB** Universal Serail Bus
**UTP** Unshielded Twisted Pair

**VM** Virtual Machine

**WCDMA** WideBand Code Division Multiple Access
**WiFi** Wireless Fidelity
**WiMP** WiFi Multi-Path

# 1 Remerciements

Je tiens tout particulièrement à remercier, pour son soutien, son courage et son engagement constant, mon épouse Nathalie sans quoi rien n'aurait été possible.

Un grand merci pour la disponibilité contante de mes parents et mes beaux-parents qui m'ont permis de dégager du temps à ce travail.

Je tiens à remercier vivement pour son avis d'expert, pour ses conseils judicieux ainsi que pour le temps qu'il m'a consacré Mr L. Schumacher, professeur à la Faculté Informatique de l'Université Notre de Dame de la Paix.

Je souhaite manifester ma gratitude à mes amis et plus spécialement à Thierry Delmotte pour sa présence et ses encouragements tout au long de ce travail.

Je voudrais témoigner ma reconnaissance au personnel dévoué à l'encadrement des élèves des cours à horaire décalé de la FUNDP pour leur énergie, leur disponibilité et leur motivation sans faille.

Je souhaite remercier la direction générale et la direction des systèmes de l'information du Centre Hospitalier Régional de la Citadelle pour l'aide qu'ils m'ont apportée.

Je voudrais remercier la société abNetwork S.A. pour la mise à ma disposition de tout le matériel réseau nécessaire à ce travail.

Je souhaite avoir une pensée pour ceux qui nous ont quittés trop tôt.

Merci à toutes et à tous, vous, qui m'avez de près ou de loin, aidé et encouragé à mener à bien ce travail.

Enfin, j'adresse un grand merci à mes deux correctrices, Maggy et Nathalie.

## 2 Résumé

A différents niveaux dans la pile protocolaire TCP/IP, le recours à plusieurs chemins simultanés a été proposé : SCTP et Multipath TCP en couche transport, multihoming en couche IP, systèmes MIMO en IEEE 802.11n et UMTS Rel'7. Notre projet visera à mettre en œuvre un prototype de combinaison de ces techniques, par exemple MPTCP + 802.11n, pour déterminer si les mérites respectifs de ces techniques se combinent ou pas.

Mot Clefs : Diversity, transmit selection, multipath

# 3 Introduction

De nos jours, la mobilité est un besoin grandissant dans le monde professionnel (techniciens itinérants, prospection commerciale, monitoring médical, etc) et privé (réseaux sociaux, chat, etc).

Depuis l'avènement des laptops, tablettes, ou autres smartphones, l'utilisateur mobile désire accéder rapidement à ses données via les différentes technologies de réseaux sans fil disponibles sur le marché.

Dès lors, le matériel mobile d'aujourd'hui embarque souvent plusieurs interfaces de technologie différente (UMTS/3G, WiFi IEEE 802.11b, g, n), toujours utilisées séparément.

Le passage d'un type de réseau à un autre se traduit souvent par une rupture de communication au niveau de l'application (vertical handover).

L'utilisateur a besoin d'avantage de robustesse au niveau du signal et les applications nécessitent un débit de données de plus en plus important (video streaming, video on demand, etc).

Nous proposons, dans ce mémoire, de combiner plusieurs technologies permettant d'améliorer la robustesse et la performance d'une transmission en utilisant plusieurs chemins simultanés.

L'objectif de ce mémoire est d'observer si leurs bénéfices respectifs peuvent être combinés sans qu'ils interfèrent entre eux.

Nous commençons, à la Section 4, par introduire le concept de la pile protocolaire TCP/IP qui nous servira de fil conducteur pour l'ensemble du travail.

Nous continuons, à la Section 5, par la définition du concept des systèmes MIMO.

A la Section 6, nous décrivons brièvement le protocole SCTP utile pour la compréhension des sections suivantes.

Le nouveau protocole MPTCP est décrit plus en détails à la Section 7.

La Section 8 présente l'état de l'art.

Nous expérimentons la combinaison d'un système MIMO et du protocole MPTCP à la Section 9.

Nous analysons les résultats obtenus lors de l'expérimentation à la Section 10.

Enfin, nous terminons ce mémoire par une conclusion à la Section 11.

# 4 La pile protocolaire TCP/IP

Dans ce chapitre, nous introduisons le concept de la pile protocolaire TCP/IP. Nous utilisons ce concept tout au long de cet ouvrage comme fil conducteur afin d'introduire les concepts des chapitres suivants.

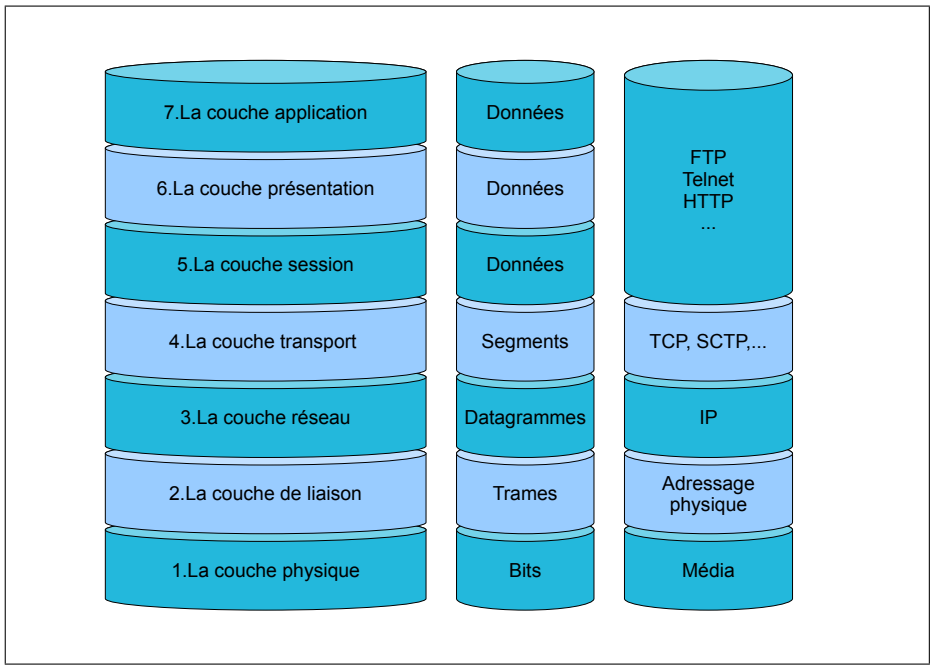


FIGURE 1 – Pile Protocolaire TCP/IP selon le modèle ISO

La pile protocolaire TCP/IP, dont le modèle est spécifié au chapitre 7 du document ISO 7498-1 [15], se définit en sept couches : physique, liaison, réseau, transport, session, présentation et application (Figure 1).

Afin d'illustrer les principaux concepts de ce projet, nous allons détailler un modèle *simplifié* de la pile protocolaire TCP/IP (Fig.2) couramment utilisé dans la littérature. Ce modèle réunit les trois couches supérieures du modèle ISO (application, présentation et session) en une seule couche, la couche *application*. Le modèle ISO de la pile protocolaire TCP/IP n'a pas la fonction de spécifier l'application.

Ce modèle en couches est conçu dans un principe d'isolation. Ce mémoire en explore certaines limites.

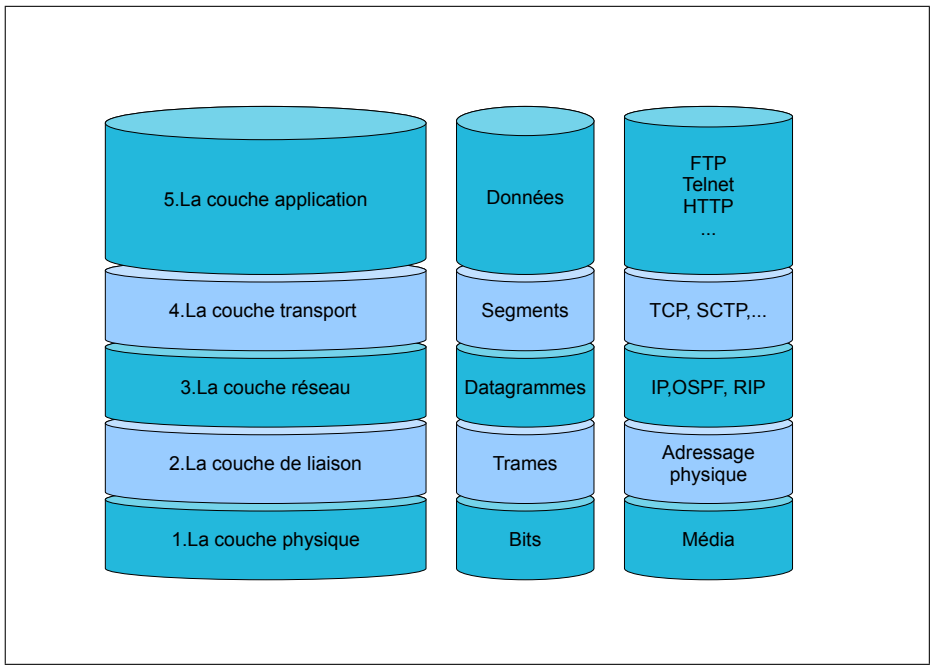


FIGURE 2 – Pile Protocolaire TCP/IP simplifiée

## 4.1 La couche physique

Elle permet le transfert des bits de données sur un canal de communication. Ces bits sont représentés physiquement sous forme analogique. La couche physique ignore la signification des bits qu'elle produit. Lors de leur création, elle fait intervenir des interfaces physiques sur le média physique (câble UTP, fibre optique, radio-électricité , etc). Elle reçoit des signaux qu'elle convertit en bits de données pour la couche de liaison et inversément.

## 4.2 La couche de liaison

Elle reçoit les bits générés par la couche physique et les groupes en paquets structurés appelés *trames*.

C'est la première couche qui fait intervenir des mécanismes de correction d'erreurs de transmission tels que la génération d'un checkum et le Contrôle de la Redondance Cyclique (CRC). Le contrôle d'erreurs est intégré dans la structure de la trame.

Dans les standards 802.x (Fig. 3), cette couche est subdivisée en deux sous-couches : la couche de contrôle de la liaison logique ou Link Layer Control (LLC) et la couche de contrôle d'accès au support ou Media Access Control.

La sous-couche LLC augmente la fiabilité de la couche MAC par un contrôle d'erreur et un contrôle de flux.

La sous-couche MAC joue le rôle d'interface entre la partie logicielle contrôlant la liaison d'un nœud (contrôle de la liaison logique) et la couche physique. Par conséquent, elle est typique au média physique utilisé.

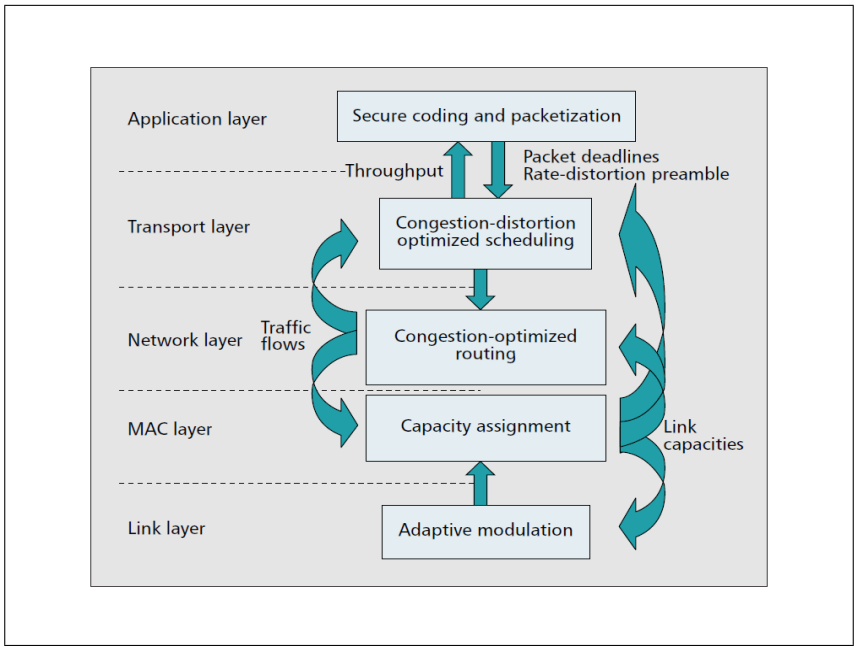


FIGURE 3 – Pile protocolaire TCP/IP simplifiée dans les standards 802.x [30]

## 4.3 La couche réseau

Elle assure *au mieux* l'acheminement des paquets. La couche réseau ignore le contenu des paquets. Elle fournit une méthode pour les mener à destination.

Internet Protocol [4] implémente cette couche.

Il autorise la définition d'une ou plusieurs adresses sur une ou plusieurs interfaces physiques. Cette fonctionnalité s'appelle le *multihoming*.

Par un mécanisme de *routage* sur base des adresses *sources* et *destinations*, IP peut connecter plusieurs réseaux différents. Pour établir cette connexion entre réseaux, il existe des nœuds particuliers appelés *routeurs*. Ces routeurs ont plusieurs interfaces physiques, chacune connectée à un réseau différent. Une table de routage permet au routeur de connaître le chemin vers lequel

il doit envoyer des paquets. Un routeur ne connaît jamais l'entièreté des réseaux connectés. Chaque nœud du réseau possède une table de routage.

Lors d'une transmission d'un paquet de données en dehors du réseau du nœud, celui-ci consulte sa table de routage pour connaître le routeur auquel il doit envoyer le paquet. Une fois le paquet reçu, le routeur vérifie dans sa propre table de routage vers quelle interface il doit le transmettre. Si le paquet est à destination d'un réseau se trouvant au bout d'une des interfaces, le chemin se termine. Dans le cas contraire, le paquet est envoyé au routeur suivant (next hop routing).

Il existe deux principaux types de routage : le routage statique et le routage dynamique. Pour le routage statique, tous les chemins sont figés dans la configuration des tables de routage. Le routage dynamique utilise différents protocoles comme le RIP (Routing Information Protocol) [18] ou OSPF (Open Shortest Path First) [21] permettant aux routeurs de communiquer entre eux et de s'échanger des informations de mise à jour des chemins du réseau sans avoir besoin de modifier manuellement les tables de routage.

## 4.4 La couche transport

Elle peut garantir l'intégrité des données en se servant des mécanismes de contrôle des couches inférieures.

### 4.4.1 Transmission Control Protocol

Ce protocole implémente la couche *transport*. TCP offre un transfert fiable de données avec un ordre strict de séquences mais parfois lent. En effet, si une séquence est perdue, cela introduit du retard par le délai de retransmission (head-of-line blocking [28]).

La nature orientée *flux de séquences de bytes* de TCP est un inconvénient pour les applications orientées *flux de séquences de messages* (*stream*) car elles doivent gérer elles-mêmes le marquage des messages, augmentant ainsi le temps de traitement.

TCP [23] gère le contrôle de flux et de congestion.

L'algorithme de gestion de la congestion le plus répandu est nomé *RENO*. Ce dernier utilise une fenêtre de congestion (W) représentant le nombre de paquets que TCP peut transmettre. Il possède un seuil (Wt) pour le mécanisme du *Slow-start*. Pour chaque acquittement de paquet, si W<Wt alors W=W+1 (*Slow-start phase*) , sinon W=W+1/W (*Congestion Avoidance Phase*).

Reno considère un début de congestion lorsqu'il reçoit trois ACK identiques pour le même paquet. Dans ce cas, Wt = W/2 et W = Wt (*Fast Retransmit*)

puis il se remet dans la *Congestion Avoidance Phase*. Lorsqu'une perte de paquet est détectée par un time out, alors Wt = W/2 et W=1 (*Slow Start phase*).

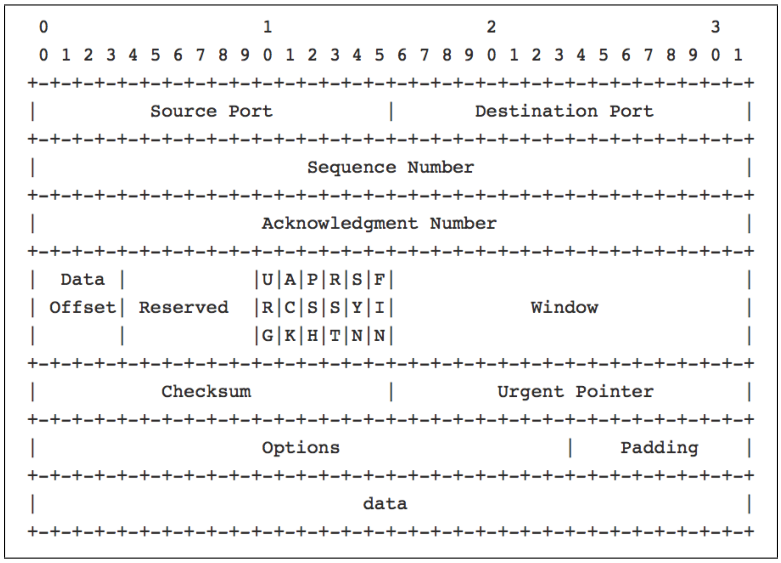


FIGURE 4 – TCP datagram header format

Au niveau sécurité, TCP est très vulnérable aux attaques DoS (Denial of Service) comme, par exemple, les attaques *SYN* [29]. La couche transport est responsable du multiplexage lors de plusieurs connexions simultanées par la même couche physique.

Les fonctionnalités limitées dans l'usage des sockets en TCP rendent très difficile l'implémentation de la haute disponibilité pour le transfert de données par l'utilisation de plusieurs chemins.

### 4.4.2 User Datagram Protocol

UDP est un autre protocole appartenant à la couche transport. Il est défini par le document *RFC 768*

Ce dernier permet une communication simple et rapide. L' intégrité du contenu d'un paquet UDP est assurée par un checksum.

Fonctionnant en mode *déconnecté*, il ne possède ni contrôle de flux, ni contrôle de congestion. Plus particulièrement, l'absence d'un mécanisme de retransmissions automatique des paquets perdus en fait un protocole réputé non fiable. Il est utilisé quand on privilégie la rapidité à la fiabilité, à l'inverse de TCP.
[22].

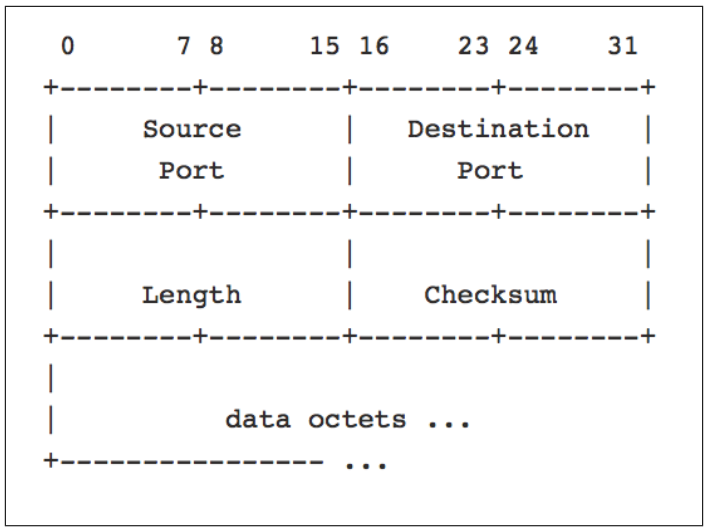


FIGURE 5 – UDP datagram header format

#### 4.4.3 La couche application

Elle gère la synchronisation des échanges, en déterminant celui qui peut émettre et à quel moment.

Un mécanisme de *transaction* permet de restaurer la dernière opération correcte en cas d'échec.

Elle permet l'ouverture et la fermeture de session entre chaque application coopérante.

Hormis les champs *techniques* nécessaires à leur fonction, les couches précédentes ignorent le contenu des données transmises.

La couche application assure un codage et un décodage cohérents de la partie *données* à destination de l'application.

Elle est considérée comme un point d'accès applicatif aux couches précédentes.

Elle prend en charge la représentation des données pour l'utilisateur et le contrôle de dialogue entre l'homme et la machine.

# 5 Les systèmes MIMO

C'est au niveau de la couche physique de notre pile protocolaire TCP/IP simplifiée que travaillent les systèmes MIMO (Multiple Inputs Multiple Outputs). Un système MIMO est un système d'émission/réception intelligent combinant plusieurs techniques : la diversité [27], le multiplexage spatial [27] et le *beamforming*.

Un système MIMO permet d'augmenter la capacité d'un système. Il améliore sa fiabilité en diminuant le taux d'erreurs bit (Bit Error Rate).

L'élément frontal d'un système radio a une complexité, une taille et un prix directement proportionnels au nombre d'antennes. Dans [27], les auteurs proposent de diminuer ce coût en utilisant les avantages des systèmes MIMO.

## 5.1 La diversité

Evoquée à la Section 8.1, c'est une méthode améliorant la fiabilité d'un signal de données. La diversité se base sur le phénomène suivant : si un signal est transmis sur plusieurs canaux de communication indépendants de caractéristiques différentes, il ne se dégrade pas de la même manière sur chaque canal. Plusieurs copies d'un signal sont transmises ou reçues et peuvent donc être combinées au récepteur.

Alternativement, on peut ajouter un système de correction redondant d'erreurs en amont, et transmettre plusieurs morceaux d'un message par différents canaux. Le bénéfice obtenu par la propagation multiple de la diversité est appelé *l'ordre de la diversité* (diversity order). L'ordre de la diversité est le nombre de canaux parallèles utilisés.

Les principales catégories pour ces techniques sont : la diversité dans le temps, la diversité de fréquence, la diversité spatiale, la diversité de polarisation, la diversité multi-utilisateurs et la diversité coopérative.

### 5.1.1 La diversité dans le temps

Plusieurs signaux identiques sont transmis à des moments différents. Le *Space Time Coding* [1] améliore la robustesse de la transmission sans utiliser un code d'erreurs redondant. Un seul flux de données est reproduit et émis par plusieurs antennes de telle sorte que les séquences transmises par chaque antenne soient orthogonales.

### 5.1.2 La diversité de fréquences

Un signal est émis en utilisant plusieurs canaux de fréquence ou en couvrant un large spectre qui est affecté par une atténuation de fréquences sélective.

### 5.1.3 La diversité spatiale

Le même signal est transmis par plusieurs chemins de propagation différents. Pour une transmission filaire, les données sont transmises sur plusieurs fils. Pour une transmission sans fil, on implémentera la diversité d'antenne en utilisant plusieurs antennes émettrices (diversité d'émission) et/ou plusieurs antennes de réception (diversité de réception).

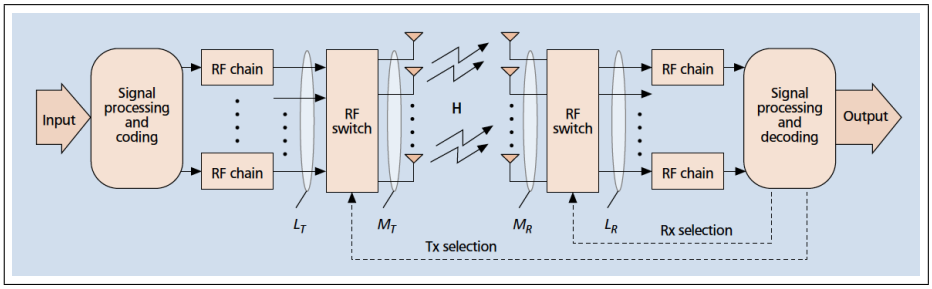


FIGURE 6 – Sélection d'antenne pour la transmission par un système MIMO

Les principales techniques de la diversité de réception et d'émission sont :

- la sélection des antennes sur base de celles qui ont le meilleur rapport signal/bruit (SNR)
- l'évaluation des signaux en choisissant les plus forts et rejetant les plus faibles (Maximum Ratio Combining)
- l'addition des signaux après une remise en phase de ceux-ci (Equal Gain Combining).

Si les antennes sont éloignées, on parlera de macro-diversité et de diversité de site. Si les antennes sont proches, à une distance de l'ordre d'une longueur d'onde, on parlera plutôt de micro-diversité.

L'une des limitations les plus importantes se pose chaque fois que la bande passante du système est plus grande que la bande de cohérence du canal. La réponse différente du canal à différentes fréquences implique que pour chaque bande, une autre sélection de l'antenne soit optimale. Par conséquent, chaque fois que le canal de fréquence est très sélectif avec de nombreuses bandes de fréquences non corrélées, la sélection d'antenne peut ne pas être réalisable ou pertinente. Cependant, d'une manière modérée, avec les canaux sélectifs en fréquence, la sélection d'antenne permet d'obtenir des gains significatifs (Figure 7)

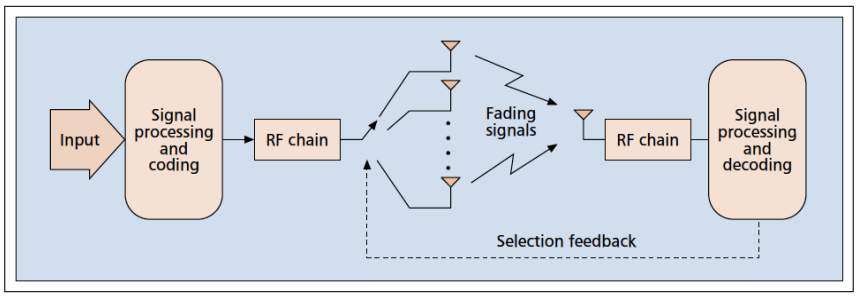


FIGURE 7 – Transmission par sélection d'antenne [27]

Les commutateurs *Radio Fréquence*, en anglais *switches RF*, sont actuellement loins d'être optimaux ; cela peut être compensé par certains avantages de la technique de sélection d'antenne. La plus importante lacune des switches RF est l'atténuation du signal lors du transfert. Il faut compenser par une augmentation de puissance pour l'amplificateur de l'étage de sortie de l'émetteur et une plus grande sensibilité, à faible bruit, pour l'amplificateur du récepteur.

L'analyse et la conception du code de sélection d'antenne nécessitent encore plus de recherches.

Le problème de l'optimisation de la transmission/réception reste important et ouvert.

L'évaluation de la performance des algorithmes de la sélection d'antenne, lorsqu'il manque des informations au niveau du récepteur, est un problème important relativement inexploré.

La combinaison de sélection d'antenne par l'espace-temps (Space Time Coding) a été abordée par plusieurs chercheurs, mais il reste encore beaucoup de travail sur le sujet.

### 5.1.4 La diversité de polarisation d'antenne

On utilise des antennes à double polarisation (+45 °, -45 °). La polarisation d'un signal est caractérisée par l'orientation de son champ électrique. Il est parallèle à la surface terrestre (polarisation linéaire horizontale) ou perpendiculaire (polarisation linéaire verticale). Un téléphone célulaire n'est jamais orienté complètement verticalement, ni complètement horizontalement. En utilisant une polarisation ni trop verticale, ni trop horizontale, on augmente la possibilité de recevoir un niveau de signal correct.

### 5.1.5 La diversité multi-utilisateurs

Cette technique ordonnance l'émission ou la réception sur base de la qualité du canal d'un utilisateur. L'ordonnancement est effectué en fonction de l'information sur la qualité des canaux disponibles entre l'émetteur et le récepteur.

La technique nécessite de connaître, de manière continue, l'information sur la qualité des canaux entre les émetteurs et les récepteurs. Il est à remarquer que ce n'est envisageable que dans un réseau sans fil, là où le canal fluctue rapidement.

### 5.1.6 La diversité coopérative

Plusieurs points d'accès relayent le signal pour le compte d'un émetteur.

### 5.1.7 La combinaison de diversités

Plusieurs techniques de diversité peuvent être combinées afin de tirer parti des avantages de chacune d'elles.

Par exemple, il est possible de combiner les avantages de la diversité d'antennes avec la diversité multi-utilisateur pour optimiser ou améliorer la capacité de l'ensemble des canaux utilisés.

## 5.2 Le multiplexage spatial (Spatial Multiplexing)

SM permet de faire passer plusieurs communications différentes, aussi appelées *streams*, sur des canaux qui apparaissent indépendants.

A la réception, une analyse des valeurs propres de la matrice de corrélation permet, par excitation des canaux avec les vecteurs propres correspondants, d'identifier les canaux virtuels ayant le meilleur gain pour en extraire le signal utile. Le gain obtenu s'appelle le *Multiplexing Gain*.

## 5.3 Le beamforming

Le principe est de créer des interférences *utiles* dirigées vers les clients récepteurs. A la réception, elles amplifient le signal résultant de la somme des signaux émis.

Il est à noter que ce principe apporte un bénéfice, également, aux systèmes IEEE 802.11a/g.

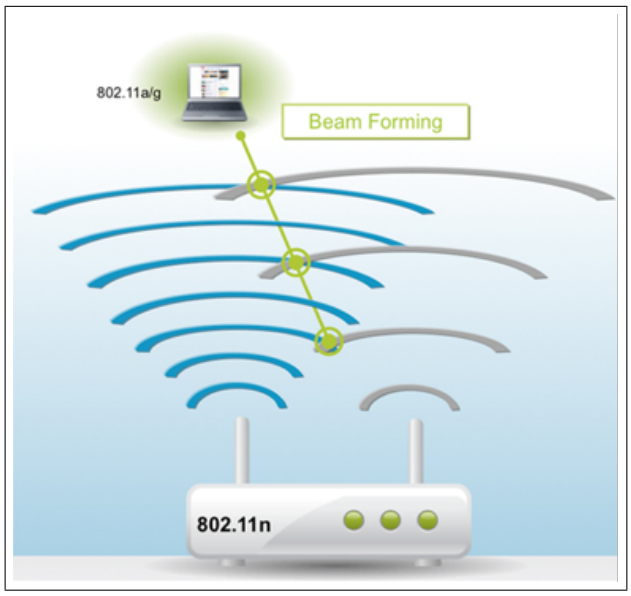


FIGURE 8 – Principe du beamforming [7]

## 5.4 Implémentation des systèmes MIMO

Un exemple d'application de la technologie MIMO est le standard de transmission sans fil IEEE 802.11n, ratifié en 2009. Il est le successeur des normes IEEE 802.11a, b et g. Ce nouveau standard s'appuie sur la technologie MIMO combinée avec l'*Orthogonal Frequency Division Multiplexing* (MIMO-OFDM). *Frequency Division Multiplexing* est une technologie qui permet de transmettre plusieurs signaux, simultanément, par un seul chemin (filaire ou non filaire). Chaque signal voyage sur sa propre porteuse. L'OFDM reprend ce principe mais il distribue les signaux sur un grand nombre de porteuses. Celles-ci sont espacées à des fréquences précises. Cette espacement précis produit l'orthogonalité. OFDM apporte de la robustesse face aux interférences. Il permet d'atteindre un taux de transfert élevé en augmentant la capacité du canal de communication.

Le débit est considérablement amélioré par l'agrégation de paquets (Aggregate MAC Protocol Data Unit) [31].

Toutes ces technologies combinées par ce standard offrent une meilleure couverture même pour les clients non IEEE 802.11n.

Ce standard est rétro compatible vers les standards précédents. Le taux de transfert peut atteindre théoriquement 300Mbits/s contre 54Mbits/s pour IEEE 802.11g.

Les normes IEEE 802.11a, b, g ont une largeur de bande de 20 MHz. IEEE 802.11n peut, optionnellement travailler sur une bande passante de 40 MHz. La bande passante du canal étant doublée, le nombre de sous-porteuses de données est un peu plus que doublé. Par la combinaison de quatre canaux avec la technologie MIMO, on peut obtenir un débit du canal total de 600 Mbits/s. Cette option est controversée car non applicable dans tous les environnements.

Un autre exemple d'application de la technologie MIMO est l'*Universal Mobile Telecommunications System*. UMTS est l'une des technologies (3G) de téléphonie mobile européenne.

UMTS Release 7 [24], aussi appelé *Evolved High-Speed Packet Access* ou HSPA+, incorpore la technologie MIMO [2]. HSPA était déjà inclus dans UMTS release 5 et 6. HSPA est une combinaison de deux protocoles, le protocole *High Speed Downlink Packet Access* [17] et le protocole *High Speed Uplink Packet Access* [10].

Ces deux protocoles sont issus de l'amélioration du protocole *Wideband Code Division Multiple Access* [25].

# 6 Stream Control Transmission Protocol (SCTP)

Une première extension du protocole TCP, pertinente pour ce mémoire, est le Stream Control Transmission Protocol. SCTP est défini par le document *RFC 4960* [32].

Nativement, le SCTP permet le *multihoming*, c'est à dire la possibilité pour un système d'établir plusieurs connexions en utilisant une adresse IP pour chacune d'elles.

Chaque adresse IP est vue comme un chemin (multi-path).

Certaines applications ont besoin d'un transfert fiable mais un ordre partiel de séquences serait suffisant.

Travaillant au niveau de la couche *transport* de la pile protocolaire TCP/IP, SCTP se présente comme une alternative et permet de lever certaines limitations de TCP. La Figure 9 présente un comparatif entre TCP et SCTP.

| Protocol | TCP | SCTP |
|---|---|---|
| Setup messages | Three-way handshake | Four-way handshake |
| Shutdown messages | Four-way handshake | Three-way handshake |
| Half-open support | Supported | Not supported |
| Ordered delivery | Strict ordered | Ordered within a stream |
| Unordered delivery | Not supported | Supported |
| Message boundary | No boundary<br>Stream-oriented | Boundary preserved<br>Message-oriented |
| Multihoming | Not supported | Supported |
| SACK support | Optional | Mandatory |
| Keep-alive heartbeat | Optional | Mandatory |
| Heartbeat interval | ≥ Two hours | 30 seconds by default |

FIGURE 9 – Comparatif entre TCP et SCTP [8]

SCTP est un protocole point-à-point (unicast) mais avec des propriétés de multi-diffusion (multicast) permettant du multi-path TCP.

Le chemin initial est appelé *primaire*. Un chemin appelé *secondaire* sert de chemin de secours en cas de perte du chemin *primaire*.

Ces liens sont de type *actif/passif* (un seul lien fonctionne à la fois, l'un ou l'autre mais pas les deux en même temps).

Au niveau de la fiabilité, SCTP utilise les sommes de contrôle et l'acquittement sélectif des paquets (SACK) [20]. SACK est optionnel en TCP et est obligatoire en SCTP [8]. Cette fonctionnalité permet de limiter la ré-émission de tout un flux lors de la perte de paquets. Seuls les paquets perdus sont retransmis.

SCTP est orienté *flux de séquences de messages* (*stream*) par rapport à TCP qui est plutôt orienté *flux de séquences de bytes*. Cette propriété permet de se libérer d'un ordre strict de séquences de données.

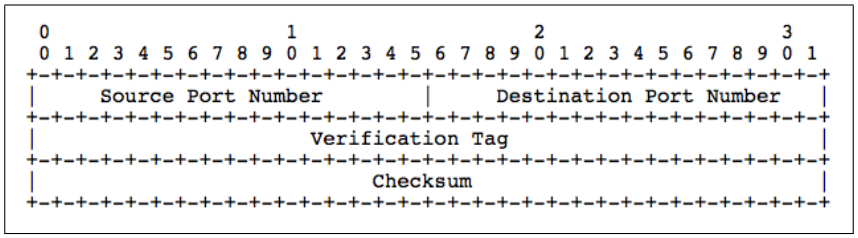


FIGURE 10 – SCTP datagram header format

SCTP est conçu pour coopérer avec TCP au niveau de la bande passante. Il utilise les mêmes algorithmes de régulation de trafic.

SCTP permet de gérer plusieurs flux simultanés à l'intérieur d'un même flux. Il peut y avoir un flux de données et un flux de contrôle en même temps.

Afin de contrer les attaques de type *DoS* (Denial of Service), le préambule de la communication de SCTP se fait en quatre étapes, au lieu de trois en TCP. Un mécanisme de *cookies* permet de placer les informations au niveau du réseau et du client plutôt qu'en mémoire. Au niveau de la sécurité, ces deux propriétés empêchent bon nombre d'attaques de type DoS. SCTP combine fiabilité et rapidité de transmission.

La première application de SCTP est la téléphonie sur IP.

# 7 Le Multi Path-Transmission Control Protocol (MPTCP)

Nouvelle extension du protocole TCP, son architecture est définie par le document RFC 6182 [9].

Dans [34], les auteurs nous proposent une nouvelle approche permettant l'utilisation de plusieurs chemins actifs simultanément. Les objectifs fonctionnels de MPTCP sont de garantir une stabilité de transfert sur les différents chemins (résilience), et d'augmenter la bande passante par l'agrégation de celle de chaque chemin.

MPTCP se veut compatible avec les applications et l'infrastructure existantes. S'il rencontre des difficultés, il doit être rétro-compatible avec TCP pour assurer une complète transparence vis-à-vis des *middle boxes* (Network Address Translation, FireWall, Proxy, etc).

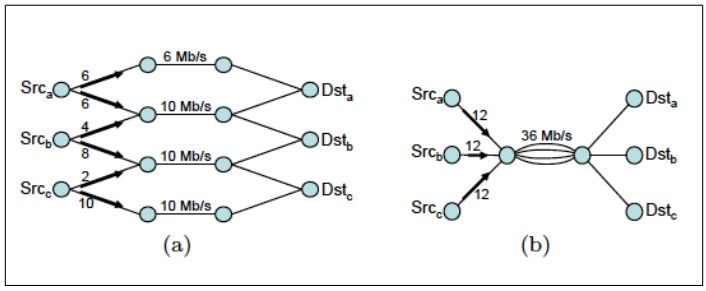


FIGURE 11 – Principe du *Ressource Pooling*

Basé sur le concept du *Resource Pooling* (Figure 11), chaque chemin fait partie d'un pool apparaissant comme une seule voie de communication pour la couche application (*flux principal*). Ce concept améliore la robustesse du protocole devant la perte d'un chemin. Il permet également d'accroître l'agilité dans la gestion de la charge du trafic et offre une optimisation de l'utilisation de chaque chemin.

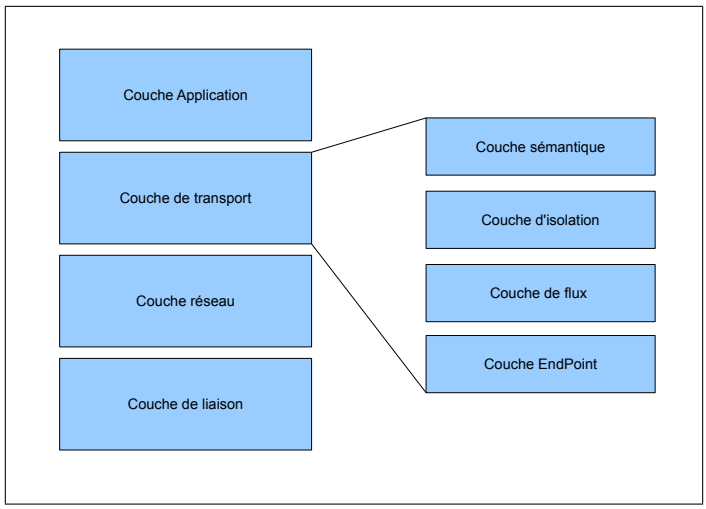


FIGURE 12 – Modèle MPTCP

Comme l'illustre la Figure 12, on peut modéliser la couche transport en 4 sous-couches organisées en deux groupes [19]. Le groupe *orienté application* se compose d'une sous-couche sémantique permettant aux canaux de communication de s'abstraire de l'application et d'une sous-couche d'isolation fournissant une protection de bout en bout et garantissant la fiabilité. Le groupe *orienté réseau* se compose d'une sous-couche de flux fournissant les mécanismes de contrôle de la congestion et de la performance. Enfin, une sous-couche *Endpoint* assure la gestion des services et d'identification comme par exemple le numéro de port.

MPTCP s'insère entre la couche application et la couche transport (Figure 13). Chaque chemin est une session TCP indépendante. MPTCP gère la sélection des interfaces, des chemins et la réorganisation des paquets distribués sur chaque chemin. MPTCP permet de transporter des données par plusieurs chemins entre un émetteur et un receveur mais ne garantit pas que ces deux chemins ne se rencontreront pas à un moment donné pour provoquer un goulot d'étranglement. Comme chaque chemin est totalement *compatible TCP*, chaque flux MPTCP peut passer en IPv4 ou IPv6.

L'application ouvre un dialogue avec la sous-couche sémantique. Cette connexion représente le flux principal de MPTCP. Le préambule de communication entre deux hôtes se base sur le préambule *trois voies* de TCP (*three-way handshake*) et a été enrichi de l'option $MP\_CAPABLE$. Cette option est activée par l'initiateur ($HOST\ A$) de la communication dans la séquence $SYN$. Si le message $SYN{+}ACK$ ne contient pas l'option, le reste du préambule se déroulera comme pour une communication TCP standard. Si le receveur ($HOST\ B$) est capable de comprendre cette option, il enverra un message $SYN{+}ACK$ contenant l'op-

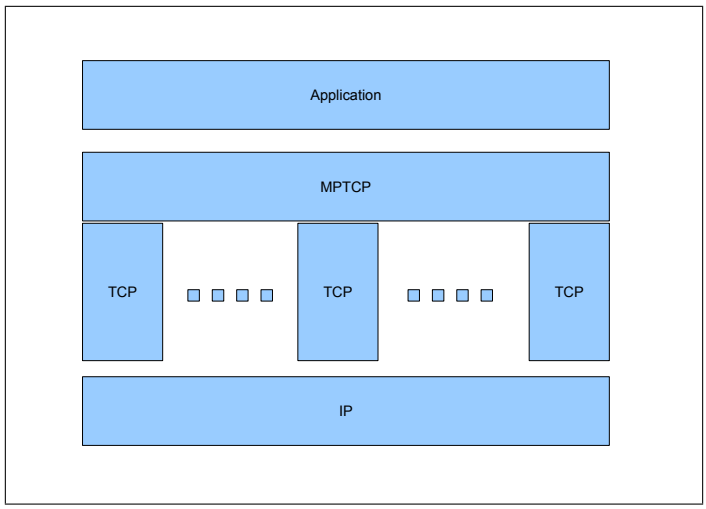


FIGURE 13 – Pile protocolaire MPTCP

tion *MP_ CAPABLE* et un jeton d'identification (*token B*). Le *HOST A* envoie alors un message *ACK* avec de nouveau l'option *MP_ CAPABLE* activée et son propre jeton d'identification (*token A*). Pour terminer le préambule, le *HOST B* envoie enfin un message ACK, en faisant un préambule *quatre voies* (*four-way handshake*). Les autres interfaces du *HOST A* initieront une connexion sur la première interface de contact du *HOST B* en s'identifiant via le *token B* reçu lors du préambule. *HOST B* enverra l'adresse de chacune de ses interfaces à *HOST A*. Chaque chemin de *HOST A* établira une connexion de la même manière sur chaque adresse de *HOST B* se trouvant dans le même réseau. La fin d'une connexion se fait également en quatre voies.

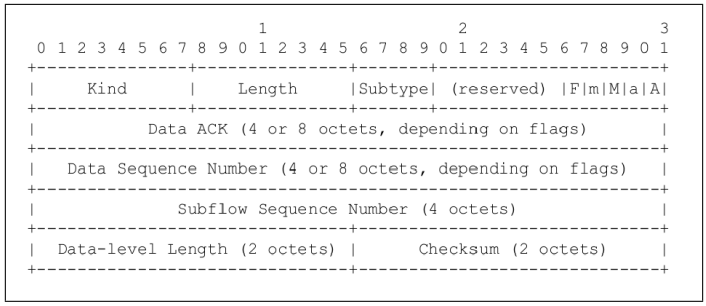


FIGURE 14 – Structure d'un paquet MPTCP

Comme on peut le voir à la Figure 14, la numérotation du séquencement des paquets MPTCP est adressée sur 64-bit (*Data Sequence Number*) tandis que chaque sous-flux possède son propre espace d'adressage sur 32-bit (*Subflow Sequence Number*) pour garantir la compatibilité TCP. Une correspondance entre

le DSN et le SSN est maintenue par MPTCP (*Data Sequence Mapping*). MPTCP utilise l'algorithme du SACK pour les sous-flux et celui du *Cumulative ACK* pour le flux principal (Data connexion level).

La gestion de la congestion standard TCP n'est pas efficiente pour une solution *multi-chemins*. En effet, dans [35], les auteurs proposent un algorithme plus adapté à MPTCP (*coupled congestion control*). Cet algorithme permet d'éviter aux sous-flux de s'approprier la bande passante (*fairness*) et de sélectionner les chemins les moins congestionnés.

Pour indiquer une fin de connexion, TCP emploie un message *FIN* dans le meilleur des cas. En cas de problème, il enverra un message *RST*. MPTCP doit pouvoir arrêter proprement chaque sous-flux dès que les messages *DATA ACK's* attendus par le flux principal sont arrivés. Autrement dit, MPTCP distingue la fin d'une connexion du flux principal et la fin d'une connexion de ses sous-flux. Pour un message *RST*, il terminera le sous-flux sur lequel il a été reçu. Lorsqu'un hôte MPTCP veut terminer sa connexion, il envoie un message *DATA FIN* dès que l'application ferme son socket. Pour terminer proprement, l'émetteur du message *DATA FIN* attendra de recevoir l'*ACK* de ce message avant d'envoyer un *FIN* sur chacun de ses sous-flux. Il peut envoyer également un message *DATA FIN* sur chaque sous-flux. Enfin, MPTCP possède un message *REMOVE_ADDR* principalement créé pour supporter la mobilité.

Pour qu'un nouveau protocole suscite l'adhésion, il doit offrir de nouvelles opportunités et avoir un impact minimal sur l'infrastructure existante, voire pas du tout. Dans [6], les auteurs passent en revue les principaux mécanismes sur lesquels s'appuie MPTCP avant de passer à son implémentation afin d'en vérifier l'efficacité.

Comme expliqué plus haut, MPTCP utilise le premier échange entre deux hôtes pour déterminer, via l'option MP_CAPABLE, si MPTCP est activé de part et d'autre. Les auteurs ont réalisé une étude [11] démontrant que sur l'ensemble des tests effectués, seulement 6 % des chemins testés retirent l'option du message *SYN*, augmentant à 14 %, pour du trafic exclusivement *HTTP* (port 80). Une étude séparée [3] a permis de démontrer que ce mécanisme ne posait pas de problème.

Pour l'ouverture des chemins, l'idéal serait d'éviter de transmettre un message *SYN* mais il est rare que les *middle boxes* laissent passer des données non précédées par une négociation. L'ajout de nouveaux sous-flux pose plusieurs difficultés. La première est de ne pas pouvoir utiliser, comme identifiant, le traditionnel 5-tuple (PROTOCOL, local IP-addr, local TCP-port, foreign IP-addr, foreign TCP-port) car les *NAT's* les substitueraient par nature. La deuxième difficulté est qu'MPTCP doit pouvoir résister à un éventuel attaquant voulant lui subtiliser un sous-flux. Pour ce faire, dès que la connexion est établie, le serveur insère une clé aléatoire de 64-bit dans l'option MP_CAPABLE. Elle sera

utilisée pour valider l'authenticité des nouveaux sous-flux du pool. Dès lors, pour ouvrir un nouveau sous-flux, MPTCP fera un nouvel échange *SYN* utilisant les adresses ou ports qu'il souhaite utiliser. Une nouvelle option *MP_JOIN* est ajoutée dans les messages SYN et *SYNACK* transportant un token généré à partir de la clé 64-bit. Par un des sous-flux, le serveur informe le client qu'il possède une adresse supplémentaire, via l'option *ADD_ADDR*.

Dans un monde sans *middle boxes*, nous pourrions simplement utiliser le séquencement de paquets de TCP. Les auteurs ont observé que 10 % du trafic internet (18% sur le port 80) voit son séquencement *optimisé* par les *middle boxes* qui suppriment, au passage, les nouvelles options. Cette contrainte empêche MPTCP d'ouvrir un nouveau chemin ayant le même espace d'adressage que le chemin principal.

La répartition des paquets sur plusieurs chemins introduit des intervalles dans la numérotation des paquets par chemin. Ce comportement est détecté par certains *middle boxes* qui bloquent ce trafic empêchant ainsi l'activation de MPTCP (5 % pour le trafic standard et 11% sur le port 80).

Plus surprenant, 26 % des chemins observés (33% sur le port 80) n'ont pu être établis car les *middle boxes* ont supprimé ou corrigé un message *ACK*. Les auteurs en concluent qu'il faut utiliser un espace d'adressage séparé par sous-flux dans l'espace d'adressage global du flux de donnée (Data Sequence Space).

Afin d'éviter une situation d'inter-blocage, chaque connexion possède un seul pool de tampon de réception (*receive buffer*), partagé entre tous les sous-flux. Dès lors, la fenêtre de réception (*receive window*) indique le numéro de séquence maximale du flux principal de données (*Data Sequence Number*) et plus celui du sous-flux. Le problème est que chaque sous-flux doit garder sa propre numérotation de séquences.

Pour garantir le mécanisme du *Cumulative acknowledgement*, le flux principal de données maintient un tableau (*DSM*) reprenant quelle séquence du flux principal a été envoyée sur quel sous-flux. La solution a été d'introduire un champ (*data acknowledgement*) dans la structure de MPTCP. Cette information est contenue dans le champ *options* d'un message *ACK* TCP. L'information de *mapping* contient l'offset de la séquence initiale du sous-flux et sa longueur, au lieu du *mapping* de la séquence initiale MPTCP. Cela permet d'éviter que les cartes ethernet et les *middle boxes*, optimisant la segmentation en re-segmentant le trafic et en recopiant la zone *options*, ne perdent le *mapping*.

L'émetteur libère ses segments uniquement s'il reçoit les accusés de réception du flux principal (*DATA ACK*). Si l'accusé de réception d'un sous-flux est arrivé, ses données sont mémorisées jusqu'à ce que le *DATA ACK* arrive. S'il y a une perte d'un *DATA ACK*, l'émetteur retransmettra cette donnée après un time-out.

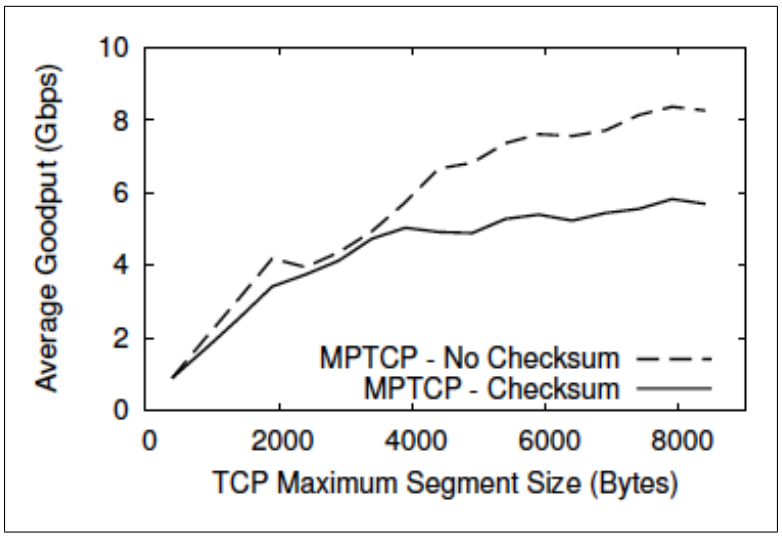


FIGURE 15 – Impact du calcul checksum sur la performance [6]

Les *middle boxes* et plus particulièrement les *NAT*, ont parfois une fonctionnalité de passerelle applicative pour certains protocoles. Par exemple, pour le protocole *FTP* (*ASCII mode*), les adresses IP et les ports du canal de contrôle sont réécrits dans la partie des informations utiles du paquet (*payload*). En modifiant le *payload*, le *NAT* rend erronée l'information contenue dans le *DSM*. Pour détecter ce genre de situation, le *DSM* contient son propre checksum (complément à 1 sur 16-bit comme pour TCP). Si le checksum est erroné sur un sous-flux, MPTCP ferme le sous-flux. S'il n'y a qu'un seul sous-flux, MPTCP bascule en TCP.

Le calcul d'un checksum est consommateur de temps (Figure 15), surtout si c'est le programme qui doit le calculer. Pour ne pas devoir calculer à la fois celui du *payload* et celui du *DSM*, MPCTP calcule le checksum du *payload* incluant le *DSM*. Il recopie ensuite ce checksum dans le *DSM*.

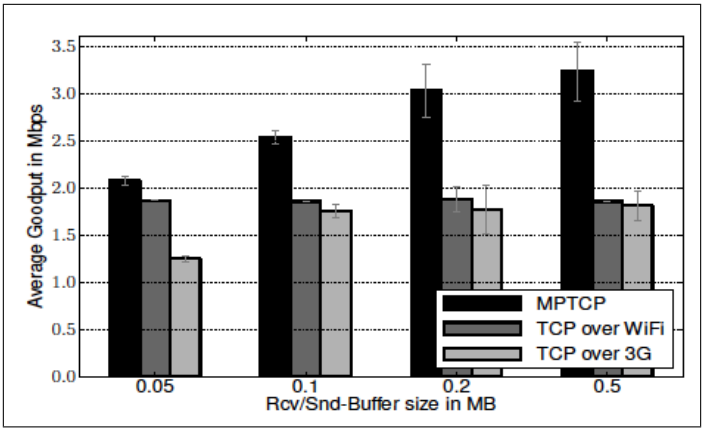


FIGURE 16 – Impact de la taille du rcv/snd buffer sur la performance [6]

Comme MPTCP utilise le *Cumulative Acknowledgement*, il doit avoir une taille de *receive buffer* raisonnable pour rester attractif sur du matériel mobile où la taille de la mémoire est limitée. La difficulté se présente lorsque les sous-flux n'ont pas le même $RTT$ (ex : un premier sous-flux wifi avec un $RTT$ +/- 20 ms et un deuxième sous-flux 3G +/- avec un $RTT$ de 150 ms). Il faut plus de place pour accumuler les paquets qui ne sont pas encore acquittés.

Pour optimiser cette mémoire, MPTCP possède plusieurs mécanismes. Lorsque la fenêtre de réception est saturée mais qu'il y a de l'opportunité (*Opportunistic retransmission*) avec la fenêtre de congestion, MPTCP tente de soulager la fenêtre de réception en émettant de nouveau sur le sous-flux le plus rapide, des paquets en attente d'acquittement. Ce mécanisme s'active uniquement s'il n'y a plus de place disponible dans le *receive buffer*. S'il est sous dimensionné, cela occasionnera un gaspillage de la capacité du canal de transmission. Pour éviter une retransmission coûteuse, MPTCP va diminuer la fenêtre de réception du sous-flux le plus lent (*Penalizing slow subflows*). Les implémentations modernes de TCP, augmentent automatiquement la taille du tampon de réception si cela est nécessaire. Si MPTCP suit cette logique, il allouera inutilement de la mémoire en fonction du sous-flux ayant le plus mauvais $RTT$. Une solution est de limiter la taille de la fenêtre de congestion si le $RTT$ du sous-flux le plus lent est le double du plus petit $RTT$ de base du sous-flux le plus rapide (*Buffer autotuning with capping*).

Les auteurs constatent qu'il y a 20 ans que la *RFC1323* [16] a été publiée mais que bon nombre de *middle boxes* bloquent toujours les extensions de TCP normalisées. Pour être largement déployable, les extensions de TCP doivent obligatoirement pouvoir détecter une erreur provoquée par une mauvaise interprétation du *middle boxe* et basculer en TCP basic pour garantir un transfert de données aussi fiable que TCP .

Enfin, les auteurs concluent qu'il est possible de déployer leur implémentation de MPCTP mais en tenant compte du comportement des *middle boxes* identifiés à ce jour. Leur implémentation a permis de démontrer que les différents algorithmes mis en place sont efficients. Ce travail met en évidence que les *middle boxes* augmentent la complexité de l'Internet et pénalisent son évolution. Il faudrait revisiter l'architecture de l'Internet pour reconnaître explicitement leur rôle mais le challenge principal est d'avoir une solution déployable sur l'Internet d'aujourd'hui.

# 8 Etat de l'art

Dans ce chapitre nous nous intéressons à la notion de chemins créés à partir de la couche physique et de la couche transport.

A la section 8.1, nous mettons en évidence l'intérêt d'une interaction entre la couche physique et la couche transport de la pile protocolaire TCP/IP afin d'optimiser l'utilisation de leurs mécanismes respectifs de résilience et d'amélioration de la bande passante.

Nous examinerons l'utilisation de plusieurs interfaces réseau simultanément à la section 8.2.

Dès lors, nous pourrons évoquer, à la section 8.3, l'opportunité d'améliorer la bande passante ou la robustesse d'une transmission par l'utilisation de plusieurs chemins, au niveau de la couche transport.

Enfin, nous cloturerons ce chapitre par une définition des problèmes à la section 8.4.

## 8.1 Interaction entre la couche physique et la couche transport de la pile protocolaire TCP/IP

Dans [26], les auteurs expliquent que la combinaison des différentes technologies MIMO n'est pas toujours efficiente car pour un gain de robustesse maximal, on obtient un gain de capacité nul et vice-versa. Il faut donc faire un compromis entre les deux pour garder un système à la fois robuste et performant.

Depuis l'avènement des réseaux sans fil, le protocole TCP n'a pas été revisité pour être adapté à cet autre contexte de la couche physique. Les auteurs nous rappellent l'algorithme de gestion de la congestion le plus répandu pour TCP, à savoir, *Reno*.

Dans [36], les auteurs constatent que TCP a été conçu pour fonctionner sur des réseaux câblés et qu'il perd de sa performance sur les réseaux sans fil.

TCP se méprend sur la situation du contexte de la couche physique. Les mécanismes de gestion de la congestion sont démarrés trop tôt et provoquent des retransmissions inutiles.

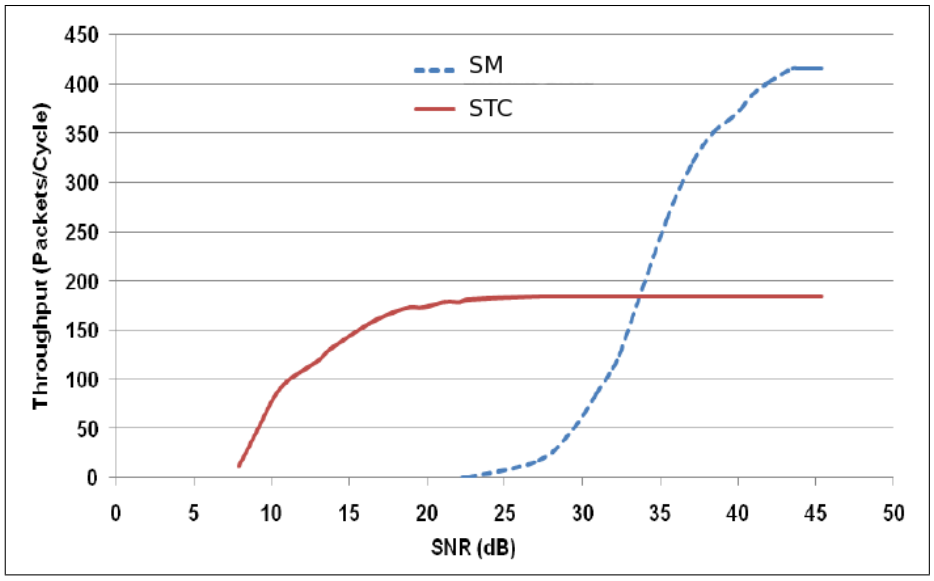


FIGURE 17 – Comparaison entre STC et SM [36]

Par simulation, les auteurs de [26] remarquent qu'au delà d'un certain SNR (Figure 17), le Space Time Code, évoqué à la section 5.1.1, n'apporte plus qu'une augmentation négligeable de la bande passante.

A partir d'un SNR un peu plus élevé, le Spatial Multiplexing, évoqué à la section 5.2, continue d'améliorer significativement le débit. A faible SNR, le gain de robustesse est élevé tandis que le gain de capacité est élevé à haut SNR.

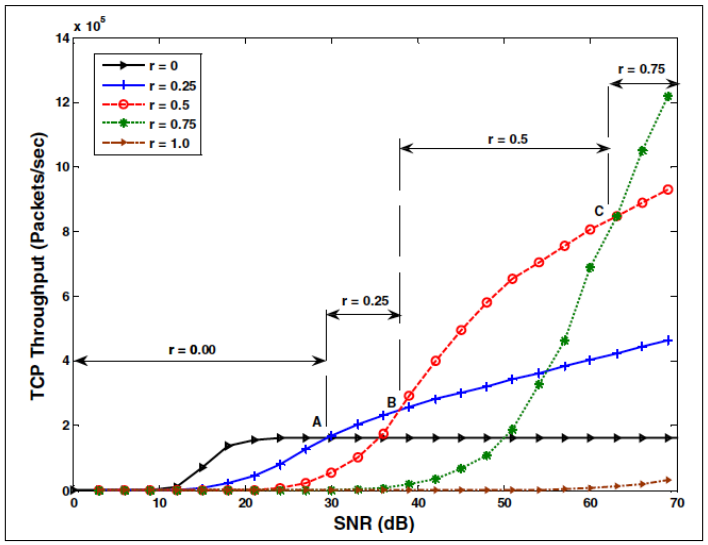


FIGURE 18 – Délimitation des zones par SNR [36]

Dans leur évaluation, les auteurs ont identifié quatre zones délimitées par trois SNR (Figure 18).

Ils proposent une solution dite *Rate-Switching* basée sur une information embarquée dans les messages ACK. Le receveur, connaissant son contexte radio, peut déterminer le SNR. Il indiquerait au transmetteur TCP cette information dans un message ACK. Ce dernier pourrait avertir sa couche physique pour sélectionner le meilleur mécanisme pour une transmission TCP optimale.

A taux fixe, une variable binaire indiquerait d'utiliser STC à la valeur zéro et SM à la valeur 1. A taux fluctuant, il faudrait que la variable contienne ce taux et applique STC ou SM en fonction.

En guise de conclusion, les auteurs indiquent que ce mécanisme de collaboration entre la couche transport et la couche physique permettrait à TCP d'obtenir une transmission optimale quel que soit le SNR.

## 8.2 Activation de TCP sur de multiples interfaces (multi-homing, multi-path)

La possibilité d'emprunter plusieurs chemins pour une ou plusieurs adresses *source* se nomme *Multi-Path TCP*. Le *multihoming* est la possibilité d'obtenir un chemin distinct pour chaque interface de communication. Ces deux propriétés permettent d'augmenter la fiabilité d'une transmission de données en établissant plusieurs chemins.

Dans le monde du stockage de données de type *Storage Area Network* (SAN), le *multi-path* est une fonction essentielle pour la haute disponibilité (ESCSI, Fiber Channel). Le MPTCP peut s'utiliser avec une seule adresse source et destination se reposant sur les protocoles de routages dynamiques qui gèreront les différents chemins en fonction de leur fiabilité.

Dans [5], les auteurs s'intéressent à la possibilité d'utiliser simultanément plusieurs liaisons sans fil de technologies différentes. Ceci permettrait de lever les limitations des technologies sans fil existantes en offrant de nouveaux services tels que : l'agrégation de bande passante, la suppression de la coupure lors du passage d'un réseau sans fil à un autre (*handover*), l'augmentation de la fiabilité de la transmission, le partage des ressources entre machines et, pour terminer, la séparation entre les données et leur contrôle sur des canaux séparés. Les auteurs ont choisi l'agrégation de bande passante à titre d'illustration.

Le problème majeur est le réordonnancement de paquets. Celui-ci peut provoquer des retards dans le calcul des temps de réponse ainsi que des "time-out" erronés occasionnant des retransmissions, pénalisant la vitesse de transmission de l'application.

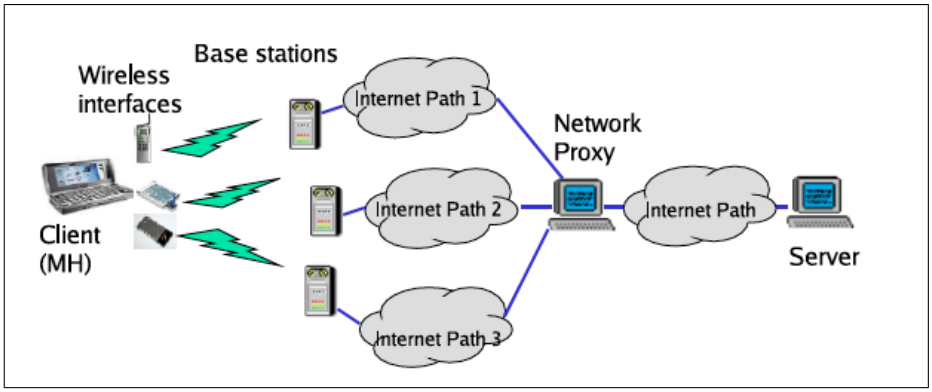


FIGURE 19 – Schéma de l'architecture [5]

L'objectif est d'implémenter une solution qui entraîne le moins de changements au niveau de l'infrastructure et des applications existantes. La modification du comportement de la couche réseau de la pile protocolaire TCP/IP a été choisie car une modification de sa couche physique, de sa couche de liaison ou de sa couche transport est plus complexe en termes de déploiement et de coût.

Les auteurs proposent une architecture sur base d'un proxy (Figure 19) dont les paramètres sont adaptés pour l'expérience.

Les critères de design sont : l'utilisation de la bande passante de toutes les interfaces sans fil, la minimisation du réordonnancement des paquets, la dissimulation des effets du réordonnancement, la détection des pertes de paquets et leur traitement dans un délai opportun, l'évitement du *burstiness* ou *rafale* de paquets et, pour terminer, l'isolation des pertes de paquets par chemin.

L'algorithme se base sur la discipline de planification *Earliest Delivery Path First)* minimisant le réordonnancement des données et la technique *packet pair* pour évaluer la bande passante. Des règles de gestion de tampons gèreront les critères de design restants.

La possibilité de découper un bloc de données pour l'envoyer sur plusieurs chemins différents (data-stripping) est abordée ainsi qu'une comparaison avec SCTP qui est évoqué à la Section 6.

Les différents appareils mobiles actuels possèdent plusieurs types de connexions sans fil (Wifi, UMTS/3G, etc).

Il semble donc possible d'augmenter le débit et la robustesse de la transmission, au niveau de l'application, en utilisant plusieurs chemins de connexion (multi-path) aux réseaux sans fil.

La désagrégation de données (data-stripping) permet d'augmenter le débit tandis que la duplication de données (data-duplicating) permet d'améliorer la robustesse de la transmission.

La sélection du mode de fonctionnement peut se faire en fonction de l'état des liaisons sans fil établies par les interfaces réseau d'un système mobile.

La manière d'utiliser simultanément de multiples interfaces réseau d'un appareil mobile est devenue un sujet de recherche.

La solution est performante et facile à déployer.

## 8.3 Transmission de données sur plusieurs chemins (multi-path, multihoming) en utilisant SCTP

Dans [12], les auteurs proposent WiMP-SCTP( Wifi Multi Path-SCTP) permettant à un système de transmettre des données par plusieurs chemins : soit par désagrégation (data-stripping mode), soit par duplication (data-duplicating mode).

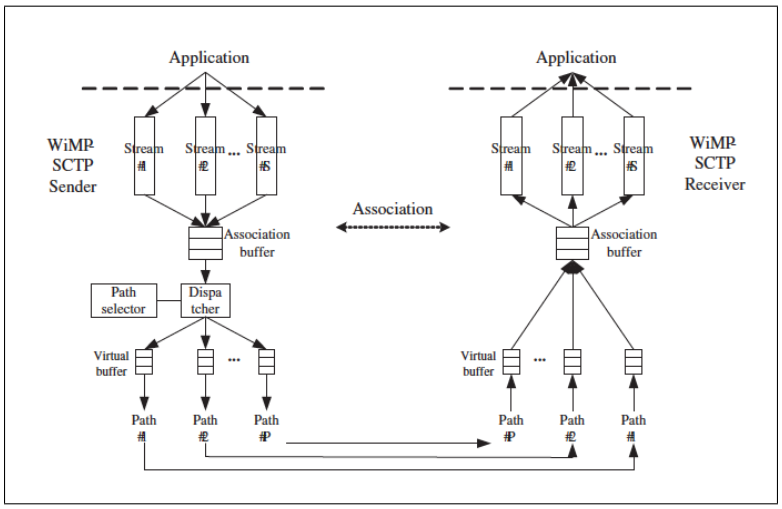


FIGURE 20 – Schéma de l'architecture de WiMP-SCTP [5]

Dans le mode data-stripping, les données sont découpées et transmises par plusieurs chemins, tandis que dans le mode data-duplicating, les données sont dupliquées et également transmises par plusieurs chemins. Le mode duplication de données améliore considérablement la fiabilité de la transmission de données tandis que le mode désagrégation améliore fortement le débit de la transmission.

Le passage du mode data-stripping au mode data-duplicating s'effectue lorsque la qualité des chemins de transmission diminue. La sélection de mode sera basée sur le calcul du nombre de retransmissions consécutives et le nombre de dépassements de délais de transmissions. WiMP-SCTP utilise un seuil de basculement paramètrable. Si le nombre de connexions possibles est supérieur au nombre d'interfaces réseau disponibles, il faut faire un choix. Ce mécanisme de décision s'enclenche soit lors de l'initialisation de la communication avec un partenaire, soit lors de la détection d'un nouveau chemin ou lorsqu'un chemin est perdu.
La sélection d'un chemin est effectuée en évaluant un délai de transmission aller-retour de données. Pour supporter la sélection de mode, un nouveau champ binaire a été ajouté au datagramme d'un paquet SCTP. Le SCTP de base a été modifié pour supporter les fonctions de WiMP-SCTP. L'émetteur et le récepteur reçoivent les fonctions nécessaires pour traiter les deux modes de transmission.

De plus, l'émetteur reçoit deux fonctions supplémentaires : un dispatcheur de données et un sélecteur de chemins. L'émetteur et le récepteur se voient équipés d'un tampon virtuel pour l'assignation des données devant suivre un chemin déterminé. Le seuil de basculement d'un mode à l'autre doit être adapté aux conditions de l'environnement.

Le mode data-stripping pose un problème important de réordonnancement des paquets de données quand la qualité de la transmission n'est pas bonne. Ce problème provoquera des retransmissions et augmentera le taux de perte de paquets de données.

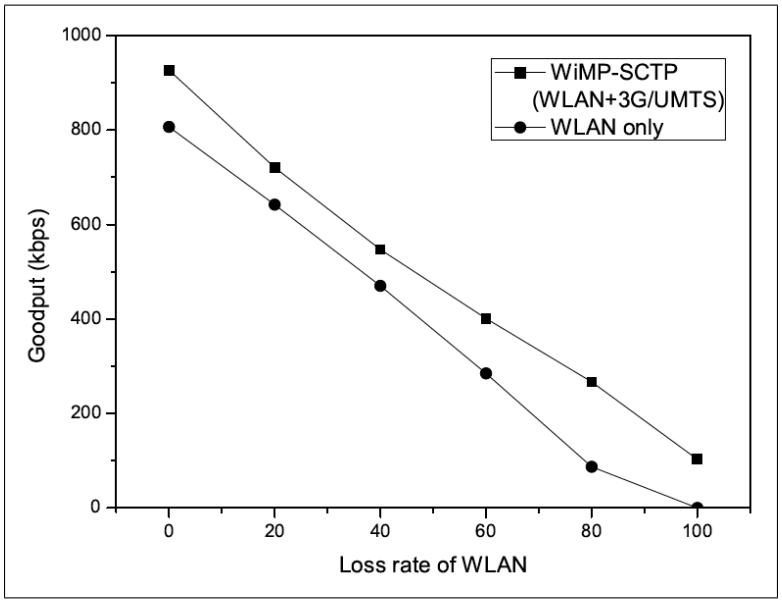


FIGURE 21 – Comparaison entre la performance de WiMP-SCTP et un seul chemin WiFi [5]

L'expérience montre que les meilleurs résultats sont obtenus généralement avec un seuil de basculement peu élevé.

Le mode data-stripping apporte un réel bénéfice lorsque le taux de perte est faible tandis que le mode data-duplicating apporte un réel bénéfice lorsque le taux de perte est élevé.

La Figure 21 montre la comparaison de la performance obtenue avec WiMP-SCTP contre l'utilisation d'un seul chemin WiFi.

## 8.4 Définition du problème

Nous avons introduit la possibilité d'utiliser plusieurs chemins radio au niveau de la couche physique à la Section 5.

A la Section 8.1, nous mettons en évidence qu'à eux seuls, les systèmes travaillant au niveau de la couche physique et de la couche de liaison peuvent assurer un débit et une fiabilité de transmission uniquement sur leur réseau local. Une collaboration entre la couche physique et la couche transport pourrait améliorer le rendement de la couche transport.

Nous abordons à la Section 8.2, la possibilité de mutualiser plusieurs interfaces réseau et, à la Section 8.3, la possibilité de mutualiser des connexions au niveau de la couche transport. Les problèmes majeurs rencontrés sont le réordonnancement des séquences et la sélection entre la performance ou la robustesse.

De plus, un nœud peut se connecter par un réseau performant, mais au-delà de son point d'accès, il est impossible de connaître la nature du matériel qu'il y a derrière et son comportement (middle box).

Un chemin entre deux nœuds n'est pas toujours déterministe. Le chemin peut changer s'il rencontre une difficulté sur son trajet.

La gestion de l'équilibrage de la charge peut contraindre un chemin (Load Balancing, Least-Cost Routing layer 3, Spanning-Tree layer 2) plutôt qu'un autre.

La bande passante entre chaque nœud est une inconnue.

Il est primordial de s'abstraire des *middle boxes* et de permettre une *rétro-compatibilité TCP* en cas de difficulté.

Agir sur le comportement de la couche *transport* est une alternative mais à condition d'être transparent pour la couche applicative.

Travailler au niveau de la couche *réseau* de la pile protocolaire TCP/IP ne permet pas une implémentation moins coûteuse que d'aligner une solution matérielle sur les chemins possibles entre les nœuds d'un réseau IP.

Toute modification de la couche *réseau* (IPv4 ou IPv6) implique aussi d'importants coûts de mise à jour de l'infrastructure de transport.

Théoriquement, il n'y pas de contres indications plus importantes à combiner un système MIMO et un protocole comme MPTCP vu à la Section 7, que de combiner un système MIMO et le protocole TCP tel qu'on le connait aujourd'hui.

# 9 Expérimentation

Dans cette section, nous tentons de vérifier si le bénéfice des technologies utilisées pour la couche matérielle de la pile TCP/IP s'ajoute ou entre en concurrence avec les technologies employées pour la couche transport. Nous allons combiner un système MIMO-OFDM avec MPTCP et mesurer la performance de cette combinaison.

## 9.1 Environnement

L'expérimentation se déroule dans le centre de données du Centre Hospitalier Régional de la Citadelle de Liège. L'intérêt d'effectuer nos tests dans cet environnement vient du fait que ce local est une cage de Faraday. Une cage de Faraday garantit qu'aucune onde électro-magnétique ne peut entrer, ni sortir de celle-ci. Nous garantissons ainsi que nos mesures ne seront pas perturbées par d'éventuels systèmes *radio-fréquence* externes. L'alimentation électrique est régulée ; ce qui nous protège des perturbations du réseau électrique.

## 9.2 Méthodologie

### 9.2.1 Instrumentation

Chaque élément technique participant à l'expérimentation est dimensionné pour ne pas être une entrave à la performance. Chaque élément logiciel est au même niveau de version pour tous les environnements utilisés (client et serveur). Chaque périphérique utilisé fonctionne avec un pilote certifié par l'éditeur de la distribution du système d'exploitation. Cette certification nous assure la stabilité du périphérique. Afin d'être accessible au plus grand nombre et pour la facilité d'accès au code source, une préférence est donnée aux logiciels libres.

a) **Le Matériel**

- Nous utilisons deux ordinateurs à architecture *Intel* (Core i7) équipés, chacun, de 4GB de RAM et d'une unité de stockage de 250GB. L'un aura pour fonction d'être *le serveur* et l'autre sera *le client*.

- Pour la partie communication filaire (Unshielded Twisted Pair), nous employons deux commutateurs *réseau* (switches) identiques ayant la particularité de pouvoir imposer la vitesse de communication de leur port en 10BASE-T, 100BASE-T et 1000BASE-T. Tous les câbles de connexion sont certifiés *catégorie 5E* et testés. Cette catégorie permet le transport d'information de 10BASE-T à 1000BASE-T. Chaque chemin MPTCP a son propre switche afin d'isoler physiquement le trafic

- Pour la partie WiFi, nous utilisons un point d'accès (*Access Point*) répondant à la norme IEEE 802.11n (MIMO-OFDM). Il possède deux modules

radio distincts. Le premier permet de travailler simultanément avec les standards IEEE 802.11b, g et n, à 2,4GHz. Le second permet de travailler simultanément avec les standards IEEE 802.11a et n, à 5GHz. Chaque module permet l'activation ou la désactivation de la diversité. Lorsque la diversité est désactivée, seules une antenne est active. Il n'y a donc plus de diversité spatiale. Nous utilisons un seul module à la fois.

b) **Les logiciels**

- Le système d'exploitation (plateforme) pilotant ces machines est une distribution *Linux 64-bit*. L'équipe de l'*IP Networking Lab* (INL) de l'Université Catholique de Louvain (UCL) en a modifié le noyau (kernel) pour l'intégration de MPTCP.

- Le choix du logiciel qui mesure la bande passante se porte sur IPERF [33]. Ce logiciel est multi-plateforme (MS-Windows, Linux, Mac OS X, Free BSD, Solaris). L'abondance de ses options permet d'influer sur les principaux paramètres de la transmission (Receive Buffer, protocole, durée du test, MTU, port, etc.).

### 9.2.2 Configuration de base

Dans ce scénario, nous avons trois acteurs principaux, la machine *serveur*, le point d'accès WiFi et la machine *client*.

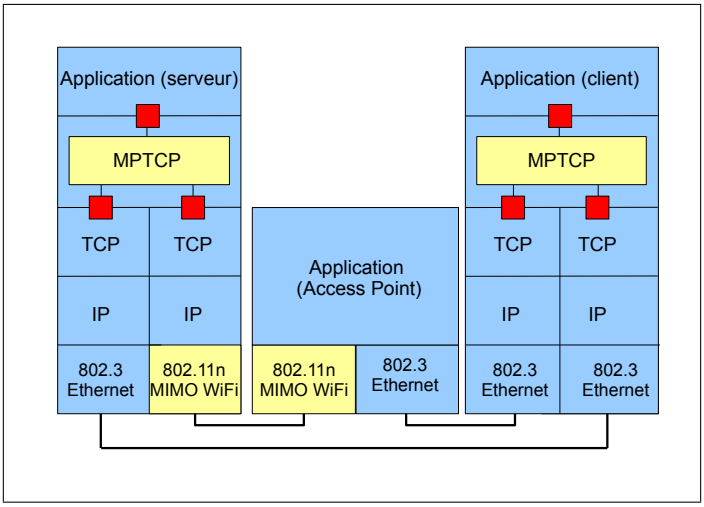


FIGURE 22 – Schéma de principe

La première interface ethernet de chaque machine est définie dans un premier réseau IP et inter-connectée à un premier réseau physique. La deuxième interface de chaque machine est définie dans un second réseau IP et est inter-connectée à un deuxième réseau physique. Ainsi les deux chemins créés sont totalement indépendants aussi bien physiquement, que logiquement. La figure 22 représente notre scénario suivant la pile protocolaire TCP/IP.

Nous voulons observer le comportement de notre combinaison avec des chemins ayant une bande passante différente. Pour ce faire, nous bridons l'interface gigabit ethernet 802.3 du client et du serveur, à 10 Mb/s et à 100 Mb/s. Cette diminution de vitesse est obtenue en modifiant la vitesse des ports ethernet du switch concerné. Nous effectuons les tests aux trois vitesses (10 Mb/s, 100 Mb/s et 1Gb/s). Nous effectuons nos mesures alternativement sur les bandes de fréquence WiFi 2,4 GHz (20 MHz) et 5 GHz (40 MHz).

Chaque mesure de l'expérience est renouvelée dix fois sur trois durées de temps (10, 30 et 60 secondes) afin de détecter un éventuel comportement non-déterministe.

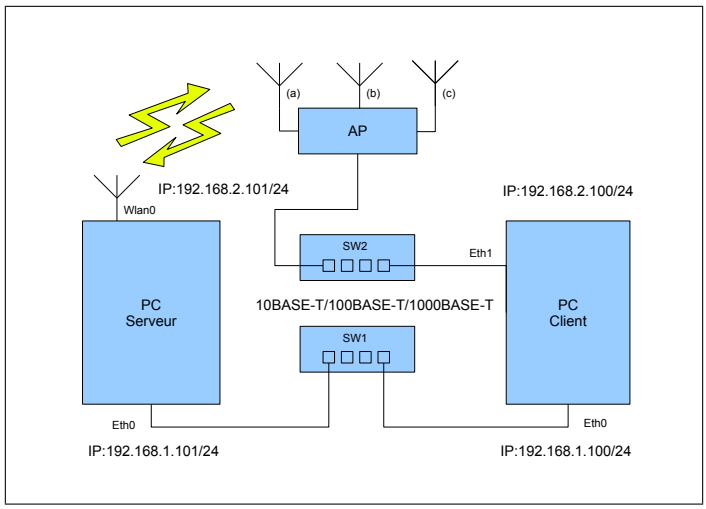


FIGURE 23 – Schéma d'implémentation de la configuration de base

La répétition des mesures est scriptée afin d'éviter le facteur humain au maximum. Par ce script, nous mesurons la latence (RTT) par la commande *ping* du système d'exploitation et la bande passante par le programme IPERF. Ce dernier est démarré en mode *serveur* sur la machine *serveur* et en mode *client* sur la machine *client*.

### 9.2.3 Paramétrage

L'installation du kernel *MPTCP* (3.0.0-19-mptcp) est réalisée suivant la note d'installation fournie par l'INL [14]. Par facilité, nous avons opté pour l'installation par package pré-compilé tout en sachant que ses options de compilation sont orientées sur la robustesse et non sur la performance.

La configuration IP des deux chemins est effectuée suivant les recommandations de l'INL [13]. Il y a deux réseaux IP (192.168.1.0/24 et 192.168.2.0/24). Le serveur et le client ont une interface dans chacun des réseaux.

Le point d'accès WiFi est configuré suivant un profil prédéfini par le constructeur assurant le meilleur débit.

La configuration des switches est une configuration standard du constructeur permettant le changement de la vitesse des ports en 10BASE-T, 100BASE-T et 1000BASE-T.

Les fonctions d'*energy saving* ou tout autre type d'économiseur sont désactivées pour ne pas influencer la mesure.

## 9.3 Calibration

Afin de détecter, dès le départ, d'éventuels problèmes de configuration, une calibration des performances s'impose.

Nos mesures se reposent sur un kernel pré-compilé avec le TCP de base tel que distribué par l'éditeur (3.0.0-19-generic).

Nous calibrons la performance de chaque interface réseau (client et serveur).

Nous croisons les résultats obtenus avec une référence pour valider la calibration.

Enfin, nous créons un tableau de calibration (Tableau 1, Section 9.3.5) que nous utilisons tout au long de notre expérimentation pour valider les résultats obtenus.

### 9.3.1 Le programme de mesure

Pour vérifier la qualité des mesures du programme IPERF, nous avons comparé les valeurs avec celles obtenues par la mesure chronométrée d'un transfert de fichier d'une taille de 250 MB, 1GB et 4,7GB en 10, 100 et 1000Mb/s. Les mesures sont concordantes.

### 9.3.2 Les interfaces IEEE 802.3

Les interfaces ethernet sont calibrées sur base des normes 10BASE-T, 100BASE-T et 1000BASE-T.

### 9.3.3 L'interface IEEE 802.11n

Lors de la calibration en 5GHz, nous nous sommes aperçus que le pilote de la carte WiFi ne gère pas le 5GHz sous *Linux 64-bit*. Dès lors, pour s'abstraire du problème *matériel*, nous utilisons une autre machine ayant une configuration *matérielle* similaire mais opérationnelle en 2,4 GHz et en 5GHz, sous MAC OS X. Un nouveau schéma de principe est proposé à la figure 24.

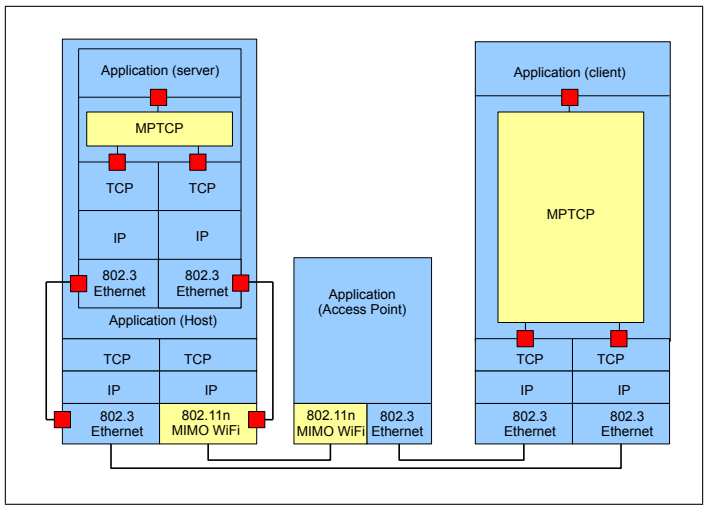


FIGURE 24 – Schéma de principe adapté à la virtualisation

A partir de ce nouvel environnement, nous utilisons une couche de virtualisation nous permettant de faire fonctionner une machine virtuelle (VM) *Linux 64-bit* à l'aide de la machine physique (HOST). La VM possède deux interfaces ethernet IEEE 802.3, l'une mappée sur l'interface ethernet *physique* gigabit ethernet (*eth0*) et l'autre sur la carte *physique* WiFi (*eth1*). Nous adaptons notre schéma d'implémentation en conséquence à la figure 25.

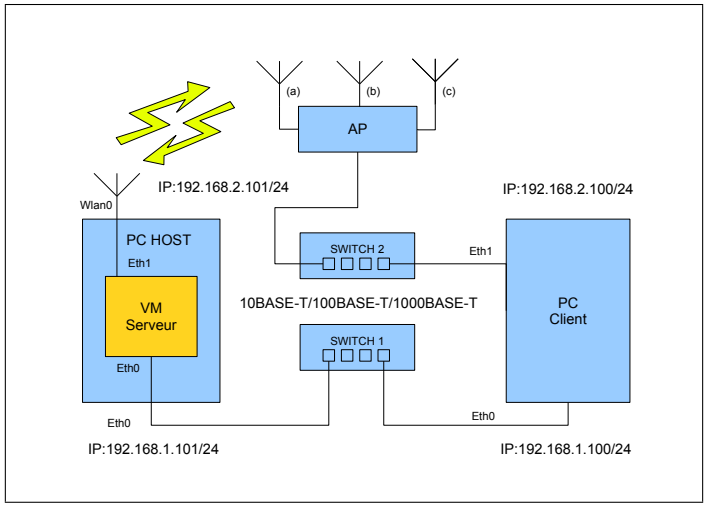


FIGURE 25 – Schéma d'implémentation adapté à la virtualisation

Le revers de cette solution de contournement est l'introduction des pertes de performance globale (overhead) dues aux latences créées par les mécanismes de la virtualisation.

Pour implémenter la virtualisation, nous faisons appel au logiciel *Virtual Box*. Multi-plateforme, il permet d'émuler les principaux systèmes d'exploitation présents sur le marché en 32-Bit ou en 64-Bit sur les principaux systèmes d'exploitation du marché.

Il est important d'installer les outils de gestion de la virtualisation au sein de la machine virtuelle pour un fonctionnement optimal des ressources (disques, cartes ethernet, etc). Dès qu'il y a un changement de kernel, il est impératif de recompiler les outils de gestion dans la VM.

Notre point de mesure sera situé sur l'interface ethernet du point d'accès WiFi.

### 9.3.4 Le fonctionnement simultané des interfaces

Pour chaque terminal, le calibrage des interfaces en utilisation simultanée est primordial. En effet, si le bus de données des interfaces n'est pas suffisamment dimensionné, nous nous exposons à un phénomène de *goulot d'étranglement* (shared bottleneck) et les performances attendues ne sont pas au rendez-vous.

A titre d'exemple, $n$ interfaces WiFi sont connectées chacune sur un port USB. En général, le bus de données est commun. Vous obtiendrez un calibrage correct par interface. Par contre, utilisées simultanément, vous obtiendrez une performance limitée à celle du bus de données et non égale à la somme théorique des bandes passantes propres à chaque interface.

### 9.3.5 Table de calibration

| Station | Interface | 10BASE-T (Mb/s) | | | | 100BASE-T (Mb/s) | | | | 1000BASE-T (Mb/s) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Client | eth0<br>eth1 | 10<br>10 | | | | 95<br>95 | | | | 937<br>937 | | | |
| Host | eth0 | 10 | | | | 99 | | | | 999 | | | |
| | Wlan0 | Diversity | | | | | | | | | | | |
| | | 2,4 GHz | | 5 GHz | | 2,4 GHz | | 5 GHz | | 2,4 GHz | | 5 GHz | |
| | | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF |
| | | 10 | 10 | 10 | 10 | 98 | 49 | 99 | 99 | 98 | 49 | 194 | 114 |
| VM | eth0 | 10 | | | | 99 | | | | 225 | | | |
| | eth1 | Diversity | | | | | | | | | | | |
| | | 2,4 GHz | | 5 GHz | | 2,4 GHz | | 5 GHz | | 2,4 GHz | | 5 GHz | |
| | | ON | OFF | ON | ON | ON | OFF | ON | OFF | ON | OFF | ON | OFF |
| | | 10 | 10 | 10 | 10 | 50 | 41 | 52 | 51 | 54 | 45 | 65 | 55 |

TABLE 1 – Table de calibration

### 9.3.6 Observations

La Table 1, nous permet de voir que la couche de virtualisation tend à limiter la bande passante de l'interface eth1 aux alentours de 54 Mb/s lorsque celle-ci n'est plus étranglée par la vitesse du switch (100BASE-T et 1000BASE-T). Il en va de même pour l'interface eth0 qui voit sa bande passante limitée aux alentours de 230 Mb/s en 1000BASE-T. L'éditeur a créé ces limitations pour q'une VM ne puisse pas saturer la machine physique qui l'héberge.

# 10 Analyse des résultats

Dans cette section, nous présentons les trois tableaux de nos mesures en limitant la bande passante (BandWidth) de toutes les interfaces ethernet, en 10BASE-T, 100BASE-T et 1000BASE-T.

La BW du réseau IEE 802.3 et du réseau IEEE 802.11n sont représentées par les mesures obtenues sur les interfaces *eth0* et *eth1* du serveur.

Les valeurs d'un tableau sont les moyennes des dix mesures de la BW, dans la même configuration. Ceci pour chaque chemin (eth0 et eth1) isolé et pour le pool des chemins gérés par MPTCP.

Nous mesurons le tout à 2,4 GHz, à 5GHz et en activant ou pas la diversité.

Nous avons ajouté, en complément de la moyenne arithmétique, l'écart type. Ce dernier nous permet de voir, au-delà d'une simple moyenne, la variation des mesures.

Nous utilisons notre table de calibration (Table 1) comme référence pour nos observations.

Nous souhaitons évaluer la performance de chaque chemin MPTCP isolément pour le comparer avec TCP. Ensuite, nous souhaitons évaluer la performance des deux chemins travaillant simultanément via MPTCP pour en observer le gain apporté.

## 10.1 Limitation des interfaces ethernet en 10BASE-T

| INTERFACE | 2,4 GHz | | | | 5 GHz | | | |
|---|---|---|---|---|---|---|---|---|
| | Diversity | | | | | | | |
| | ON | | OFF | | ON | | OFF | |
| | BW (Mb/s) | Ecart type | BW (Mb/s) | Ecart type | BW (Mb/s) | Ecart type | BW (Mb/s) | Ecart type |
| eth0 | 10 | 0 | 10 | 0 | 10 | 0 | 10 | 0 |
| eth1 | 10 | 0 | 10 | 0 | 10 | 0 | 10 | 0 |
| MPTCP | 19,3 | 0,1 | 19,3 | 0,1 | 19,4 | 0,2 | 19,4 | 0,1 |

TABLE 2 – Table des résultats en 10BASE-T

### 10.1.1 Observations

– **Sur le chemin IEEE 802.3 (eth0)**

Le débit est limité à 10 Mb/s par les ports du switch.

– **Sur le chemin IEEE 802.11n (eth1)**

Ici aussi, le débit est limité à 10 Mb/s.

– **Sur MPTCP**

Le résultat obtenu par MPTCP est proche de la somme des interfaces eth0 et eth1. MPTCP introduit aussi de l'overhead (réordonnancement, checksum, etc).

## 10.2 Limitation des interfaces ethernet en 100BASE-T

| INTERFACE | 2,4 GHz | | | | 5 GHz | | | |
|---|---|---|---|---|---|---|---|---|
| | Diversity | | | | | | | |
| | ON | | OFF | | ON | | OFF | |
| | BW (Mb/s) | Ecart type | BW (Mb/s) | Ecart type | BW (Mb/s) | Ecart type | BW (Mb/s) | Ecart type |
| Eth0 | 99 | 0 | 99 | 0 | 99 | 0 | 99 | 0 |
| eth1 | 44,7 | 0,7 | 45,4 | 0,5 | 56 | 1,2 | 45,4 | 0,5 |
| MPTCP | 142,5 | 4,5 | 120 | 4,2 | 155 | 8,7 | 139 | 8,2 |

TABLE 3 – Table des résultats en 100BASE-T

### 10.2.1 Observations

– **Sur le chemin IEEE 802.3 (eth0)**

Le débit est limité à 1OO Mb/s par la porte du switch.

– **Sur le chemin IEEE 802.11n (eth1)**

Comme observé à la Section 9.3.6, eth1 étant limité aux alentours de 54 Mb/s, son débit ne dépassera jamais ce seuil, en 2,4 GHz comme en 5 GHz.

On remarque que le résultat obtenu, sur eth1, avec la diversité est supérieur à celui obtenu sans la diversité. On observe un écart type légèrement plus haut lors de la limitation plus importante du débit.

En 5 Ghz, on observe le même phénomène de saturation. Le débit en amont sur la carte physique étant nettement plus haut qu'en 2,4 GHz, la saturation de eth1 est plus forte. Dans ce cas, le débit obtenu avec la diversité est sensiblement plus grand que sans la diversité.

La limitation de l'interface virtuelle reliée à l'interface physique nous empêche d'observer correctement le comportement de MPTCP en relation avec la diversité.

Sans le calibrage préalable des interfaces, nous n'aurions pas pu interpréter correctement les mesures.

– **Sur MPTCP**

Nous continuons à avoir des performances plus importantes que si nous utilisions uniquement le chemin IEEE802.3.

Le gain de la bande passante diminue car les chemins sont devenus asymétriques (100Mb/s pour eth1 et 50Mb/s pour eth1). L'algorithme d'optimisation du *receive buffer* décrit dans [6] tend à limiter l'utilisation du chemin le plus lent et possédant le RTT le plus long (7 ms de moyenne pour eth0 et 70ms de moyenne pour eth1). Le débit serait plus important si nous avions deux chemins symétriques.

## 10.3 Limitation des interfaces ethernet en 1000BASE-T

| INTERFACE | 2,4 GHz | | | | 5 GHz | | | |
|---|---|---|---|---|---|---|---|---|
| | Diversity | | | | | | | |
| | ON | | OFF | | ON | | OFF | |
| | BW (Mb/s) | Ecart type | BW (Mb/s) | Ecart type | BW (Mb/s) | Ecart type | BW (Mb/s) | Ecart type |
| Eth0 | 223,5 | 5,1 | 223,5 | 5,1 | 223,5 | 5,1 | 223,5 | 5,1 |
| eth1 | 48,2 | 1,7 | 45,5 | 1,5 | 55,7 | 1,2 | 51 | 0,6 |
| MPTCP | 250 | 18,8 | 250,5 | 18,5 | 239,5 | 17,8 | 253,5 | 17,5 |

TABLE 4 – Table des résultats en 1000BASE-T

### 10.3.1 Observations

– **Sur le chemin IEEE 802.3 (eth0)**

La vitesse de l'interface eth0 n'est plus étranglée par le switch.

On observe la même limitation remarquée lors du calibrage qui réduit le débit de l'interface virtuelle aux alentours de 220 Mb/s.

– **Sur le chemin IEEE 802.11n (eth1)**

Nous observons la même situation qu'en 100BASE-T, soit une saturation de l'interface et un travail plus intense effectué par la virtualisation pour restreindre le débit aux alentours de 54 Mb/s lorsque la diversité est activée.

– **Sur MPTCP**

On observe un écart type légèrement plus haut lors de la limitation plus importante du débit. Le gain de la bande passante diminue car la différence de vitesse entre eth0 et eth1 est plus prononcée. La performance obtenue sans la diversité est légèrement plus élevée que lorsqu'elle est active. Nous expliquons cet écart par la latence introduite par la virtualisation pour restreindre le débit qui se cumule à l'overhead de MPTCP.

La performance de MPTCP reste supérieure à celle de TCP sur le chemin le plus rapide.

De plus, nous avons poussé l'expérience plus loin en étranglant la vitesse du chemin IEEE 802.11n en 10BASE-T, tout en laissant l'interface eth0 en 1000BASE-T. L'asymétrie augmentant, il n'y a plus de différence de performance entre MPTCP et TCP par son chemin le plus rapide. Le bénéfice potentiel du second chemin est perdu.

Dès lors, nous pouvons constater que MPTCP respecte un engagement primordial à son déploiement, à savoir, ne pas être moins performant que TCP. Il maintient une vitesse de transfert similaire au chemin TCP le plus rapide.

Nous avons également testé la résilience de MPTCP face à la perte d'un chemin. En débranchant un chemin, pendant un transfert de données, nous avons observé que le transfert continue sa progression, sans interruption, à la vitesse du ou des chemins restants. L'amélioration de la bande passante revient lorsque MPTCP retrouve le chemin perdu.

La limitation de débit, propre à la couche de virtualisation, ne nous permet pas d'observer autre chose que le fonctionnement intrinsèque de MPTCP.

# 11 Conclusion

Nous rappelons que les objectifs de ce mémoire étaient de présenter un état de l'art sur la combinaison des bénéfices de plusieurs technologies travaillant à différents niveaux de la pile protocolaire TCP/IP, de mettre en oeuvre un prototype, de l'expérimenter et, enfin, d'en observer les bénéfices ou non.

Nous avons commencé par introduire notre sujet de mémoire, à la Section 3.

Nous avons évoqué, à la Section 4, le concept de la pile protocolaire TCP/IP garantissant une isolation entre les différentes couches la composant. Nous avons utilisé son principe pour présenter et situer les différents concepts.

Ensuite à la Section 5, nous avons décrit les systèmes MIMO qui sont une avancée technologique incontournable, apportée à la couche physique, pour leur fiabilité et leur performance. Ces systèmes sont de plus en plus intégrés dans les nouvelles technologies sans fil présentes et à venir (WiFi IEEE 802.11n, 3G, etc).

A la Section 6, nous avons évoqué le protocole SCTP situé au niveau de la couche transport. Il est la première extension du protocole TCP permettant d'utiliser plusieurs chemins de transmission de données. Malheureusement, ce protocole est peu supporté par les middle boxes et nécessite une adaptation des applications courantes.

Situé également au niveau de la couche transport, nous avons présenté, à la Section 7, le nouveau protocole MPTCP ayant pour principe l'utilisation simultanée d'un pool de chemins. Ce concept offre d'une part, une meilleure robustesse en cas de problème sur un ou plusieurs liens et d'autre part, le cumul de leur bande passante respective. Son déploiement n'entraîne aucune modification de l'infrastructure existante et des applications. En cas de problèmes avec les middles boxes, il se comporte comme TCP.

Nous avons réalisé, à la Section 8, un état de l'art traitant de la combinaison de plusieurs technologies travaillant sur des couches protocolaires différentes ainsi que leur interaction potentielle.

Nous avons défini un scénario d'expérimentation, à la Section 9, par la combinaison du récent protocole MPTCP et d'un système MIMO en IEEE 802.11n.

Après avoir expérimenté ce scénario, nous avons, à la Section 10, comparé les résultats des différentes mesures au calibrage effectué en TCP.

Lors de cette expérimentation, nous avons dû virtualiser la machine ayant une interface WiFi pour solutionner un problème de compatibilité technique avec le système d'exploitation (MPTCP n'est actuellement développé que pour Linux). La virtualisation a introduit une limitation de la bande passante des interfaces

réseau virtualisées. Cette limitation a étranglé la couche physique virtualisée et elle ne lui a pas permis d'atteindre les taux de transfert attendus par elle seule.

Néanmoins, nous avons pu constater l'efficacité du fonctionnement intrinsèque de MPTCP comme annoncé dans la littérature. Nous avons pu apprécier la robustesse apportée en cas perte d'un chemin. La bande passante résultante de MPTCP est la somme de la bande passante de ses chemins respectifs en déduisant l'overhead développé par ses mécanismes de gestion. Nous avons pu vérifier que plus la différence de vitesse entre les chemins augmente, plus la performance de MPTCP se dégrade.

Au terme de ce mémoire, nous sommes personnellement convaincus qu'une interaction entre la couche physique et la couche transport, consistant à communiquer, par exemple, le contexte BER de la couche physique, permettrait à la couche transport d'optimiser ses décisions. Cependant nous violerions le concept d'isolation de la pile protocolaire TCP/IP qui n'a plus été adaptée depuis plus de vingt ans.

Pour nous permettre d'augmenter la robustesse et la performance de nos transmissions de demain, il serait peut être temps de concevoir un nouveau modèle protocolaire plus interactif.

# 12 Futur Work

Nous souhaiterions tester notre modèle d'implémentation sans passer par la virtualisation.

Nous voudrions pousser plus loin notre expérience en testant notre modèle avec plusieurs types de systèmes MIMO (IEEE 802.16, IEEE 802.11ac, etc).

Aujourd'hui, les nouveaux mobiles (tablette, smartphone, etc) possèdent plusieurs coeurs à l'intérieur d'un même micro-processeur. A ce stade, MPTCP ne peut utiliser qu'un seul coeur. Il serait intéressant de se pencher sur la possibilité d'employer un pool de coeurs simultanément.

Un problème important serait la coordination des coeurs pour le réordonnancement des paquets.

L'étude d'un nouveau concept permettant une interaction entre différentes couches physiques (réseau, gestion de l'énergie, gestion multi-processeur) et la couche transport est un sujet de recherche à développer.

# 13 Annexes

## 13.1 Inventaire de l'expérimentation

### 13.1.1 Les logiciels

- **Distribution Linux**
  Ubuntu 11.10 64-bit (*The Oneiric Ocelot*)

- **IPERF**
  version 2.0.5 (08 Jul 2010) pthreads

- **Kernel MPTCP**
  3.0.0-19-mptcp #33 SMP Sat Apr 14 00 :08 :48 CEST 2012

- **Mac OS X**
  Darwin Kernel Version 11.4.0 : Mon Apr 9 19 :32 :15 PDT 2012

- **Oracle Virtual Box**
  version 4.1.14r77440

### 13.1.2 Le matériel

- **PC Serveur**
  Mac Book Pro
  - CPU Core I7
  - 4GB RAM
  - 1 port ethernet 1000BASE-T
  - 1 port IEEE 802.11n

- **PC Client**
  Fujitsu Esprimo P400
  - CPU Core I7
  - 4GB RAM
  - 4 ports ethernet 1000BASE-T

- **Point d'accés WiFi IEEE 802.11n**
  Cisco Aironet Model :1250 AIR-AP1252AG-x-K9
  - 2x3 MIMO with two spatial streams
  - Maximal Ratio Combining (MRC)
  - 20-and 40-MHz channels
  - PHY data rates up to 300 Mbps
  - Packet aggregation : A-MPDU (Tx/Rx), A-MSDU (Tx/Rx)
  - 802.11 DFS (Bin 5)
  - Cyclic Shift Diversity (CSD) support

– **Switches**
  2 x Cisco Catalyst 3750-E (24 ports Ethernet 10/100/1000)

## 13.2 Tables des mesures MPTCP

### 13.2.1 Mesures du chemin eth1 10BASE-T à 2,4 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 10 | 10 |
| 2 | 10 | 10 |
| 3 | 10 | 10 |
| 4 | 10 | 10 |
| 5 | 10 | 10 |
| 6 | 10 | 10 |
| 7 | 10 | 10 |
| 8 | 10 | 10 |
| 9 | 10 | 10 |
| 10 | 10 | 10 |
| Moyenne | 10 | 10 |
| Ecart type | 0 | 0 |

Table 5 – Table des mesures du chemin eth1 10BASE-T à 2,4 GHz

### 13.2.2 Mesures MPTCP 10BASE-T en 2,4 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 19,3 | 19,3 |
| 2 | 19,4 | 19,4 |
| 3 | 19,4 | 19,4 |
| 4 | 19,4 | 19,4 |
| 5 | 19,4 | 19,4 |
| 6 | 19,3 | 19,3 |
| 7 | 19,4 | 19,4 |
| 8 | 19,4 | 19,4 |
| 9 | 19,3 | 19,3 |
| 10 | 19,3 | 19,3 |
| Moyenne | 19,3 | 19,3 |
| Ecart type | 0,1 | 0,1 |

Table 6 – Table des mesures MPTCP 10BASE-T à 2,4 GHz

### 13.2.3 Mesures du chemin eth1 100BASE-T à 2,4 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 45 | 45,2 |
| 2 | 45,7 | 44,4 |
| 3 | 43,6 | 44,6 |
| 4 | 44,6 | 44,6 |
| 5 | 43,6 | 44,6 |
| 6 | 44,6 | 45,1 |
| 7 | 43,5 | 45,7 |
| 8 | 43,9 | 44,9 |
| 9 | 44,4 | 45,4 |
| 10 | 44,4 | 45,5 |
| Moyenne | 44,7 | 45,4 |
| Ecart type | 0,7 | 0,5 |

TABLE 7 – Table des mesures du chemin eth1 100BASE-T à 2,4 GHz

### 13.2.4 Mesures 2 chemins MPTCP 100BASE-T à 2,4 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 136 | 124 |
| 2 | 141 | 125 |
| 3 | 142 | 128 |
| 4 | 144 | 122 |
| 5 | 144 | 128 |
| 6 | 141 | 126 |
| 7 | 142 | 126 |
| 8 | 140 | 128 |
| 9 | 133 | 118 |
| 10 | 131 | 116 |
| Moyenne | 142,5 | 120 |
| Ecart type | 4,5 | 4,2 |

TABLE 8 – Table des mesures MPTCP 100BASE-T à 2,4 GHz

### 13.2.5 Mesures du chemin eth1 1000BASE-T à 2,4 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 47,3 | 43,8 |
| 2 | 49,4 | 47 |
| 3 | 47,2 | 10 |
| 4 | 51,5 | 47,1 |
| 5 | 49,5 | 45,7 |
| 6 | 49,7 | 46,2 |
| 7 | 50,4 | 49 |
| 8 | 52,6 | 48 |
| 9 | 49,1 | 48,9 |
| 10 | 49 | 47,1 |
| Moyenne | 48,2 | 45,5 |
| Ecart type | 1,7 | 1,5 |

TABLE 9 – Table des mesures du chemin eth1 1000BASE-T à 2,4 GHz

### 13.2.6 Mesures MPTCP 1000BASE-T à 2,4 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 245 | 254 |
| 2 | 286 | 287 |
| 3 | 275 | 305 |
| 4 | 251 | 265 |
| 5 | 300 | 286 |
| 6 | 245 | 257 |
| 7 | 277 | 280 |
| 8 | 251 | 282 |
| 9 | 251 | 282 |
| 10 | 255 | 247 |
| Moyenne | 250 | 250,5 |
| Ecart type | 18,9 | 18,5 |

TABLE 10 – Table des mesures MPTCP 1000BASE-T à 2,4 GHz

### 13.2.7 Mesures du chemin eth1 10BASE-T en 5 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 10 | 10 |
| 2 | 10 | 10 |
| 3 | 10 | 10 |
| 4 | 10 | 10 |
| 5 | 10 | 10 |
| 6 | 10 | 10 |
| 7 | 10 | 10 |
| 8 | 10 | 10 |
| 9 | 10 | 10 |
| 10 | 10 | 10 |
| Moyenne | 10 | 10 |
| Ecart type | 0 | 0 |

Table 11 – Table des mesures du chemin eth1 10BASE-T à 5 GHz

### 13.2.8 Mesures MPTCP 10BASE-T en 5 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 19,3 | 19,3 |
| 2 | 19,4 | 19,4 |
| 3 | 19,4 | 19,4 |
| 4 | 19,4 | 19,4 |
| 5 | 19,4 | 19,4 |
| 6 | 19,3 | 19,3 |
| 7 | 19,4 | 19,4 |
| 8 | 19,4 | 19,4 |
| 9 | 19,3 | 19,3 |
| 10 | 19,3 | 19,3 |
| Moyenne | 19,3 | 19,3 |
| Ecart type | 0,1 | 0,1 |

Table 12 – Table des mesures MPTCP 10BASE-T à 5 GHz

### 13.2.9 Mesures du chemin eth1 100BASE-T à 5 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 54,1 | 45,2 |
| 2 | 56,8 | 44,4 |
| 3 | 57,3 | 44,6 |
| 4 | 58,4 | 46,2 |
| 5 | 58 | 45,4 |
| 6 | 57,5 | 45,1 |
| 7 | 57,1 | 45,7 |
| 8 | 57,8 | 44,9 |
| 9 | 57,3 | 45,6 |
| 10 | 57,8 | 45,5 |
| Moyenne | 56 | 45,4 |
| Ecart type | 1,2 | 0,5 |

TABLE 13 – Table des mesures du chemin eth1 100BASE-T à 5 GHz

### 13.2.10 Mesures MPTCP 100BASE-T à 5 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 144 | 137 |
| 2 | 151 | 122 |
| 3 | 151 | 130 |
| 4 | 143 | 138 |
| 5 | 155 | 143 |
| 6 | 160 | 148 |
| 7 | 165 | 125 |
| 8 | 165 | 140 |
| 9 | 162 | 140 |
| 10 | 166 | 141 |
| Moyenne | 155 | 139 |
| Ecart type | 8,7 | 8,2 |

TABLE 14 – Table des mesures MPTCP 100BASE-T à 5 GHz

#### 13.2.11 Mesures du chemin eth1 1000BASE-T à 5 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 54,1 | 51,8 |
| 2 | 57,1 | 51 |
| 3 | 56,8 | 50,6 |
| 4 | 57,3 | 50,6 |
| 5 | 58,4 | 50,8 |
| 6 | 58 | 52 |
| 7 | 57,5 | 51,1 |
| 8 | 57,1 | 51,3 |
| 9 | 57,8 | 50,7 |
| 10 | 57,3 | 50,1 |
| Moyenne | 55,7 | 51 |
| Ecart type | 1,2 | 0,6 |

TABLE 15 – Table des mesures du chemin eth1 1000BASE-T à 5 GHz

#### 13.2.12 Mesures MPTCP 1000BASE-T à 5 GHz

| Numéro | Diversity | |
|---|---|---|
| | ON | OFF |
| | BW (Mb/s) | BW (Mb/s) |
| 1 | 241 | 270 |
| 2 | 264 | 238 |
| 3 | 233 | 249 |
| 4 | 263 | 234 |
| 5 | 286 | 249 |
| 6 | 241 | 285 |
| 7 | 277 | 244 |
| 8 | 251 | 267 |
| 9 | 245 | 271 |
| 10 | 238 | 237 |
| Moyenne | 239,5 | 253,5 |
| Ecart type | 17,8 | 17,5 |

TABLE 16 – Table des mesures MPTCP 1000BASE-T à 5 GHz

### 13.2.13 Mesures du chemin eth0

| Numéro | 10BASE-T (Mb/s) | 100BASE-T (Mb/s) | 1000BASE-T (Mb/s) |
|---|---|---|---|
| 1 | 10 | 99 | 225 |
| 2 | 10 | 99 | 222 |
| 3 | 10 | 99 | 222 |
| 4 | 10 | 99 | 216 |
| 5 | 10 | 99 | 222 |
| 6 | 10 | 99 | 226 |
| 7 | 10 | 99 | 217 |
| 8 | 10 | 99 | 234 |
| 9 | 10 | 99 | 222 |
| 10 | 10 | 99 | 226 |
| Moyenne | 10 | 99 | 223,5 |
| Ecart type | 0 | 0 | 5,1 |

TABLE 17 – Table des mesures du chemin eth0

## 13.3 Scripts

– **Configuration du routage**

– sur le Serveur

```
1  # This creates two different routing tables,
       that we use based on the source-address.
2    ip rule add from 192.168.1.100 table 1
3    ip rule add from 192.168.2.100 table 2
4
5
6    # Configure the two different routing
       tables
7    ip route add 192.168.1.0/24 dev eth0 scope
       link table 1
8    ip route add default via 192.168.1.1 dev
       eth0 table 1
9
10   ip route add 192.168.2.0/24 dev eth1 scope
       link table 2
11   ip route add default via 192.168.2.1 dev
       eth1 table 2
12
13   # default route for the selection process
       of normal internet-traffic
14   ip route add default scope global nexthop
       via 192.168.1.1 dev eth0
```

– sur le Client

```
1  # This creates two different routing tables,
       that we use based on the source-address.
2    ip rule add from 192.168.1.101 table 1
3    ip rule add from 192.168.2.101 table 2
4
5
6    # Configure the two different routing
       tables
7    ip route add 192.168.1.0/24 dev eth0 scope
       link table 1
8    ip route add default via 192.168.1.1 dev
       eth0 table 1
9
10   ip route add 192.168.2.0/24 dev eth1 scope
       link table 2
11   ip route add default via 192.168.2.1 dev
       eth1 table 2
12
13   # default route for the selection process
       of normal internet-traffic
14   ip route add default scope global nexthop
       via 192.168.1.1 dev eth0
```

– **Lancement du test à partir du client**

```
#/bin/bash
fichier=output.txt
date > $fichier

for j in 10 30 60; do

  for i in 1 2 3 4 5 6 7 8 9 10; do
  echo
     "==================================">>
     $fichier
  echo " S T A R T      T E S T" >> $fichier
  date >> $fichier
  echo "=================================="
     >> $fichier
  echo "Ping test eth0">> $fichier
  ping -c 1 192.168.1.101 >> $fichier
  echo "=================================="
     >> $fichier
  echo "Ping test eth1">> $fichier
  ping -c 1 192.168.2.101 >> $fichier
  echo "=================================="
     >> $fichier
  echo "Iperf #$i t=$j"
  echo "=================================="
     >> $fichier
  echo "Iperf $i t=$j">> $fichier
  iperf -c 192.168.1.101 -t $j>> $fichier
  echo "=================================="
     >> $fichier
  echo " S T O P        T E S T            "
     >> $fichier
  echo "=================================="
     >> $fichier
  sleep 2
  done
done
```

# Références


[1] S.M. Alamouti. A simple transmit diversity technique for wireless communications. *Selected Areas in Communications, IEEE Journal on*, 16(8) :1451–1458, 1998.

[2] J. Bergman, M. Ericson, D. Gerstenberger, B. Göransson, J. Peisa, and S. Wager. HSPA Evolution–Boosting the performance of mobile broadband access. *Ericsson Review no 1, 2008*, 2008.

[3] A. Bittau, M. Hamburg, M. Handley, D. Mazieres, and D. Boneh. The case for ubiquitous transport-level encryption. In *USENIX Security Symposium*, 2010.

[4] V.G. Cerf and R.E. Icahn. A protocol for packet network intercommunication. *ACM SIGCOMM Computer Communication Review*, 35(2) :71–82, 2005.

[5] K. Chebrolu, B. Raman, and R.R. Rao. A network layer approach to enable TCP over multiple interfaces. *Wireless Networks*, 11(5) :637–650, 2005.

[6] Costin Raiciu and Christoph Paasch and Sébastien Barré and Alan Ford and Michio Honda and Fabien Duchene and Olivier Bonaventure and Mark Handley. How hard can it be ? designing and implementing a deployable multipath tcp. In *USENIX Symposium of Networked Systems Design and Implementation (NSDI'12), San Jose (CA)*, 2012.

[7] Paul DeBeasi. Website, Jan 2010.

[8] S. Fu and M. Atiquzzaman. SCTP : State of the art in research, products, and technical challenges. *IEEE Communications Magazine*, 42(4) :64–76, 2004.

[9] Handley, M. and Raiciu, C. and Ford, A. and Barre, S. and Iyengar, J. RFC 6182, Architectural Guidelines for Multipath TCP Development. *Internet Engineering Task Force (IETF)*, March 2011.

[10] H. Holma and A. Toskala. *HSDPA/HSUPA for UMTS*. Wiley, 2006.

[11] M. Honda, Y. Nishida, C. Raiciu, A. Greenhalgh, M. Handley, and H. Tokuda. Is it still possible to extend tcp ? In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 181–194. ACM, 2011.

[12] Chung-Ming Huang and Ching-Hsien Tsai. Wimp-sctp : Multi-path transmission using stream control transmission protocol (sctp) in wireless networks. In *AINAW '07 : Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pages 209–214, Washington, DC, USA, 2007. IEEE Computer Society.

[13] ip networking lab (UCL). http://mptcp.info.ucl.ac.be/pmwiki.php?n=Users.ConfigureRouting, May 2012 (14 :45).

[14] ip networking lab (UCL). http://mptcp.info.ucl.ac.be/pmwiki.php?n=Users.AptRepository, April 2012 (14 :46).

[15] ISO ISO and IEC Standard. 7498-1. *Information Technology–Open Systems Interconnection–Basic reference model*, 1994.

[16] V. Jacobson, R. Braden, and D. Borman. Rfc 1323 : Tcp extensions for high performance, may 1992. *Obsoletes RFC1072, RFC1185 [12, 13]. Status : PROPOSED STANDARD*, 1992.

[17] T.E. Kolding, K.I. Pedersen, J. Wigard, F. Frederiksen, and P.E. Mogensen. High speed downlink packet access : WCDMA evolution. *IEEE Vehicular Technology Society News*, 50(1) :4–10, 2003.

[18] G. Malkin. RIP version 2, 1998.

[19] Mark Grayson, Kevin Shatzkamer and Klaas Wierenga. *Building the Mobile Internet*. Cisco Press, 2011.

[20] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, and T.C.P.S.A. Options. RFC 2018. *Internet Engineering Task Force (IETF)*, 1996.

[21] J. Moy. RFC2328 : OSPF Version 2. *RFC Editor United States*, 1998.

[22] J. Postel et al. RFC 768 : User datagram protocol. *Network Information Center, August*, 18, 1980.

[23] J. Postel et al. Transmission Control Protocol RFC 793, 1981.

[24] M.T. POWER and C. REDUCTION. High-Speed Packet Access Evolution in 3GPP Release 7. *IEEE Communications Magazine*, page 30, 2007.

[25] R. Prasad. An overview of third-generation wireless personal communications : a European perspective. *IEEE Personal Communications*, page 60, 1998.

[26] V. Ramamurthi, A. Reaz, D. Ghosal, and B. Mukherjee. Mimo-based rate adaptation to enhance tcp throughput over wireless fading channels. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5. IEEE, 2008.

[27] S. Sanayei and A. Nosratinia. Antenna selection in MIMO systems. *IEEE Communications Magazine*, 42(10) :68–73, 2004.

[28] M. Scharf and S. Kiesel. Head-of-line blocking in tcp and sctp : analysis and measurements. In *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE*, pages 1–5. IEEE, 2006.

[29] C.L. Schuba, I.V. Krsul, M.G. Kuhn, E.H. Spafford, A. Sundaram, and D. Zamboni. Analysis of a denial of service attack on tcp. In *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*, pages 208–223. IEEE, 1997.

[30] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod. Cross-layer design of ad hoc networks for real-time video streaming. *IEEE Wireless Communications*, 12(4) :59–65, 2005.

[31] D. Skordoulis, Q. Ni, H.H. Chen, A.P. Stephens, C. Liu, and A. Jamalipour. IEEE 802.11 n MAC frame aggregation mechanisms for next-generation high-throughput WLANs. *IEEE Wireless Communications*, 15(1) :40–47, 2008.

[32] R. Stewart. RFC 4960, Stream Control Transmission Protocol (SCTP), 2007.

[33] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. Iperf : The tcp/udp bandwidth measurement tool, 2005.

[34] D. Wischik, M. Handley, and M.B. Braun. The resource pooling principle. *ACM SIGCOMM Computer Communication Review*, 38(5) :47–52, 2008.

[35] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath tcp. *Proc. Usenix NSDI 2011*, 2011.

[36] G. Xylomenos, G.C. Polyzos, P. Mahonen, and M. Saaranen. Tcp performance issues over wireless links. *Communications Magazine, IEEE*, 39(4) :52–58, 2001.