ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE
CONCENTRATION RÉSEAUX DE TÉLÉCOMMUNICATIONS
M.Ing.

PAR
Zied NAAS

ÉTUDE DE L'ÉQUITÉ DU PROTOCOLE TCP DANS LES RÉSEAUX MULTI-SAUT

MONTRÉAL, LE 29 MARS 2010

PRÉSENTATION DU JURY

CE RAPPORT DE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Dziong Zbigniew, directeur de mémoire
Département Génie Électrique à l'École de technologie supérieure

M. Jean-Charles Grégoire, codirecteur de mémoire
Centre Énergie Matériaux et Télécom à l'Institut National de la Recherche Scientifique

M. Jean-Marc Robert, président du jury
Département Génie Électrique à l'École de technologie supérieure

M. Michel Kadoch, membre du jury
Département Génie Électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 24 MARS 2010

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens à remercier tout d'abord mon directeur de recherche Pr. Zbigniew Dziong pour l'opportunité qu'il m'a accordée en acceptant de travailler au sein de son équipe et pour ses conseils précieux.

Ma gratitude et ma profonde reconnaissance va également à mon co-directeur de recherche Pr. Jean-Charles Grégoire pour l'honneur qu'il m'a fait en acceptant d'être mon co-directeur.

Je le remercie de m'avoir accueilli dans son laboratoire à l'INRS, de son encadrement précieux et de sa patience, pour sa disponibilité, sa spontanéité et ses conseils précieux.

Je remercie les membres de jury, pour l'intérêt qu'ils ont manifesté pour évaluer ce travail.

Je remercie également tous les membres de l'équipe de CogMesh pour m'avoir accueilli au sein du groupe de recherche, pour les précieuses discussions durant les réunions hebdomadaires de ce groupe et pour le soutien et l'ambiance familiale qu'ils n'ont pas manqués d'apporter.

Mes remerciements aux membres de l'INRS-ÉMT, site Bonaventure pour leur accueil chaleureux.

J'aimerais, finalement, exprimer ma gratitude à tous ceux qui m'ont aidé afin de mener à bien ce travail.

ETUDE DE L'ÉQUITÉ DU PROTOCOLE TCP DANS LES RÉSEAUX MULTI-SAUT

Zied NAAS

RÉSUMÉ

Récemment, les réseaux maillés sans-fil sont de plus en plus utilisés et constituent une nouvelle technologie qui offre un accès Internet direct à haut débit. Ces réseaux sont des réseaux muti-sauts qui offrent plusieurs avantages tels que la facilité et le faible coût de déploiement, la formation automatique, la fiabilité, etc.

Étant donné l'importance du trafic TCP qui constitue 90% du trafic Internet, notre objectif dans ce mémoire est d'analyser et d'évaluer la performance du protocole TCP dans les réseaux multi-sauts et trouver les alternatives permettant de l'améliorer. Nous nous intéressons plus particulièrement à l'équité entre les flux TCP des différentes sources.

L'étude de l'équité TCP dans les réseaux multi-sauts est réalisée à travers un ensemble de simulations en utilisant le simulateur NS-2. Nous procédons aux simulations d'une topologie filaire et nous utilisons une nature de trafic pour abstraction aux réseaux maillées sans-fil. Une étude expérimentale à l'aide d'un banc de test nous permettra d'analyser l'équité dans les réseaux multi-sauts sans fil.

Dans les réseaux maillés sans-fil, le trafic ascendant est toujours vers la passerelle. À partir des résultats des simulations, nous constatons que la source la plus proche de la passerelle s'empare de la bande passante et ainsi on a un partage non équitable de la bande passante. Cette iniquité est dû au fait que les différentes sources ont des RTTs et des taux de pertes différents au niveau de la passerelle. D'autre part, nous constatons l'effet du délai externe au réseau maillé qui permet d'améliorer l'équité TCP. Nous proposons aussi d'utiliser DiffServ pour l'équilibrage de trafic et pour éviter que des sources subissent une famine en termes de bande passante. Nous constatons aussi que DiffServ permet d'atteindre un bon niveau d'équité dans le cadre de notre étude. Suite à l'étude expérimentale, nous avons eu presque le même comportement de l'équité TCP. De plus on a réussi à atteindre une équité parfaite en utilisant un mécanisme qui permet de définir et de contrôler au niveau de la destination la bande passante souhaitée pour chaque flux.

Mots clés: TCP, équité, réseaux multi-sauts, réseaux sans fil maillés

STUDY OF TCP PROTOCOL FAIRNESS FOR MULTI-HOP NETWORKS

Zied NAAS

ABSTRACT

Recently, wireless mesh networks (WMNs) have been widely used. They are considered as a new technology that offers last-mile broadband Internet Access. WMNs are multi-hop networks which present many advantages such as the simplicity and the low cost of the deployment, the auto-formation, the reliability, etc.

TCP is an important protocol and TCP traffic forms almost 90% of the whole Internet traffic. Therefore, we propose to analyse and evaluate the performance of TCP over multi-hop networks then we study the different ways to enhance its performance. The main topic that we deal with is the equity between TCP flows of the different sources.

To do so, simulations are conducted using NS-2 simulator. We use a wired topology with a specific traffic to simulate the wireless nature of WMNs. Through an experimental study using a testbed we analyze the equity for wireless multi-hop networks.

In WMNs, uplink traffic is destined to the gateway. According to the simulation results, we notice that the source nearest to the gateway takes advantage of most of the bandwidth so bandwidth inequity occurs. This inequity is related to the different RTTs and loss rates of the different sources. Furthermore, we also notice that the external delay to the mesh network can influence the TCP performance in terms of equity. To insure traffic balancing, we propose later to use DiffServ to avoid bandwidth starvation at the sources. We conclude that with DiffServ we reach a good equity level in the context of our study. The experimental results suggest that the equity has the same behaviour. We also reached a perfect equity using a destination-based mechanism that defines and controls the required bandwidth for each flow.

Keywords: TCP, fairness, multihop networks, wireless mesh networks

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

PDA	Personal Digital Assistant
WLAN	Wireless Local Area Network
MANET	Mobile Ad hoc Networking
WMN	Wireless Mesh Network
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	HyperText Transfer Protocol
FTP	File Transfer Protocol
SMTP	Simple Mail Transfer Protocol
RTT	Round-Trip Time
RED	Random Early Detection
DiffServ	Differentiated Services
MP	Mesh Point
MPP	Mesh Portal Point
MAP	Mesh Access Point
AP	Access Point
QoS	Quality of Service
MIMO	Multiple-Input Multiple-Output
SNR	Signal to Noise Ratio
UWB	Ultra-Wideband
CSMA/CA	Carrier-Sense Multiple Access / Collision Avoidance
RIP	Routing Information Protocol
BGP	Border Gateway Protocol
EGP	Exterior Gateway Protocol
OSPF	Open Shortest Path First
ZRP	Zone Routing Protocol
AODV	Ad hoc On Demand Distance Vector

OLSR	Optimized Link State Routing
ACK	Acknowledgement
Cwnd	Contention Window
Ssthresh	Slow start Threshold
MSS	Maximum Segment Size
DCF	Distributed Coordination Function
PCF	Point Coordination Function
DIFS	Distributed Inter Frame Space
RTS	Ready To Send
CTS	Clear To Send
AIMD	Additive Increase Multiple Decrease
TOS	Type Of Service
DSCP	Differentiated Services Code Point
PHB	Per Hop Behavior
WRED	Weighted RED
RIO	RED In/Out
PQ	Priority Queuing
WRR	Weighted Round Robin
TSW2CM	Time Sliding Window Two Color Marker
CIR	Committed Information Rate

INTRODUCTION

Avec l'augmentation de l'utilisation d'équipements mobiles tels que les ordinateurs portables, les PDAs, et les téléphones cellulaires, il y a de plus en plus d'accès sans fil à l'Internet. L'IEEE a développé un standard pour les réseaux locaux sans-fil (WLAN) qui est l'IEEE 802.11. Depuis l'apparition de ce standard, les équipements dotés de cette technologie ont connu une importante croissance de vente à travers le monde. Les WLANs 802.11 sont devenus une solution intéressante pour les organisations qui profitent des avantages de mobilité, de la facilité d'installation et d'utilisation et surtout pour des raisons économiques puisque les WLANs éliminent les coûts de câblage. Les WLANs offrent alors un accès Internet en un seul saut, à travers le point d'accès qui est relié par câblage à l'Internet. Cependant, un WLAN ne permet pas une couverture très large. Pour augmenter la couverture, il faut ajouter des points d'accès et des câbles pour les relier à l'Internet. Cette solution devient alors plus coûteuse et son intérêt diminue. Une alternative est de créer alors un réseau sans fil multi-sauts. Les réseaux ad-hoc sont un exemple de réseaux sans-fil multi-sauts sans infrastructure, et utilisés essentiellement pour les applications civiles, tactiques et militaires. D'autre part, plusieurs technologies de réseaux sans-fils multi-sauts ont vu le jour comme les réseaux maillés sans fils, les réseaux de capteurs, etc.

Le réseau maillé sans fil (WMN) est une technologie émergente et prometteuse. Elle attire de plus en plus l'attention des fournisseurs de services, des entreprises, des municipalités, etc., pour établir un service d'accès sans fil à haut débit, robuste, efficace, à moindre coût et à large couverture. Il s'agit d'un réseau sans fil multi-sauts. Au lieu d'ajouter plusieurs points d'accès dont chacun est relié par fil à l'Internet pour augmenter la couverture, dans les WMNs, il n'y a que les passerelles qui seront reliées à l'Internet à travers des câbles. Les WMNs consistent en un ensemble de clients sans fil et un ensemble de routeurs sans fil fixes dont les points d'accès. Les routeurs sont liés l'un à l'autre sans fils pour former un large réseau sans fil multi-sauts. Les clients se connectent aux points d'accès avec la méthode traditionnelle 802.11.

Étant donné l'intérêt continu et croissant aux WMNs, un groupe de travail IEEE a été créé dans le but de préparer un standard pour cette technologie, désigné comme IEEE 802.11s.
Le trafic TCP constitue aujourd'hui le trafic majoritaire sur les réseaux IP. Par dessus le protocole TCP fonctionnent une multitude d'applications telles que les applications Web HTTP, les applications de mail SMTP, les applications FTP, etc.

Dans ce mémoire notre objectif est d'analyser l'équité de TCP dans les réseaux multi-sauts avec application aux réseaux maillés sans fil multi-radios à base de 802.11, et l'améliorer par des solutions simples et réalisables *sans apporter aucune modification au protocole TCP*. Le trafic ascendant dans les réseaux WMNs est dirigé vers les passerelles. Ce trafic va engendrer une congestion au niveau des liens reliant les passerelles à l'Internet. L'équité est le fait de partager la bande passante équitablement entre les différents flux. Dans les réseaux multi-sauts, et donc les WMNs, les nœuds sont situés à des nombres de sauts différents de la passerelle et ont ainsi des temps aller-retour (RTT) différents. Le débit TCP est sensible aux pertes et aux RTT, et les flux des différents nœuds peuvent alors avoir des débits différents. De ce fait on peut avoir un partage non équitable de la bande passante. L'équité est l'une des exigences importantes dans les WMNs, et l'une des exigences des utilisateurs.

Nous allons alors analyser comment se manifeste l'équité dans les réseaux multi-sauts, quels sont les paramètres qui influent sur l'équité et quelles sont les alternatives pour l'améliorer. Notre méthodologie est d'étudier en première partie l'équité TCP dans les réseaux multi-sauts d'une manière générale à travers des simulations avec le simulateur NS-2 sur une topologie filaire. La topologie filaire simulée est une abstraction des réseaux maillés sans fil. En deuxième partie nous allons procéder à une analyse de l'équité TCP dans les réseaux sans fil multi-sauts et valider les resultats obtenus par les simulations à travers un ensemble d'expériences réalisées sur un banc de test.

La suite du mémoire sera comme suit : le chapitre 1 introduit et décrit l'état de l'art des réseaux maillés sans fil. Le chapitre 2 présente le protocole TCP, les propriétés de 802.11 et

quels sont les problèmes rencontrés par le TCP standard, qui a été conçu pour les réseaux filaires, dans les réseaux sans-fil. Le chapitre 3 sera présenté en deux parties : la première partie sera consacrée à l'analyse de TCP à travers des simulations utilisant le mécanisme de gestion de file d'attente DropTail au niveau de la passerelle. La deuxième partie présente la même étude mais en utilisant le mécanisme RED. Dans le chapitre 4 on propose d'appliquer les mécanismes de DiffServ et analyser la performance TCP à travers les simulations dans le cas d'un réseau surdimensionné et dans le cas d'un réseau sous-dimensionné. Le chapitre 5 sera consacré pour l'étude expérimentale des réseaux sans fil multi-sauts réalisée sur un banc de test.

CHAPITRE 1

LES RÉSEAUX MAILLÉS SANS FIL

1.1 Définitions

Le mot clé pour ce type de réseaux est « maillé » (Mesh). Un réseau maillé est tout simplement un réseau où chaque nœud peut communiquer, directement ou non, avec n'importe quel autre nœud, et où la perte d'un lien ne compromet pas (complètement ?) la connectivité. Un réseau maillé peut être défini comme un réseau qui, pour un nœud donné, fournit au minimum deux chemins différents vers n'importe quel autre nœud du réseau. Les données circulent alors de nœud à nœud (hop by hop) jusqu'à atteindre le nœud de destination. Un réseau sans fils maillé WMN, est un cas particulier de réseau maillé où pour une transmission entre deux nœuds, les autres nœuds fonctionnent comme des routeurs et effectuent les fonctions de relayage, via des liens sans-fil.

Après le succès qu'ont connu les réseaux 802.11 avec infrastructure et les réseaux ad-hoc au cours des dernières années, les WMN se présentent comme une nouvelle architecture réseau qui permet de combler les faiblesses et les limites de ses prédécesseurs en améliorant les services tout en minimisant les coûts. Les WMN permettent alors de fournir une connectivité à large bande avec un déploiement facile et à faible coût.

Les WMN permettent d'offrir une connectivité aux utilisateurs en tout temps et en tout lieu grâce à une dorsale réseau (*backhaul*) constituée de routeurs sans fils qui ont pour fonction de relayer le trafic jusqu'à une passerelle qui est connectée d'une manière filaire aux réseaux extérieurs et essentiellement Internet et vice versa. Au contraire des réseaux cellulaires et les WLAN, les WMN fournissent alors à travers cette topologie maillée aux clients mobiles une

infrastructure décentralisée qui constitue comme on le verra plus tard l'un des avantages des WMN, à savoir la résistance aux pannes.

Le caractère multi-sauts des WMN permet entre autres d'augmenter la couverture du réseau.

1.2 Les composantes WMN

Avant de voir les architectures des WMN, il est essentiel d'identifier les différentes composantes de ces réseaux. Notons que plusieurs dénominations sont présentes dans la littérature et on a choisi celle proposée par IEEE 802.11s :

- MP (Mesh point): ils forment la dorsale du réseau. Ce sont des routeurs sans fils qui ont la capacité de router et de relayer le trafic d'un MP à un autre jusqu'à la passerelle. Les MP sont généralement fixes et n'ont pas de contraintes de consommation d'énergie (Akyildiz, Wang et Wang, 2005) , (Ekram Hossain, 2008).
- MAP (Mesh Access Point) : c'est un MP qui joue parallèlement le rôle d'un point d'accès. Il fournit l'accès au réseau pour les stations ou les clients mobiles (Akyildiz, Wang et Wang, 2005), (Ekram Hossain, 2008).
- MPP (Mesh Portal Point) : un MP qui joue aussi le rôle de passerelle vers d'autres types de réseaux comme Wi-Max. Il est généralement connecté au réseau filaire afin de fournir aux clients une connectivité Internet en tout temps et en tout lieu (Akyildiz, Wang et Wang, 2005), (Ekram Hossain, 2008).
- STA (Stations) : il s'agit du client ou l'utilisateur. Les STA ne participent pas au routage et les services Mesh. Ils sont mobiles et ils communiquent entre eux à travers leur station de base. Les STA peuvent être un ordinateur portable, un PDA, etc. (Akyildiz, Wang et Wang, 2005), (Ekram Hossain, 2008).

1.3 Les architectures WMN

Un WMN peut avoir l'une des trois architectures suivantes, le choix de l'une ou de l'autre dépend du but d'utilisation. Il est à noter que la dénomination varie dans la littérature.

1.3.1 WMN avec infrastructure ou hiérarchique

Ce type d'architecture est caractérisé par plusieurs niveaux. Les clients constituent le niveau le plus bas. Ce sont l'origine ou la destination du trafic et ils ne participent pas au routage ou à l'acheminement (relais). Les MPs forment la dorsale et ont pour fonction le relais du trafic. Le troisième niveau est constitué des passerelles MPP qui permettent l'accès à l'Internet ou bien l'intégration des WMN avec les réseaux sans fils existants. Cette architecture qui est nommée aussi WMN fixe permet d'étendre la couverture réseau pour les clients. Les MP peuvent être installés à l'extérieur, dans les rues, dans les lieux publiques, etc., tout comme ils peuvent être installés à l'intérieur dans des entreprises, par exemple, et ainsi le client qui est mobile peut toujours avoir une connectivité. Cette architecture est la plus utilisée et elle est illustrée par la Figure 1.1.

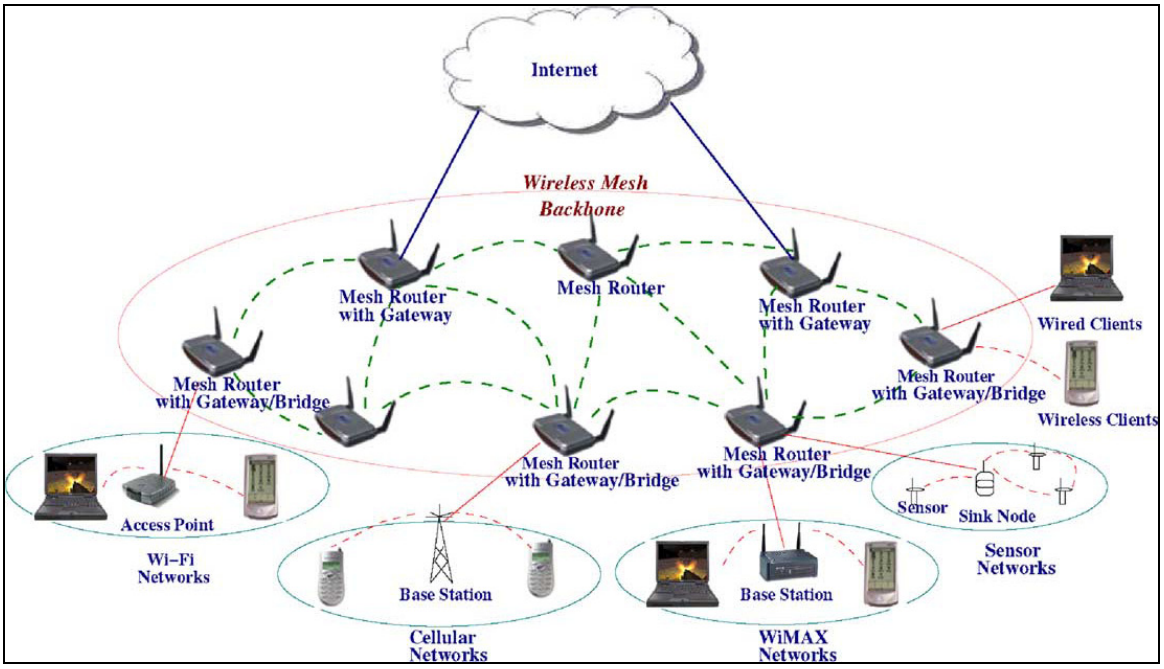

Figure 1.1 WMN Hiérarchique.
Tirée de Akyildiz, Wang et Wang (2005, p. 4)

1.3.2 WMN plats ou mobiles

Ce type d'architecture est nommé aussi WMN ad-hoc car elle est très similaire aux réseaux ad-hoc. Tous les équipements sont sur le même niveau et ils peuvent être à la fois des clients et des routeurs. Ce type d'architecture fournit un réseau point à point entre les clients (Akyildiz, Wang et Wang, 2005). Les WMN ne nécessitent pas les MAP et les MPP qui sont nécessaires dans le cas où il y a un besoin de joindre des réseaux externes comme l'Internet. Ce type d'architecture est idéal pour les applications distribuées. La topologie du réseau change fréquemment en raison de la mobilité des équipements, ce qui exige une mise à jour fréquente des tables de routage et résulte en une charge importante. Cette architecture est utilisée pour plusieurs cas d'usage comme par exemple pour les services de sécurité publique, les applications de transport, etc. (Bing, 2008). La Figure 1.2 illustre ce type d'architecture.

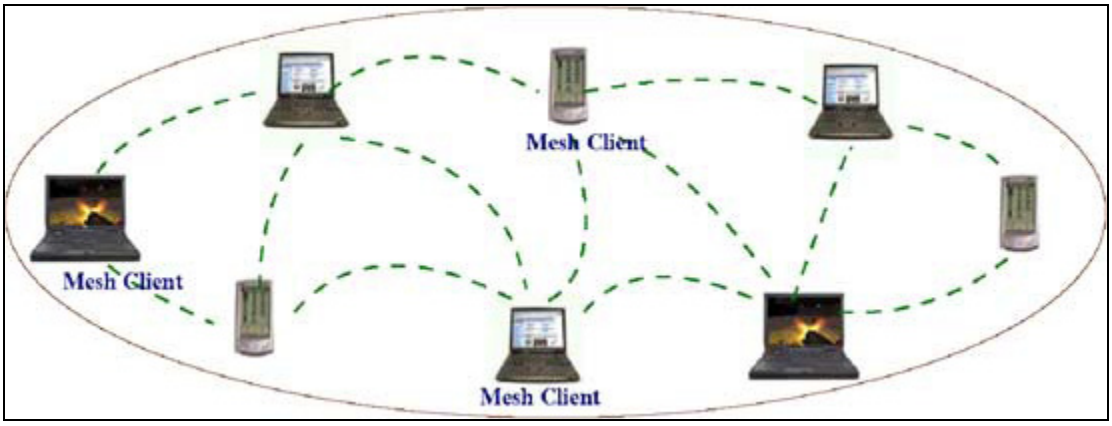

Figure 1.2 WMN clients.
Tirée de Akyildiz, Wang et Wang (2005, p. 4)

1.3.3 WMN Hybride

Cette architecture est une combinaison entre les deux architectures précédentes. Les clients peuvent communiquer directement entre eux et ils sont munis des fonctions de routage pour passer le trafic d'un client à un autre, et utilise l'infrastructure pour avoir une connectivité aux réseaux externes (donc Internet). Cette architecture devient de plus en plus importante pour le développement des WMN (Akyildiz, Wang et Wang, 2005). La Figure 1.3 illustre cette architecture.

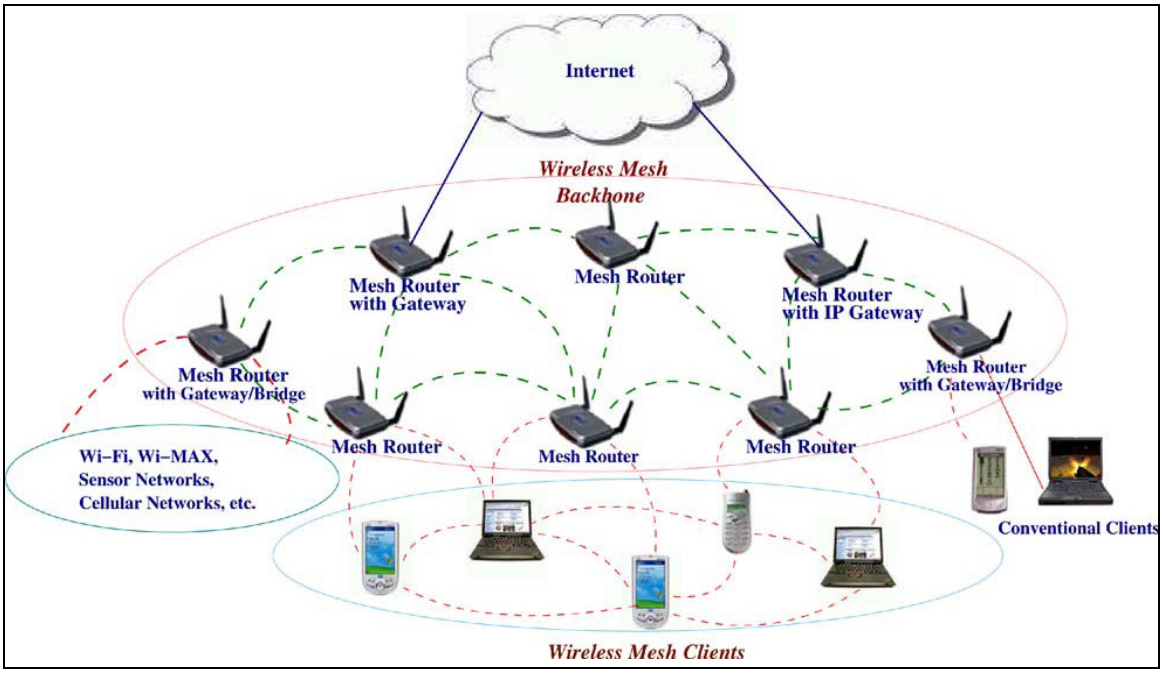

Figure 1.3 WMN Hybride.
Tirée de Akyildiz, Wang et Wang (2005, p. 5)

1.4 De Wi-Fi vers les WMNs

La technologie Wi-Fi basée sur le standard IEEE 802.11 a connu un grand succès depuis son apparition et a été la base pour la mise en place des réseaux locaux sans fils (WLAN). Les WLANs sont des architectures avec infrastructure où l'accès est réalisé à travers les points

d'accès (*Hot-Spot*) qui sont connectés directement aux réseaux filaires. La popularité de Wi-Fi n'a fait que croître, et cette technologie est déployée partout dans les lieux résidentiels, les lieux publics, les entreprises, etc. Étant donné les limites de la zone de couverture des points d'accès, étendre cette zone et augmenter la couverture réseau passe par l'ajout et l'installation de points d'accès supplémentaires, ce qui implique la connexion câblée des points d'accès entre eux engendrant ainsi des coûts et une complexité élevée. Ainsi, interconnecter les points d'accès d'une manière sans fil va réduire énormément la complexité et les coûts de l'étalement de la zone de couverture. Il est évident aussi que la réussite du standard 802.11 lui a permis d'être la base des WMNs.

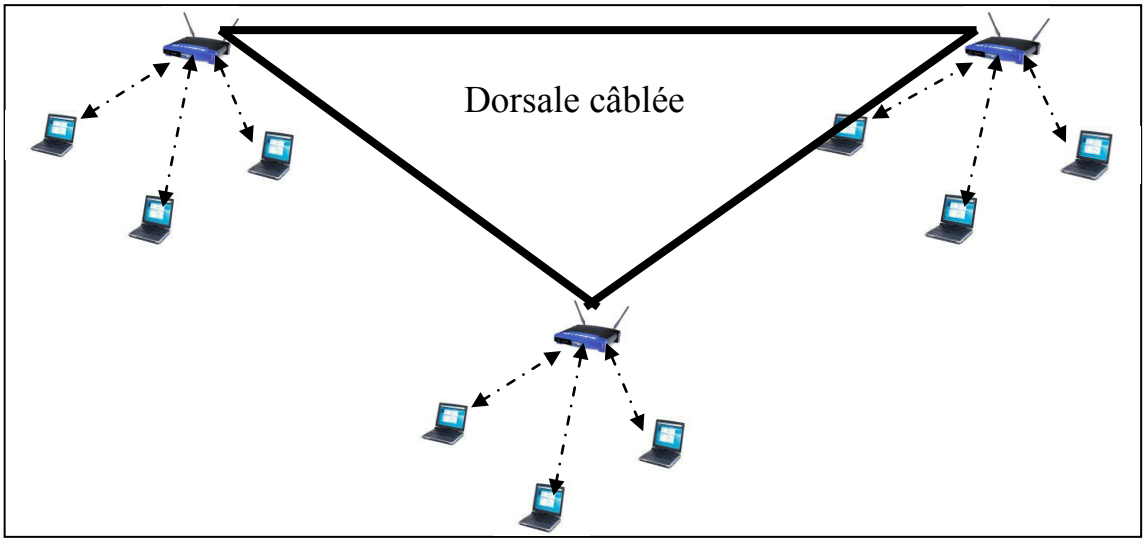

Figure 1.4 Interconnexion câblée de points d'accès.

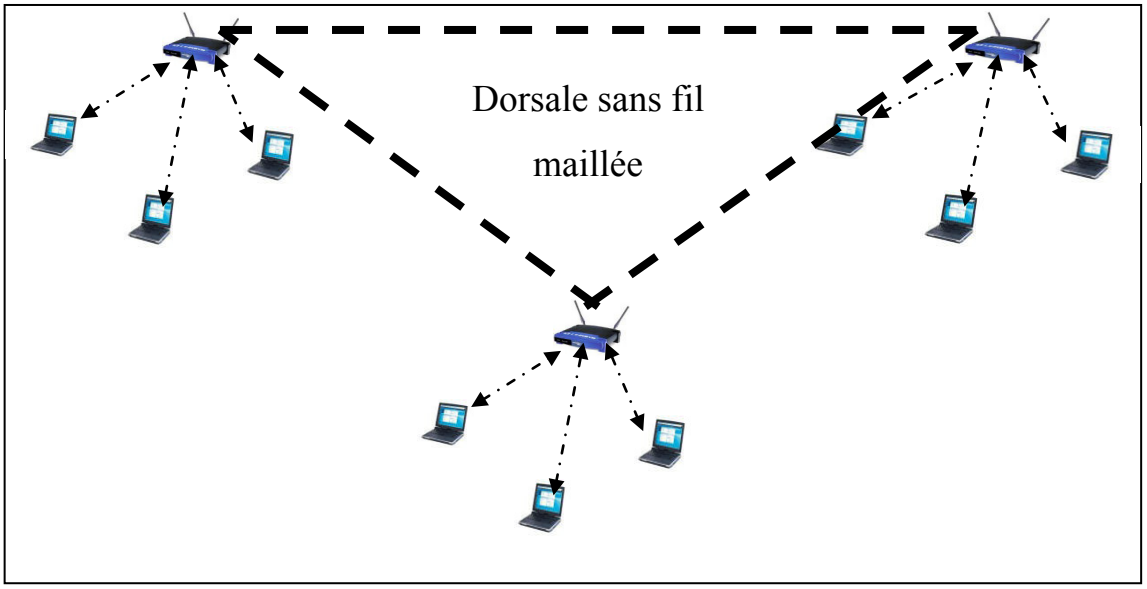

Figure 1.5 Interconnexion sans fil de points d'accès.

1.5 Comparaison entre les WMN et les réseaux ad-hoc mobiles (MANET)

Comme déjà mentionné, il y a une certaine similarité entre les WMN et les réseaux ad-hoc : les WMN utilisent un relais multi-sauts de proche en proche comme dans le cas des réseaux ad-hoc. Il est essentiel alors de connaître les différences entre ces deux technologies :

- Mobilité : la principale différence entre ces deux technologies est le degré de mobilité des nœuds. Les réseaux ad-hoc sont des réseaux potentiellement mobiles, dont la topologie varie et change dynamiquement. Dans les WMN, les MP sont fixes et ainsi la topologie ne change pas fréquemment.
- Énergie : si dans les réseaux ad-hoc les nœuds ont une contrainte d'énergie, les MP sont dépourvus de cette contrainte et par suite les algorithmes de routage relatifs aux WMN sont conçus sans prendre en compte la minimisation de la consommation d'énergie à la différence des algorithmes conçus pour les réseaux ad-hoc.
- Performance de routage : comme on le verra plus tard, bien que les technologies soient similaires, les algorithmes de routage conçus pour les réseaux ad-hoc ne performent pas bien quand ils sont appliqués dans les WMN.

- Nature du trafic : dans les réseaux ad-hoc, le trafic est point à point entre deux nœuds passant par les nœuds intermédiaires si c'est nécessaire, alors que le trafic dans les WMN est principalement entre les MP et les MPP.
- Architecture : les réseaux ad-hoc sont conçus pour les communications au sein de petits groupes d'utilisateurs, ce qui permet d'avoir une architecture plate sans infrastructure, sans possibilité d'accéder à d'autres réseaux qui ont une technologie radio différente. Les WMN ont cependant une architecture hiérarchique avec une possibilité d'accès à des réseaux extérieurs grâce aux MPP.
- Radio-multiple : dans les WMN les MP peuvent être équipé de plusieurs radios ce qui permet d'augmenter la capacité et le débit du réseau.
- Scénarios d'application : au contraire des réseaux ad-hoc qui ont été conçus pour les applications militaires et tactiques, les WMN peuvent être utilisés, comme on le verra par la suite, pour plusieurs scénarios d'applications, civiles et militaires.

1.6 Caractéristiques et avantages des réseaux maillés

Les WMN ont plusieurs avantages par rapport aux réseaux sans fil traditionnels comme le Wi-Fi et les réseaux ad-hoc. Les WMN sont des réseaux qui ont la capacité d'auto-formation, auto-configuration et auto-cicatrisation (self-healing). Nous résumons ci-dessous les principales caractéristiques et avantages des WMN :

- Formation automatique : les réseaux maillés sans fil sont conçus pour être formés automatiquement. Un nœud MP exécute une procédure de découverte de réseaux présents. Si aucun réseau n'est présent, le MP peut initier un nouveau réseau. Pour la procédure de découverte, deux approches existent : une approche passive basée sur la réception de messages balise, qui se base sur l'écoute du canal mais engendre un temps de découverte plus long que la deuxième approche, active, qui est basée sur l'envoi de messages de sondage (sondage du canal).
- Auto-cicatrisation et fiabilité : au contraire des WLANs où le point d'accès constitue le point d'échouement, dans le cas des WMNs, en cas de défaillance d'un lien, les paquets

pourront toujours atteindre leur destination via un autre lien. La redondance de chemins assure une tolérance aux pannes.

- Configuration automatique : Dans les réseaux maillés on peut se passer d'administrateur de réseau puisque chaque nœud effectue la découverte de ses voisins et par suite de tous les chemins vers tous les autres nœuds.
- Coût d'installation moins élevé : les réseaux Wifi sont des réseaux sans fil avec infrastructure. Les nœuds du réseau sont connectés au point d'accès « AP » qui est relié au réseau filaire. La couverture de l'AP étant limitée, pour assurer une couverture totale dans une aire très vaste, il faut déployer un grand nombre d'AP ce qui mène à un coût très élevé pour câbler tous les AP. Les réseaux maillés sans-fil sont par contre plus faciles et moins coûteux à installer puisque seulement quelques nœuds seront connectés au réseau filaire.
- Surface de couverture plus large : Le caractère multi-saut des WMN permet et assure des communications sur de plus larges distances et augmente la distance de couverture. Cependant, augmenter la zone de couverture pour servir plus d'utilisateurs conduit à un problème d'extensibilité du réseau (Akyildiz, Wang et Wang, 2005).
- Facilité de déploiement, d'extension et de maintenance : l'absence de câblage, l'organisation, la configuration, et la cicatrisation automatiques font en sorte que ces réseaux sont faciles à mettre en place, faciles à étendre et faciles à maintenir et ne nécessitent pas trop d'interventions humaines.
- Ces réseaux n'ont pas de contraintes d'énergie pour les nœuds MP.
- Intégration : l'existence de passerelles dans les WMN permet l'intégration de ces réseaux avec d'autres réseaux sans fils comme les réseaux de capteurs et les réseaux Wi-Max.

1.7 Scénarios d'application

Les réseaux WMN sont de nos jours de plus en plus sollicités. Plusieurs applications ont été déployées à la base des WMN. Nous présentons ci-dessous quelques domaines d'application pour les WMN :

- Accès Internet : récemment, plusieurs fournisseurs de services Internet ont mis en place des WMN pour permettre un accès sans fil à large distance dans les villes, les zones rurales, etc. Cette technologie leur permet de couvrir des zones plus larges à un coût moins élevé que les Wi-Fi traditionnels tout en assurant de bonnes performances. Les WMN sont déployés aussi dans les lieux de divertissement, dans les hôtels, les aéroports, dans les campus etc. Parmi les cas de déploiement, TROPOS (*TROPOS Networks*) a réalisé un réseau municipal dans la ville Chaska de l'état de Minnesota à l'aide de quelques 250 nœuds. D'autre part, Nortel (*NORTEL*) a été choisie par la ville de Taipei pour déployer un réseau municipal en 2005 à l'aide de 10 000 nœuds dans une surface de 272 Km^2.
- Sécurité publique : les réseaux WMN constituent une solution prometteuse pour combler les limites des solutions utilisées par les services de sécurité publique comme la police, la protection civile et les services d'urgence. Ce genre d'applications permet aux agents de la sécurité publique d'avoir un accès permanent aux données qui leur sont utiles au cours de leurs opérations. Comme exemple on peut citer le cas où il y a une intervention pour un incendie, et le pompier peut accéder du site d'intervention même aux plans du lieu. Des produits commerciaux ont été développés par plusieurs manufacturiers pour ce genre d'applications. On peut citer BELAIR Networks (*BelAir Networks*), Strix Systems (*Strix SYSTEMS*), TROPOS (*TROPOS Networks*). Le routeur BELAIR 100 (*BelAir Networks*) par exemple est un routeur qui peut être utilisé pour ce genre d'application. Il est équipé pour cela de deux modules radio PSM1 et PSM2 pour opérer dans la bande de fréquence 4.9 GHZ sous la licence de la sécurité publique. La compagnie de Strix Systems (*Strix SYSTEMS*) de sa part à été choisie pour fournir une solution pour les services de sécurité publique lors des jeux olympiques 2008 de Beijing.

- Systèmes de transport : depuis l'apparition des WMN, plusieurs compagnies de transport public et des agences gouvernementales se sont intéressées aux solutions apportées par ce type de réseaux pour des systèmes de transport intelligent. Les routeurs extérieurs installés sur les routes et les autoroutes permettent aux différents utilisateurs dans leurs véhicules, y compris les agents de sécurité publique, de bénéficier de plusieurs informations échangées d'un véhicule à un autre. Ces informations vont les aider à prendre des décisions. Le trafic routier, les conditions météo diffusées font partie de ces informations échangées en vidéo. Strix Systems a réalisé le projet de mettre en place un système de transport intelligent en Corée sur l'autoroute qui relie Pangyo et Osan sur une distance de 31 Km. La solution permet d'assurer des connections fluides de bout en bout avec un débit de 8 Mbps pour des applications vidéo, voix sur IP et multimédia. Les véhicules peuvent rouler à une vitesse allant jusqu'à 100 Km/h (*Strix SYSTEMS*). Le routeur BELAIR 100M est un modèle qui peut être monté dans des bus, des trains, des voitures de police et tout autre type de véhicule. Il est intégré uniformément avec l'infrastructure BELAIR déployée afin de fournir une connectivité mobile et transparente à large bande pour les applications vidéo, voix et données. Les véhicules peuvent aller jusqu'à une vitesse de 240 Km/h (*BelAir Networks*).

Outre ces scénarios d'application qu'on vient de citer, les WMN sont utilisés aussi dans les cas suivant (cette liste n'est pas exhaustive):

- Vidéosurveillance,
- Automatisation des locaux : pour le contrôle à distance des équipements électriques dans une maison ou dans un immeuble,
- Les opérations militaires et tactiques,
- Les réseaux de capteurs (Sensor Networks),
- Systèmes de santé et médicaux.

1.8 Les défis et les problématiques des WMN

Malgré les avantages apportés par les WMN et leur popularité qui augmente de plus en plus, les WMN connaissent plusieurs défis et problématiques à résoudre afin d'améliorer et optimiser leur performance. Parmi les facteurs qui influencent la performance des WMN on trouve : les techniques radios utilisées, l'extensibilité (Scalability), la QoS, la sécurité, l'interopérabilité et l'intégration de réseaux hétérogènes, la mobilité, la connectivité et l'auto-configuration, et les outils de gestion de réseau. La capacité des WMN est d'autre part influée par des facteurs comme l'architecture et la topologie du réseau, le modèle de trafic, le placement des nœuds, le degré de mobilité, etc. (Akyildiz, Wang et Wang, 2005). La pile de protocole traditionnelle peut être améliorée et adaptée aux WMN. Plusieurs travaux de recherche sont réalisés au niveau des différentes couches. Ci-dessous les problématiques relatives aux trois couches, physique, liaison et réseau. La couche transport et plus particulièrement le protocole TCP fera l'objet du chapitre suivant.

1.8.1 La couche physique

Les WMN comptent fournir un accès à large distance avec un débit considérable. Comme un grand nombre de nœuds peuvent participer à un réseau WMN, les interférences augmentent substantiellement. L'objectif principal des technologies à utiliser au niveau de cette couche est de maximiser le taux de transfert de données, maximiser l'efficacité spectrale et d'avoir la capacité d'opérer en présence d'interférences et les minimiser. Parmi les alternatives on trouve :

- Le type d'antenne à utiliser : l'utilisation d'antennes omnidirectionnelles ne permet pas de concentrer le signal vers une destination bien déterminée et conduit à une réduction du débit à cause des interférences. Les antennes directionnelles permettent par contre une meilleure réutilisation de fréquences (Ekram Hossain, 2008).
- Le nombre de radios utilisées : dans le cas où on équipe une seule radio dans un nœud maillé (single radio), cette même radio est utilisée pour l'accès et pour relayer les

données dans le réseau. La capacité du canal est alors divisée. La deuxième alternative est d'utiliser des nœuds à deux radios (dual radio). Cette alternative améliore la performance puisque les deux radios vont opérer sur deux fréquences différentes sans interférences. Une radio sera utilisée pour l'accès et l'autre pour le relayage. Bien que cette alternative soit meilleure, elle a une limite : puisque le réseau est partagé et constitué de plusieurs nœuds, une seule radio pour le relayage conduit à une latence suite aux problèmes de contentions du standard 802.11. La dernière alternative consiste à utiliser des radios multiples (multi-radio) au sein du même nœud. Ainsi, une radio sera dédiée à l'accès, et les autres seront dédiées au relayage. Chaque nœud communique avec ses voisins d'une manière parallèle sur des fréquences différentes. Le modèle BELAIR 300 (*BelAir Networks*), par exemple, est équipé de six interfaces radios.

- De nouvelles technologies radios : récemment de nouvelles technologies ont été développées pour améliorer la capacité des WMN. La technologie MIMO (multiple-input multiple-output) figure parmi ces technologies. Les systèmes MIMO exploitent la diversité d'antennes et le multiplexage spatial. Ces systèmes peuvent avoir des complexités différentes. De ce fait, les systèmes à complexité réduite sont préférables pour les clients tandis que les autres peuvent s'appliquer aux MP (Ekram Hossain, 2008). MIMO permet aussi de décoder les signaux ayant un faible SNR (Signal to Noise Ratio). D'autre part les antennes intelligentes permettent d'améliorer la capacité des WMN. L'idée d'utiliser les antennes intelligentes est d'exploiter la capacité de formation de faisceaux (*beamforming*) par les différentes antennes émettrices/réceptrices (Ekram Hossain, 2008). Le taux de transfert peut être amélioré par l'utilisation de l'UWB (Ultra-Wide-Band). L'UWB ne peut par contre être exploité que pour les communications à courte distance.
- La radio cognitive : selon la FCC (Federal Communications Commission), presque 70 % du spectre alloué n'est pas utilisé. L'idée est alors d'exploiter ces fréquences pendant l'absence des utilisateurs primaires (les utilisateurs propriétaires de la licence de la bande de fréquence). La radio-cognitive constitue alors une bonne alternative à exploiter pour les WMN.

1.8.2 La couche MAC

La couche MAC standard a été conçue pour les réseaux 802.11 qui sont des réseaux à un saut. Pour cette raison, elle n'est pas bien adaptée pour les réseaux multi-sauts, dont les WMN. L'utilisation de DCF (Distributed Coordination Function) et CSMA/CA peuvent causer un problème de famine pour certains nœuds à cause de la contention. Cette contention et le taux élevé des collisions dues aux problèmes de nœuds cachés et exposés influent sur la performance augmentent la latence. D'autre part, la conception de la couche MAC prend en considération la contrainte d'énergie des nœuds. Cette contrainte n'existe pas dans les réseaux WMN. La couche MAC alors doit être améliorée et adaptée pour opérer dans des systèmes multi-radio et multi-canal (MR-WMN), et d'autre part pour supporter les différentes technologies de la couche physique déjà citées comme le MIMO. D'autre part, l'amélioration de la couche MAC doit passer par l'échange d'information avec les autres couches (Cross-layer). Les mécanismes de QoS de la couche MAC doivent être vérifiés et s'il le faut améliorés. Dans ce contexte le standard préparé par IEEE 802.11s présente des améliorations sur la base de 802.11e pour être compatible avec les WMN.

1.8.3 La couche réseau

La performance des WMN repose énormément sur le routage. L'algorithme de routage doit prendre en compte l'extensibilité, la découverte rapide des routes et des pannes, doit supporter la mobilité et assurer la flexibilité et la QoS. Plusieurs algorithmes de routage ont été conçus pour les réseaux ad-hoc. Le développement de ces protocoles prend en considération la mobilité des nœuds et les contraintes de consommation d'énergie. Cependant, les nœuds dans les WMN ont une mobilité réduite et pas de contrainte d'énergie. Ces protocoles peuvent être utilisés pour les WMN : LUCEOR (*LUCEOR*) utilise le protocole OLSR, Microsoft (*Microsoft*) utilise DSR, FIRETIDE (*Firetide*) utilise TBRPF et d'autres compagnies utilisent AODV ; cependant, ces protocoles ont un niveau de

performance qui n'est pas idéal pour les WMN et des protocoles mieux adaptés doivent être alors développés. Le protocole à développer doit maximiser le débit, minimiser le taux de perte, minimiser la latence et la gigue, prendre en compte la qualité des liens et assurer l'extensibilité. Trois grandes familles de protocoles ont été conçues et développées essentiellement pour les réseaux ad-hoc : réactives, proactives et hybrides :

Pour les protocoles de routage proactifs, comme pour les protocoles de routage dans les réseaux filaires, les deux méthodes utilisées sont :

- Méthode du vecteur de distance : il s'agit de la méthode la plus ancienne. Elle est utilisée par les protocoles RIP, BGP et EGP. Le principe de cette méthode est que chaque nœud ait connaissance de la distance vers les autres nœuds et leur distribue cette information. Cette information n'est stockée que si elle soit meilleure que l'existante. Cette solution est simple mais elle est *lente à converger* d'une part et d'autre part le fait de transmettre toute la table de routage à chaque fois cause une dégradation de performance.
- Méthode à état de lien : cette méthode est la base de plusieurs algorithmes de routage tel qu'OSPF. Le principe est de distribuer, par une inondation contrôlée (inondation qui veille à minimiser l'excès d'utilisation de la bande passante), les LSU (Link State Units), soit la topologie, à tous les nœuds. Chacun de ses derniers calcule sa propre table de routage en se basant sur l'algorithme du plus court chemin de Dijkstra. Il n'y a plus besoin alors de transmettre toute la table de routage. Cette méthode permet une convergence plus rapide que celle du vecteur de distance.

Dans cette famille de protocoles, chaque nœud maintient une table de routage qui contient les informations de routage de tous les nœuds du réseau. Chaque nœud met à jour périodiquement sa table. Les routes sont immédiatement disponibles lors de la demande. Les protocoles à état de lien fonctionnent efficacement dans des réseaux de petite taille, mais quand la taille augmente ce sera plus difficile pour les nœuds de garder les informations de tous les autres nœuds et, dans le cas de changements dynamiques très fréquents de la topologie du réseau, le nombre de messages de contrôle augmente, ce qui va influer sur la bande passante.

Les protocoles de routage réactifs sont surtout utiles quand il y a une mobilité des nœuds. Ils ont été conçus pour réduire la charge qui résulte de l'usage des protocoles proactifs, qui doivent garder à jour l'information relative à la route active. Ils créent une route entre une source et une destination seulement lorsque la source décide d'envoyer des paquets. Ils utilisent la méthode d'inondation du réseau pour découvrir les routes dont ils ont besoin : pas de table de routage maintenue, aucun échange de paquets de contrôle pour construire des tables de routages et une consommation d'une grande quantité de ressources pour découvrir une simple route entre deux points du réseau. Les protocoles réactifs modernes ont un mécanisme de découverte de route et un mécanisme de maintenance qui enregistre les informations de routage jusqu'au moment où la source n'a plus besoin d'elles ou les routes deviennent invalides. Ceci implique un échange de messages de contrôle, échange qui influe sur la bande passante et augmente la charge. Les protocoles doivent alors n'envoyer les messages de mises à jour que pour les nœuds concernés, et les nœuds qui ne sont pas liés à une route n'ont pas besoin de garder des informations concernant cette route : ils n'ont que les informations liées à leurs routes actives. Il n'y a pas de trafic de contrôle continu pour les routes non utilisées mais par contre cette technique impose un grand délai avant l'ouverture de chaque route. Au niveau des réseaux ad-hoc, puisqu'il y a souvent et fréquemment des coupures de liens à cause de la mobilité des nœuds, cette méthode s'avère fournir une grande connectivité de réseau et une charge non utile moins importante que dans les méthodes proactives. Mais dans les réseaux maillés, les nœuds sont fixes et les liens demeurent beaucoup plus en vie puisque la fréquence des arrivées des flux est plus supérieure que celle de coupure des liens. Donc cette découverte de route basée sur l'inondation est redondante et très couteuse en termes de volume des messages de contrôle dans les réseaux, maillés. D'une manière générale, le principal avantage des protocoles réactifs est la minimisation du contrôle de trafic dans le réseau mais le coût de cet avantage est le long délai de découverte de route ce qui rend ce type de protocole non convenable pour les applications temps réel, d'autre part, l'inconvénient majeur de ce type de protocole est la taille des paquets puisque tout le chemin est inséré dans le paquet.

- Protocoles de routage hybrides : alors que les protocoles proactifs sont efficaces dans les réseaux où la mobilité est réduite, les réactifs conviennent quand la mobilité est élevée. De nouveaux protocoles ont été conçus en combinant les forces des deux types de protocoles : ce sont les protocoles hybrides comme ZRP (Zone Routing Protocol). La structure commune de ces protocoles est la suivante :

  Ils contiennent deux sous-protocoles A et B. A couvre une aire locale et B couvre une zone globale. L'idée de base de ces protocoles est d'organiser des groupes de nœuds pour maintenir un minimum d'informations sur la topologie de leurs voisins, un protocole « A » va gérer les paquets échangés au sein de son petit groupe (A est proactif) alors que « B » (B est réactif) va assurer le passage des paquets entre ces groupes.

Des travaux de comparaison entre ces différents types de routage ont été réalisés, p.ex. dans (Jiwei Cen, 2006) qui présente une comparaison de performance dans les WMN entre les protocoles AODV et OFLSR, et dans (Anna Zakrzewska, 2008) les auteurs effectuent une comparaison entre les performances des protocoles de routage DSDV (*Dynamic Destination Sequenced Distance Vector*), OLSR (*Optimized Link State Routing*), AODV(Ad-hoc On-Demand Distance Vector) dans le cadre des réseaux maillés sans fil. D'autre part, des métriques de routage ont été développées pour prendre en compte toutes les caractéristiques des WMN comme ETX (*Expected Transmission Count*), ETT (*Expected Transmission Time*) et le WCETT (*Weighted Culmulative ETT*) qui prend en compte l'interférence entre les liens qui utilisent le même canal (Draves, Padhye et Zill, 2004). L'équilibrage de charge, le support des radios multiples sont parmi les critères que doivent prendre en compte les algorithmes de routage.

CHAPITRE 2

TCP DANS LES RÉSEAUX SANS FIL MAILLÉS

2.1 Introduction

TCP est la technologie prédominante utilisée sur l'Internet. Il s'agit d'un protocole de transport orienté connexion, fiable et qui permet le contrôle de congestion. Il est utilisé pour transporter différents types de trafic Internet : HTTP, FTP, SMTP et TELNET. Comme déjà mentionné dans le chapitre précédent, la couche transport, comme toutes les autres couches, constitue un défi à résoudre pour les WMNs. TCP, qui a été conçu pour les réseaux filaires, présente des limites quand il est appliqué aux réseaux sans fils multi-sauts et particulièrement aux WMNs. Dans ce chapitre, on va présenter brièvement le protocole TCP et son mécanisme de contrôle de congestion, expliquer pourquoi le TCP standard n'est pas efficace quand il est appliqué dans les WMNs en faisant le lien avec les propriétés et les caractéristiques des réseaux sans fils maillés.

2.2 Le protocole TCP

Le protocole TCP est un protocole de la couche transport. TCP est l'acronyme de « Transmission Control Protocol ». Il est décrit dans la RFC 793 (Postel, 1981). Le protocole TCP permet un transport fiable et en ordre des données de bout en bout grâce à son mécanisme d'acquittement et de l'utilisation de numéros de séquence. TCP est un protocole orienté connexion, les applications communiquent alors entre elles comme si elles étaient physiquement connectées. Avant que les deux applications puissent s'échanger des informations, elles doivent établir une connexion entre elles ; cet établissement est connu sous le nom de « *Three-way handshake* ». Cette phase d'établissement permet aux deux

applications d'échanger et d'initialiser plusieurs paramètres du protocole. Une fois la connexion établie, les deux applications peuvent commencer à envoyer leurs données.

TCP fournit un service de contrôle de flux afin de s'assurer que le transmetteur n'envoie pas à un taux que le récepteur ne peut pas supporter et cause ainsi un dépassement de mémoire au niveau du tampon. Pour assurer ce contrôle de flux, TCP utilise une variable nommée « *receive window* » qui indique au transmetteur l'espace disponible dans le tampon du récepteur.

2.2.1 Le transfert fiable des données

Comme on l'a déjà mentionné, TCP fournit un transfert de données fiable. Un transfert fiable assure que les données envoyées arrivent au destinataire sans erreurs, sans lacunes, non dupliqués et en séquence. Pour assurer ce transfert fiable, TCP utilise essentiellement les deux notions suivantes : les temporisateurs et les acquittements (ACK). Au moment où le segment est émis, TCP déclenche un temporisateur. Si le transmetteur ne reçoit pas un acquittement relatif au segment émis avant l'expiration du temporisateur, le segment sera considéré comme perdu et le segment devra être retransmis. La question qui se pose à ce niveau est la valeur à assigner au temporisateur. Normalement, la valeur du temporisateur doit être supérieure au temps nécessaire à la transmission du segment jusqu'à la réception de son acquittement, soit le délai aller-retour, ou RTT (*Round Trip Time*). En effet la valeur du RTT est variable, et par suite si la valeur du temporisateur est inférieure au RTT alors on aura un *timeout* prématuré et une retransmission non nécessaire. Dans le cas où la valeur du temporisateur est beaucoup plus grande que le RTT, il y aura une lente réaction à la perte du segment. TCP fixe la valeur du timeout de la manière suivante : *SampleRTT* correspond à la mesure de l'intervalle de temps séparant l'envoi d'un segment et la réception de son acquittement. Vu que la valeur de *SampleRTT* va éventuellement connaitre des fluctuations, la valeur de *SampleRTT* ne va pas être typique. Afin de trouver une valeur typique, TCP va calculer une moyenne pour les valeurs de *SampleRTT* obtenues qui est nommée

EstimatedRTT et qui va permettre de mettre à jour la valeur de *SampleRTT* à chaque réception comme suit :

$$EstimatedRTT = (1- \alpha)*EstimatedRTT + \alpha*SampleRTT$$ (James F. Kurose, 2003).

α=0.125 est la valeur recommandée dans la RFC 2988. Outre la mesure d'*EstimatedRTT*, la norme RFC 2988 définit la variation du RTT, *DevRTT* comme suit :

$$DevRTT = (1-\beta)*DevRTT + \beta*|SampleRTT-EstimatedRTT|$$ (James F. Kurose, 2003).

β=0.25 est la valeur typiquement recommandée.

Étant donné toutes ces valeurs, la valeur du timeout sera calculée comme suit :

$$TimeoutInterval = EstimatedRTT + 4*DevRTT$$ (James F. Kurose, 2003).

2.2.2 Le contrôle de congestion TCP :

Une autre composante clé du protocole TCP, et qui peut être classifiée comme le plus grand avantage que TCP a apporté à Internet est son mécanisme de contrôle de congestion. TCP utilise un contrôle de congestion de bout en bout. Le principe de base est que chaque source TCP limite son débit de transmission en fonction de l'état de congestion tout au long du chemin vers la destination. Si aucune congestion n'a été détectée alors la source TCP peut augmenter son débit de transmission. Dans le cas de perception de congestion en détectant une perte de paquets, la source TCP doit réduire son débit de transmission pour ne pas aggraver encore la situation. Pour le contrôle de congestion, TCP utilise une variable appelée fenêtre de congestion et notée *Cwnd*. Cette fenêtre constitue une limite pour le taux d'envoi pour la source TCP. Elle constitue le nombre maximal de paquets que l'émetteur peut envoyer sans recevoir d'accusé de réception. Comment la source TCP règle-t-elle son débit de transmission ? Comment TCP détecte-t-il ou s'aperçoit-il de la congestion? C'est le rôle de l'algorithme de congestion de TCP.

La RFC 2581 (Allman, Paxson et Stevens, 1999) a défini quatre algorithmes qui globalement réalisent le contrôle de congestion :

- *Slow Start* : quand une connexion TCP commence, TCP ne sait rien de l'état et de la charge du réseau. La phase de démarrage consiste alors à tester la bande passante disponible. Au démarrage d'une connexion, cet algorithme initialise la valeur de la fenêtre de congestion *cwnd* à 1 MSS. Après la réception de l'acquittement, TCP transmet deux segments. Ainsi le transmetteur TCP augmente le taux d'envoi exponentiellement en doublant la valeur de la fenêtre de congestion *cwnd* à chaque RTT. Ce processus continue jusqu'à la détection d'une première perte soit à travers le déclenchement d'un timeout, soit suite à la réception de plus de 3 acquittements dupliqués. Dans ce cas l'algorithme *congestion avoidance* prend la relève. Le seuil de détection de congestion *ssthresh* est déterminé à la détection d'une perte d'un segment et correspond à la moitié de la taille de la fenêtre *cwnd*.
- *Congestion avoidance* : pendant la phase *Slow Start*, la fenêtre de congestion augmente exponentiellement d'où le risque d'une saturation rapide du réseau. Le but de cet algorithme est alors de réduire la cadence d'envoi après avoir atteint un certain seuil. Cette phase commence au moment où *cwnd* > *ssthresh*. Pendant cette phase la fenêtre de congestion *cwnd* augmente linéairement. La valeur de *cwnd* est incrémentée de 1 MSS à chaque RTT indépendamment du nombre d'acquittements reçus durant ce RTT. S'il y a eu une détection de perte, suite à un timeout ou bien suite à la réception de plus de deux acquittements dupliqués, le *ssthtreh* est réduit à sa moitié. Si la perte à été détectée suite à un timeout, *cwnd* est mis à 1 MSS et on reprend avec la phase *Slow Start*, mais si elle a été détectée suite à la réception de trois acquittements dupliqués ou plus TCP exécute les algorithmes *Fast Retransmit* et *Fast Recovery* qu'on va présenter ci-dessous.
- *Fast Retransmit* : lorsqu'un accusé de réception porte le même numéro qu'un accusé de réception précédemment envoyé pour le dernier segment ordonné reçu, on dit que ces accusés de réception sont dupliqués. Cet algorithme considère que la réception de plus de deux accusés de réception dupliqués est une indication qu'au moins un segment a été perdu. La source TCP ne va pas attendre le déclenchement du timeout pour retransmettre le segment perdu : il va le retransmettre immédiatement, réduire sa fenêtre de congestion de moitié et passer à l'algorithme *Fast Recovery*.

- *Fast Recovery* : cet algorithme part du principe que la réception de trois accusés de réception ou plus n'est pas un signe d'une congestion sévère. En conséquence, au lieu de passer à la phase de *slow start*, cet algorithme reprend la transmission avec une large fenêtre et l'incrémente comme dans la phase d'évitement de congestion. La nouvelle valeur de la fenêtre de congestion est $cwnd = ssthreh + 3\ MSS$; ceci est appelé gonflement artificiel du cwnd (Moraru *et al.*, 2003). Et puis à chaque acquittement dupliqué, on incrémente *cwnd* de 1 MSS.

De nos jours, plusieurs versions de TCP ont été développées. Parmi ces versions on cite TCP Tahoe (Postel, 1981) qui utilise les algorithmes *Slow Start*, *Congestion Avoidance* et *Fast Retransmit*, TCP Reno (Allman, Paxson et Stevens, 1999) qui est le même que Tahoe mais utilise en plus le *Fast Recovery*, TCP NewReno (Floyd et Henderson, 1999) qui est une version améliorée de Reno avec une légère modification de l'algorithme *Fast Recovery*, qui est capable de détecter de multiples pertes de segments. Au contraire de Reno, New Reno ne quitte la phase *Fast Recovery* qu'après la réception des accusés de réception de tous les paquets perdus au moment où il est entré en cette phase. Ainsi il résout l'inconvénient de Reno où la *cwnd* peut décroitre plusieurs fois durant un RTT.

2.3 Application du protocole TCP dans les réseaux sans fils maillés

TCP est connu par ses performances limitées dans les réseaux sans fils en général et dans les réseaux sans fils multi-sauts en particulier (S.G, T et G, 2008). Cette dégradation de performance est due essentiellement aux propriétés des canaux sans fils. Nous présentons brièvement comment les propriétés sans fils influent sur la performance de TCP, en nous concentrant sur les réseaux – ou mécanismes d'accès – IEEE 802.11.

2.3.1 Un bref rappel sur quelques propriétés sans-fil

Un canal sans fil est un canal partagé. L'accès au canal se fait à travers deux méthodes d'accès, DCF et PCF. On s'intéresse à la méthode DCF puisque PCF est utilisé uniquement pour les réseaux sans fils avec infrastructure. DCF est une méthode décentralisée qui est basée sur le mécanisme CSMA/CA. CSMA/CA est un algorithme qui permet l'écoute du médium avant de transmettre pour éviter les collisions. Cet algorithme est exécuté localement sur chaque station afin de déterminer les périodes d'accès au médium : une machine voulant émettre écoute le réseau et si le support est libre durant une période DIFS alors la station peut émettre, sinon l'accès est différé et la machine doit exécuter l'algorithme de retrait (*backoff*). Toutefois, il y a deux problèmes qui peuvent intervenir : le problème du nœud caché et celui du nœud exposé.

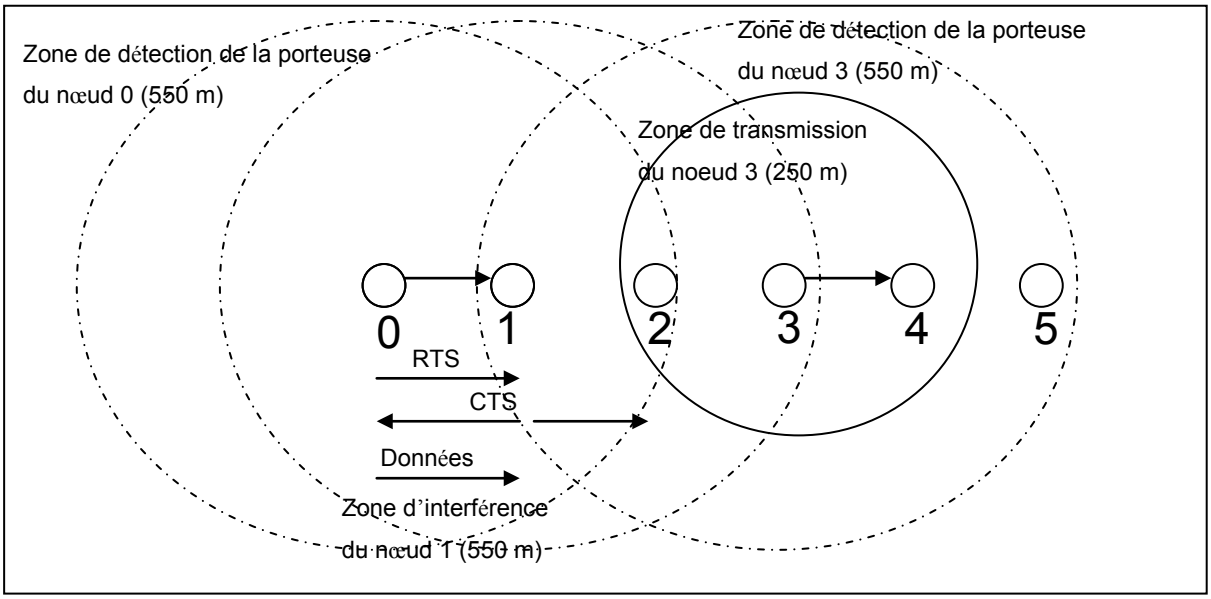

Figure 2.1 Les problèmes du nœud caché et du nœud exposé.
Adaptée de Tung *et al.* (2007, p. 65)

A partir de la Figure 2.1 ci-dessus on observe que le problème du nœud caché se produit dans le scénario suivant : si le nœud 4 est en train de transmettre au nœud 5, puisque le nœud 4 est en dehors de la zone de détection de porteuse du nœud 1 ce dernier peut initier une

transmission vers le nœud 2. Une collision se produira alors au niveau du nœud 2 puisque le nœud 4 est situé dans sa zone d'interférence. Le mécanisme RTS/CTS qui permet de résoudre le problème du nœud caché dans un réseau local sans fil à un saut est inefficace dans le cas d'un réseau sans fil multi-sauts.

Toujours à partir de la Figure 2.1, le problème du nœud exposé se manifeste dans le scénario suivant : admettons que le nœud 4 est en train de transmettre au nœud 5. Le nœud 2 ne peut pas initier une transmission car le nœud 4 est dans sa zone de détection de porteuse bien que, si le nœud 2 transmettait au nœud 1, il n'y aurait pas d'interférence entre les deux transmissions. Le nœud 2 sera alors forcé de différer inutilement sa transmission.
Contrairement aux réseaux filaires, les réseaux sans fils multi-sauts sont alors sujets à plusieurs sources d'erreurs de transmission causées non seulement par les problèmes du nœud exposé et du nœud caché que nous venons de présenter, mais également par la perte de communication due à la perte des paquets qui arrivent à la destination avec une puissance de signal insuffisante (Tung *et al.*, 2007).

Dans ce qui suit, on va présenter comment ces problèmes ainsi que le caractère multi-saut des WMNs influent sur la performance de TCP et par suite les défis rencontrés quand il est exploité dans ce genre de réseaux.

2.3.2 Les défis de TCP dans les réseaux maillés sans fils

Initialement conçu pour les réseaux filaires où les erreurs de transmissions et donc les pertes de paquets sont rares, le protocole TCP fait face à plusieurs défis quand il est déployé dans les WMNs.

Dans (Tung *et al.*, 2007), les auteurs observent à partir de leurs simulations que le débit de TCP est, approximativement, inversement proportionnel au nombre de sauts du chemin de transmission sans fil. Les problèmes des nœuds cachés et exposés peuvent être à l'origine de

la dégradation de performance de TCP. Il est à noter que dans cette étude on utilise une seule radio. Le mécanisme RTS/CTS ait été proposé pour résoudre le problème des nœuds cachés dans les WLANs. Dans les réseaux sans fil maillés, il a été montré qu'il est capable de résoudre ce problème pour la transmission des paquets ACK. Toutefois, il existe encore une grande probabilité que le problème se produise lors de la transmission des paquets de données. La perte des messages RTS constitue la majorité des pertes dues aux problèmes des nœuds cachés (73% des pertes). Puisque la transmission avec succès d'un paquet TCP nécessite la transmission avec succès d'un RTS, la performance de TCP se voit alors dégradée suite à ces pertes (Tung *et al.*, 2007).

Dans les réseaux sans fils multi-sauts, l'accès au canal se produit à chaque nœud dans le chemin. Ces erreurs de transmission dues aux nœuds cachés vont augmenter le délai d'accès au canal pour les paquets TCP ce qui conduit à une ouverture plus lente de la fenêtre de congestion et par conséquence une diminution du débit TCP.

D'autre part, le problème du nœud exposé qu'on a déjà expliqué conduit à une utilisation inefficace du canal entre les nœuds adjacents puisqu'il arrive qu'un nœud diffère inutilement ses transmissions.

Autres que les problèmes qu'on vient de citer et d'une manière générale, TCP connait les défis suivants quand il est déployé dans les WMNs :

- Les erreurs de transmission, la détection et le contrôle imprécis de congestion : la perte de paquets due aux erreurs de transmission est très fréquente dans le contexte des WMNs. Ces pertes de paquets sont considérées par TCP comme des signes de congestion. TCP va alors diminuer la taille de sa fenêtre de congestion et par suite le débit sera atténué (Liu, Shen et Sun, 2007). Donc le TCP classique ne différencie pas les pertes liées aux congestions et celles liées aux erreurs de transmission [20]. D'autre part, dans les WMNs, les coupures de liens sont fréquentes (Ekram Hossain, 2008). Ces coupures de liens engendrent un calcul de nouveaux chemins. Puisque tous les chemins ont des RTTs différents, la mesure du RTT sur les différentes routes résulte en une large variance de

son estimation ce qui donne une grande valeur pour le temporisateur de retransmission (Ekram Hossain, 2008).

- Le rétablissement des routes : suite aux coupures de liens, les nœuds procèdent à un nouveau calcul de route. Une fois la route recalculée, TCP entre dans la phase du départ lent pour chercher la bande passante disponible. Cette entrée dans cette phase engendre une dégradation du débit (Ekram Hossain, 2008).
- L'ordonnancement des paquets TCP à la réception : dans les WMNs, les paquets peuvent suivre des chemins différents et arriver à la destination d'une manière désordonnée. S'ils ne sont pas traités parfaitement, des retransmissions de paquet sont nécessaires (Liu, Shen et Sun, 2007).
- Asymétrie des réseaux : on parle de réseaux asymétriques quand le chemin de l'aller ou de transfert est différent du chemin du retour en termes de débit, de latence et de taux de perte (Hari Balakrishnan, 1997). Ce phénomène est plus important dans les réseaux multi-sauts pour différentes raisons. Les paquets et les ACKs peuvent prendre des chemins différents et ainsi ils auront une bande passante, un débit, et un taux de perte différents. Même dans le cas où ils prennent le même chemin, le problème peut toujours exister puisque la condition de canal et la bande passante changent au cours du temps (Bing, 2008). Puisque le contrôle d'erreur de TCP se base sur la temporisation des ACKs, une perturbation au niveau du feedback peut diminuer la performance. La bande passante des liens de retour peut changer et augmenter l'inter-espacement original entre les ACKs arrivant à l'émetteur, ce qui va causer un ralentissement de l'évolution de la fenêtre de congestion. D'autre part, si des données sont envoyées dans le chemin de retour, les ACKs vont trouver le chemin chargé et vont être enfilés derrière les paquets ce qui augmente le délai et la probabilité de ces ACKs et par suite il y aura une dégradation de la performance de TCP (Hari Balakrishnan, 1997).
- Un grand produit délai-bande passante : récemment, il y a eu une augmentation dans la capacité des liens sans fil surtout avec la technologie IEEE 802.11n qui permet d'atteindre des débits théoriques de 300 Mbps dans la bande de fréquences des 2.4 GHz et de 270 Mbps dans la bande de fréquences des 5GHz. Cette technologie se base

essentiellement sur la technologie MIMO. L'utilisation de ces technologies dans les WMNs qui ont un grand délai de bout en bout, conduit à un produit délai-bande passante très grand. Ce grand produit nécessite une très grande taille de tampon (Bing, 2008) pour que TCP puisse exploiter la bande passante disponible.

- Fluctuation du RTT : dans les WMNs, les délais de bout en bout sont très variables ce qui influe sur la mesure des RTTs. Le RTT est une variable très critique à TCP et cette variation peut affecter sa performance (Ekram Hossain, 2008).

2.4 Equité TCP :

Si on a k connexions TCP qui vont partager le même lien-goulot de capacité R, on a une équité absolue si et seulement si chaque connexion a une bande passante de R/k. Considérons le cas où on a deux connexions TCP qui vont partager un lien de capacité R comme le montre la Figure 2.2 ci-dessous. En supposant que ces deux connexions ont le même RTT et le même MSS et en ignorant la phase du départ lent, c'est-à-dire en supposant que les deux connexions opèrent dans la phase d'évitement de congestion, le débit réalisé par les deux connexions est illustré par la Figure 2.3. Le but étant de partager équitablement la bande passant et de maximiser l'utilisation de lien, idéalement les débits doivent être près de l'intersection de la ligne du partage équitable et celle de l'utilisation maximale du lien où la somme des débits des deux connexions soit égale à R. À partir de la Figure 2.3 nous constatons que l'algorithme AIMD permet de converger vers un partage équitable de la bande passante (James F. Kurose, 2003).

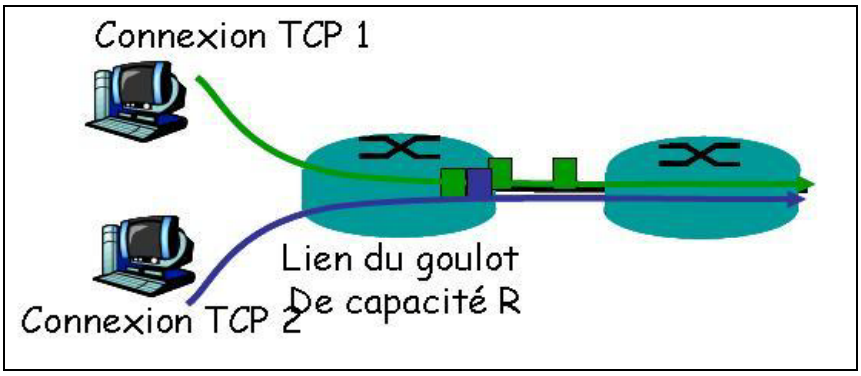

Figure 2.2 Deux connexions TCP qui partagent un lien goulot.

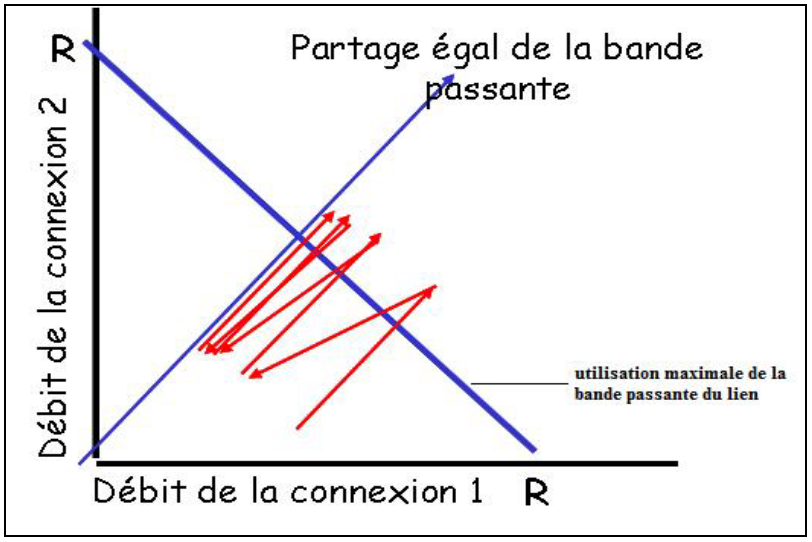

Figure 2.3 Débit réalisé par la connexion 1 et la connexion 2.

Tirée de James F. Kurose et Keith W. Ross (2005)

En se basant sur la formule *square root* (Mathis *et al.*, 1997), le débit de chaque connexion TCP peut être représenté par la formule suivante :

$$débit \leq \frac{8 \times v \times MSS}{RTT \times \sqrt{p}},$$

où ν représente une constante qui est approximativement égale à $\sqrt{2}$ quand TCP n'utilise pas le retardement des ACKs (quand TCP utilise cette option il n'accuse pas chaque segment reçu, mais il combine les accusés avec les segments de données) et elle est approximativement égale à 1 quand TCP utilise le retardement des ACKs ; p est le taux de perte et MSS est la taille du segment TCP.

En présence de plusieurs flux et de sources de trafic, c'est souhaitable d'avoir un partage équitable de la bande passante. La caractéristique multi-saut des réseaux WMNs fait que les nœuds sont éloignés d'une manière inégale de la passerelle ou de la destination, ce qui fait que les délais et par suite les RTTs seront inégaux. D'autre part, tous les problèmes déjà cités tels que les problèmes des nœuds cachés, les conditions des canaux qui changent, la diversité de chemin pour arriver à la destination, la contention pour l'accès au canal, etc., constituent des raisons pour lesquelles les RTTs et le taux de perte vont varier d'un flux à un autre. En partant de l'équation ci-dessus on peut prédire qu'on peut avoir un problème d'équité entre les différents flux. Assurer l'équité est primordial pour les WMNs pour permettre un partage équitable entre les différentes sources indépendamment de leur distance de la passerelle.

Conclusion

Dans ce chapitre, on a présenté les défis que TCP rencontre dans les réseaux sans-fil maillés. Les propriétés sans-fil et le caractère multi-saut des WMNs influent sur la performance de TCP tel qu'il a été conçu pour les réseaux filaires, où il n'y avait pas les erreurs de transmission des réseaux sans fil. D'autre part, l'algorithme AIMD et le contrôle de congestion se basent sur la perte de paquets pour la détection de la congestion, mais dans les réseaux sans-fil, la perte de paquets n'est pas forcément un signe de congestion ce qui fait qu'il y aura une détection et un contrôle imprécis de congestion ce qui dégrade la performance de TCP.

Le partage équitable de la bande passante entre les différents flux est une priorité. Dans les WMNs, les flux vont avoir des RTTs et des taux de pertes différents et par suite on pourra avoir un problème d'équité. Comment se manifeste cette équité ? Comment l'améliorer ? Ce sera l'objet de la suite de notre étude.

CHAPITRE 3

SIMULATIONS

3.1 Méthodologie

Dans notre travail on vise à étudier et analyser l'équité TCP dans les réseaux multi-sauts en mettant l'accent sur les réseaux sans-fil maillés. Cette étude sera effectuée à travers un ensemble de simulations. Nous partons avec l'hypothèse de ne rien changer dans le protocole TCP, mais plutôt proposer des mécanismes simples et réalistes pour améliorer l'équité. Pour ce faire on va utiliser le simulateur NS-2 (*Network Simulator*), quoique NS-2 ne nous permette pas de créer et de simuler le modèle exact des réseaux sans-fil maillés avec tous leurs acteurs. La Figure 3.1 illustre une topologie autorisée par NS-2 où il est possible d'intégrer des nœuds filaires et des nœuds sans-fil ; toutefois, on ne peut pas créer avec ce genre de topologie un réseau sans fil maillé. La deuxième possibilité avec NS-2 est de simuler seulement la dorsale du réseau sans-fil maillé qui sera une topologie similaire à un réseau ad-hoc. Cette option ne nous permet pas d'intégrer les clients et la passerelle et par suite nous ne pourrons pas voir l'effet du réseau externe, Internet, sur la performance TCP.

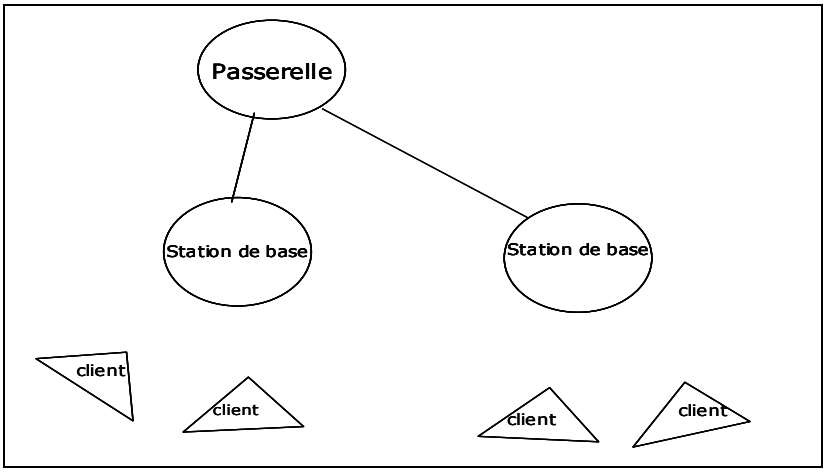

Figure 3.1 Topologie wired-cum wireless (NS-2).

Nous avons alors choisi de simuler une topologie d'un réseau multi-sauts filaire qui fait abstraction au comportement des réseaux sans-fil maillés. La topologie qu'on a choisie sera présentée dans le paragraphe suivant. Notre objectif est de commencer par un environnement filaire qui est plus propice au protocole TCP, essayer d'identifier les paramètres qui influent sur l'équité TCP et de trouver des solutions réalistes et simples pour l'améliorer. Les resultats des simulations seront ensuite validés à travers le banc de test.

A partir de ce qu'on a déjà vu dans le chapitre précédent sur les propriétés des réseaux maillés sans fil et les propriétés TCP, les paramètres à étudier et qui auront un impact sur les performances TCP sont essentiellement la taille des tampons et le RTT.

3.2 La topologie simulée

La topologie sur laquelle on va réaliser nos simulations est illustrée par la Figure 3.1. Elle est composée d'un ensemble de nœuds à différents sauts de la destination pour constituer le caractère multi-sauts. Un ensemble de nœuds va représenter l'accès au WMN et seront les sources du trafic ; ces nœuds sont source 1, source 2, source 3 et source 4. Un autre ensemble de nœuds représente les routeurs de relais (MP), soit MP_1, MP_2. Le nœud MP_3 va représenter la passerelle vers les réseaux externes et éventuellement vers Internet. Les liens entre ces nœuds, qu'on va appeler les liens internes et ayant des délais D_i vont constituer le réseau maillé. Le lien entre le nœud MP_3 et le nœud Dest sera appelé le lien externe et a pour délai D_e. Tous les liens auront un débit de 10 Mbps. Comme on a déjà mentionné, dans les réseaux tous les flux sont vers ou bien à partir de la destination. Dans notre cas on va considérer les flux ascendants vers la destination où il y aura une compétition pour le partage de la bande passante. Les sources ont toujours des données à transmettre (*Greedy sources*) et la taille des paquets est de 1000 octets. Chaque source émet 5 flux TCP pour s'assurer que les flux de la meme source ont le même comportement. Il est à noter aussi que les graphiques correspondants aux simulations sont relatifs a une seule réalisation et ne représentent pas une

moyenne. On n'a pas besoin d'effectuer plusieurs réalisations puisqu'avec NS-2, en répétant la simulation avec les mêmes paramètres on aura les mêmes résultats.

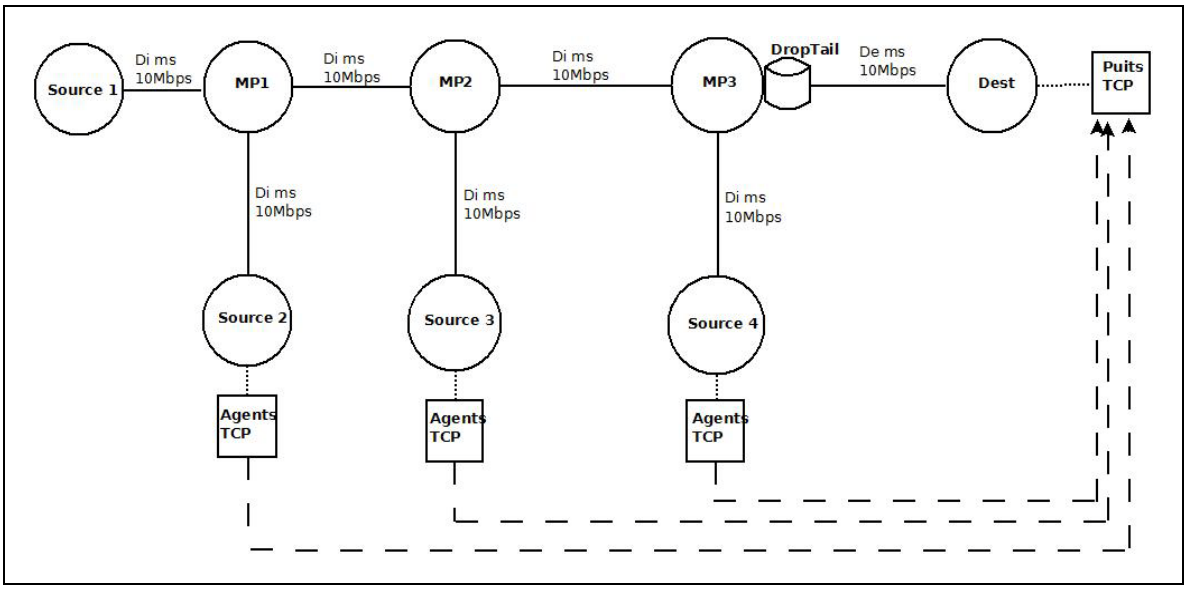

Figure 3.2 Topologie simulée.

3.3 Étude avec la gestion de file d'attente *DropTail*

Dans ce qui suit on va étudier l'équité de TCP entre les flux des différentes sources. Les sources sont à différents nombres de sauts de la passerelle MP_3 et ainsi elles auront de différentes valeurs de RTT. Notamment les sources les plus en aval auront un RTT plus important. À l'issue de quelques simulations on a constaté que la taille des tampons des nœuds MP_1 et MP_2 n'a pas d'influence sur la performance de TCP. Par contre le lien qui constitue le goulot d'étranglement est celui entre le nœud MP_3 et le nœud Dest. Comme c'est montré dans la figure, on va utiliser une file d'attente de type *DropTail* pour la passerelle MP_3. *DropTail* est un algorithme de gestion de file d'attente très simple. Le principe est d'accepter tout paquet qui arrive tant que le tampon n'est pas plein. Si le tampon est plein, chaque paquet qui arrive sera rejeté. Ci-dessous on va présenter et discuter les résultats des différentes simulations.

On a, dans l'ensemble, quatre scénarios. D'un scénario à l'autre on va modifier les délais des liens : 5 ms, 10 ms, 15 ms et 20 ms. Pour chaque cas de délai on va modifier la taille du tampon et on va suivre l'évolution des débits des différentes sources. Dans la suite, on sous-entend par débit le débit utile.

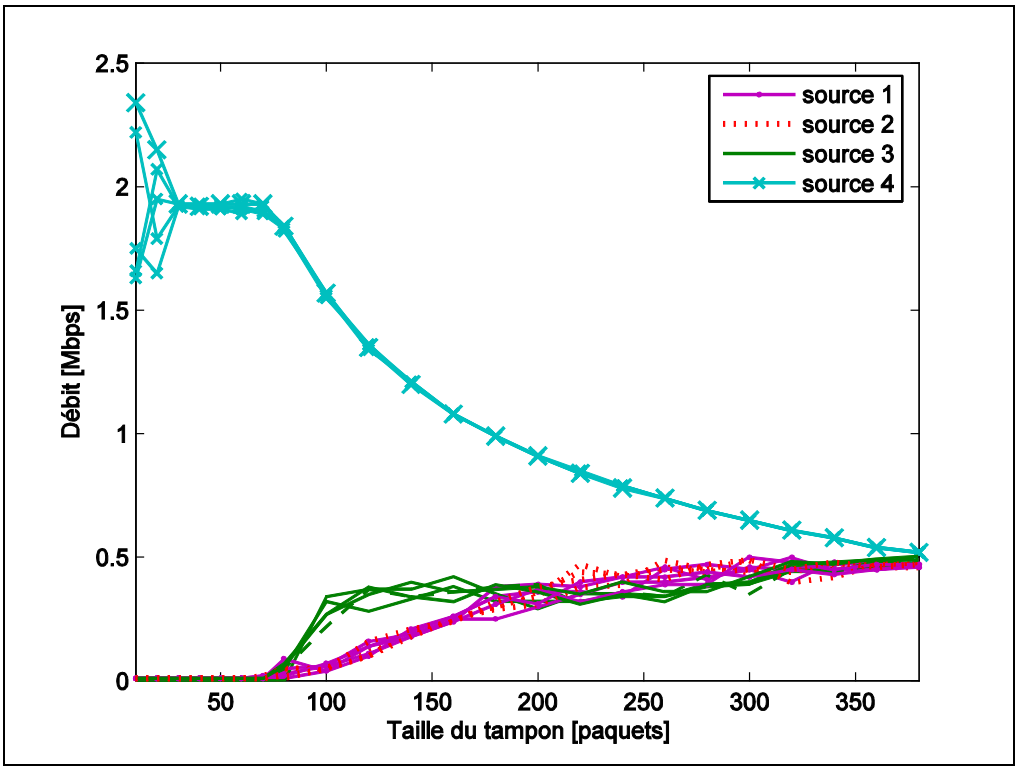

Figure 3.3 Débit en fonction de la taille du tampon avec un délai des liens de 5 ms.

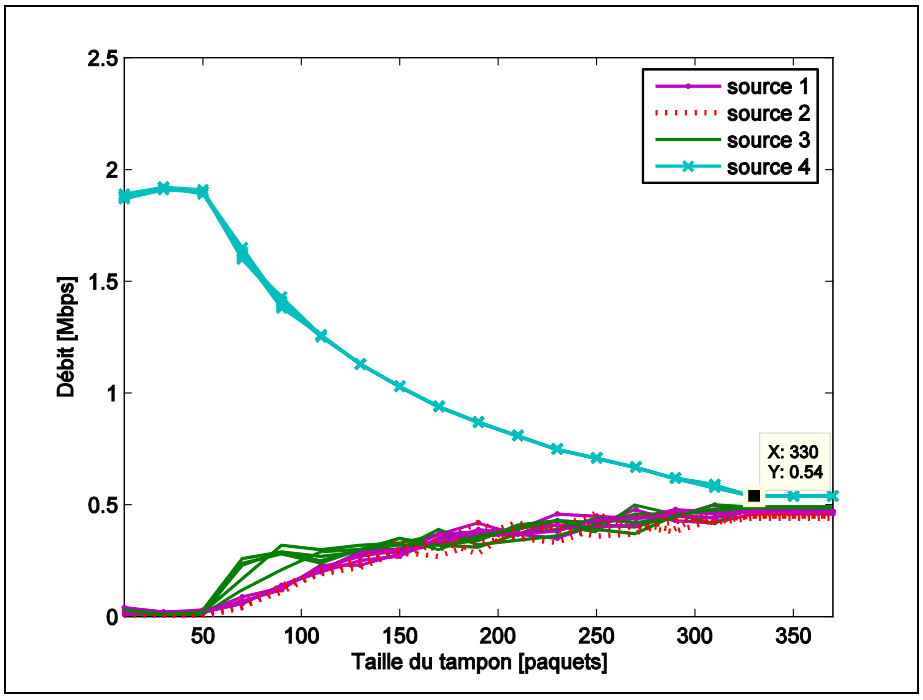

Figure 3.4 Débit en fonction de la taille du tampon avec un délai des liens de 10 ms.

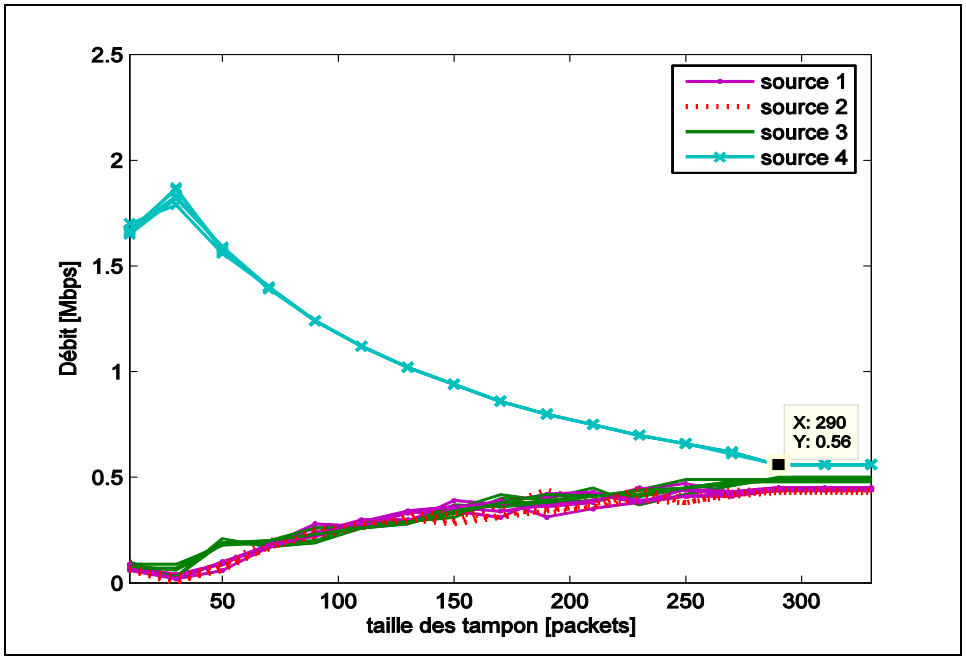

Figure 3.5 Débit en fonction de la taille du tampon avec un délai des liens de 15 ms.

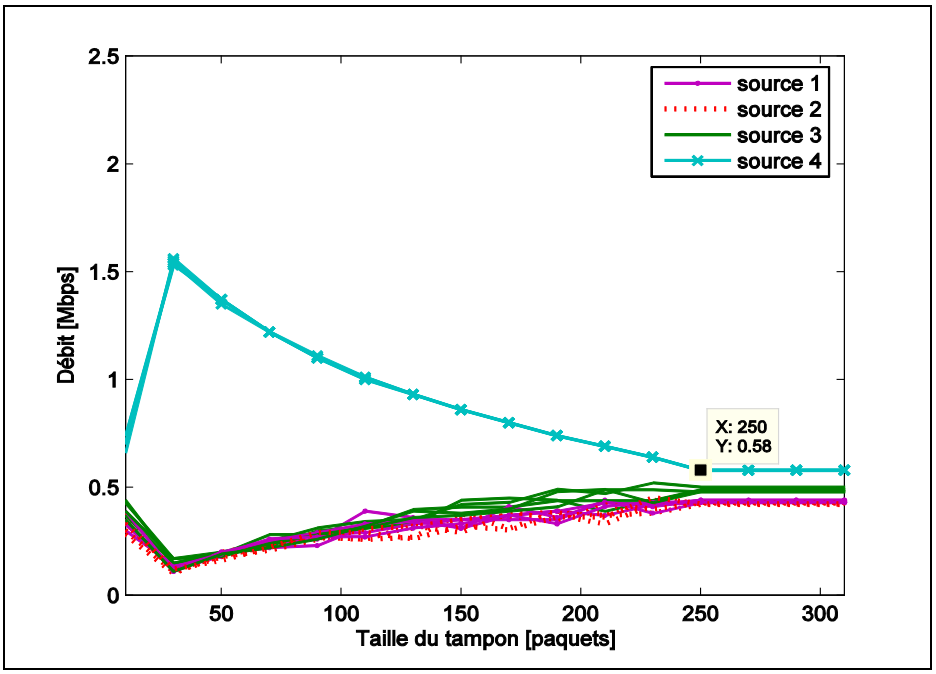

Figure 3.6 Débit en fonction de la taille du tampon avec un délai des liens de 20 ms.

D'après les quatre figures ci-dessus, la première chose à remarquer est le fait que la source 4 a toujours un meilleur débit que les autres. La source 4 est la source qui est la plus proche de la destination, et elle a le plus faible RTT. D'autre part on remarque que les différentes sources TCP ne partagent pas équitablement la bande passante. Sur la Figure 3.3 et la Figure 3.4 on constate que l'équité s'améliore en augmentant la taille du tampon. On atteint une situation stable et une meilleure équité pour une taille de tampon plus petite dans le cas où le délai est de 10 ms, où la taille est de 380 paquets, que dans le cas où le délai est de 5 ms où le meilleur point correspond à une taille du tampon de 330 paquets. Toujours concernant ces deux figures, avec une petite taille de tampon, la source 4 s'empare de la bande passante et son agressivité cause presque un débit de famine aux autres sources qui sont plus en aval. On observe que, jusqu'à une taille de tampon de 90 paquets dans le cas où le délai est de 5 ms et une taille de 50 dans le cas où le délai est de 10 ms le niveau d'équité est le pire, et les sources les plus en aval ont un débit quasiment nul. Ayant un RTT le plus faible, la source 4 se voit très agressive. Pour une taille de tampon jusqu'à 90 paquets par exemple pour le cas

de la Figure 3.3, le tampon sera rempli par les paquets de la source 4, alors que les paquets des autres vont le trouver plein en arrivant et leurs paquets seront rejetés et perdus. Ces pertes obligent l'algorithme de congestion TCP à rester dans la phase *Slow Start* et leurs fenêtres de congestion auront du mal à s'ouvrir et évoluer ce qui va affecter leur débit. La Figure 3.7 et la Figure 3.8 illustrent le taux de perte des flux dans le cas où le délai est de 5 ms et 10 ms respectivement. À partir de ces figures on constate que le taux de perte relatif à la source 4 est quasiment nul quelle que soit la taille du tampon, ce qui est conforme à ce qu'on déduit des courbes de la Figure 3.3 et la Figure 3.4, c'est-à-dire que la source 4 s'empare du tampon et les paquets des autres sources vont subir plusieurs pertes consécutives. C'est à partir d'une taille donnée comme on l'a mentionné pour les courbes de débit que les paquets des sources victimes vont commencer à avoir des chances d'être acceptés dans la file d'attente, et à partir de cette taille-là on commence à observer la chute du taux de perte. Ce taux de perte continue à diminuer jusqu'à atteindre une valeur nulle La taille de tampon correspondant à ce taux de perte nulle donne une meilleure équité. Le RTT reste alors le seul paramètre qui entre en jeu et qui favorise une source par rapport aux autres. Ces courbes confirment aussi que l'augmentation du délai de 5 ms à 10 ms améliore le comportement et fait en sorte qu'on atteint ces limites pour des valeurs de taille de tampons plus faibles.

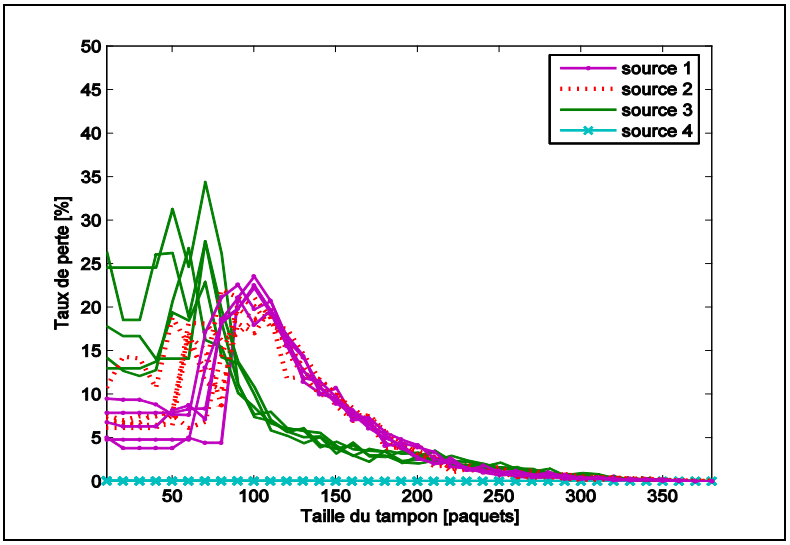

Figure 3.7 Taux de perte en fonction de la taille

du tampon.avec délai de liens de 5 ms.

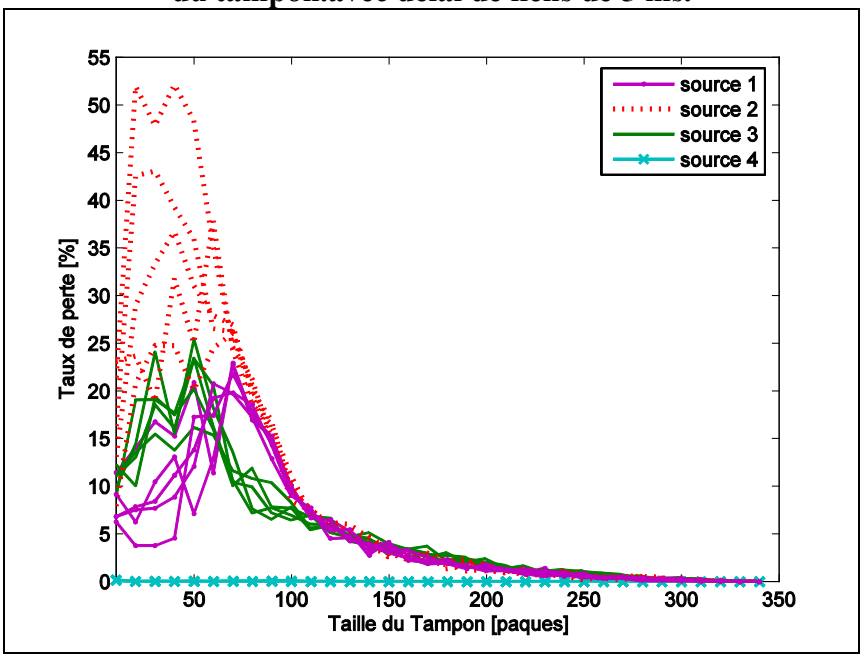

Figure 3.8 Taux de perte en fonction de la taille du tampon avec délai de liens de 10 ms.

Passons maintenant à la Figure 3.5 et la Figure 3.6. À partir de ces figures, on constate un nouveau comportement qui apparait par rapport aux courbes analysées précédemment. Pour une taille de tampon de 10 paquets, on remarque qu'on a une meilleure équité par rapport aux cas précédents. Les sources les plus en aval réussissent à avoir de la bande passante et leur débit n'est plus quasiment nul. L'augmentation du délai et par suite du RTT a provoqué quelques pertes pour les paquets de la source 4 tel qu'illustré dans la Figure 3.9 et la Figure 3.10. Ceci est expliqué par le fait que l'augmentation du RTT a fait en sorte que les paquets des sources les plus en aval ont plus de chance d'être acceptés dans le tampon, et cette taille de tampon de 10 paquets est assez limitée pour que tous les paquets de la source 4 soient acceptés dans la file. On remarque par exemple à partir de la Figure 3.10 que le taux de perte quand la taille du tampon est de 10 paquets et presque le même pour toutes les sources. Par la suite on remarque que le débit de la source 4, comme le montre la Figure 3.5 et la Figure 3.6, a atteint un sommet pour une taille de tampon donnée pour chaque valeur de délai et, parallèlement, on constate l'absence de pertes de paquets à partir de cette taille de tampon.

Cette taille de tampon et le RTT font en sorte que les paquets de la source 4 ne seront plus jetés et perdus. Maintenant, à partir de ce sommet, en augmentant la taille du tampon, on a le même comportement qu'on a vu précédemment dans la Figure 3.3 et la Figure 3.4. Les paquets des sources les plus en aval ont plus de chance â être acceptés dans la file, leur taux de perte commence à diminuer comme le montre la Figure 3.9 et la Figure 3.10 jusqu'à ce qu'ils s'annulent. En diminuant le taux de perte les fenêtres de congestion s'ouvrent mieux et le débit augmente en conséquence.

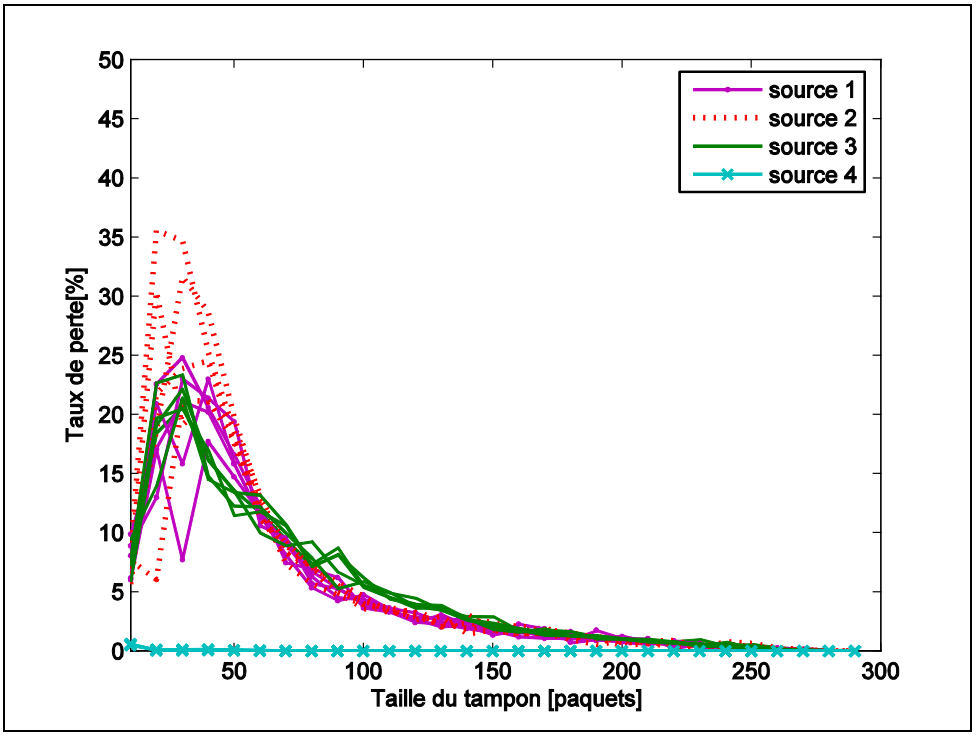

Figure 3.9 Taux de perte en fonction de la taille du tampon avec délai de liens de 15 ms.

Pour conclure, on constate que la source la plus proche de la destination et qui a le plus faible RTT est la source la plus favorisée en terme de bande passante. Son agressivité fait en sorte que les paquets des autres sources ont moins de chance d'être acceptés, et vu l'algorithme de contrôle de congestion de TCP et la dépendance du débit TCP au RTT et au taux de pertes, ces sources vont être défavorisées. La politique de gestion de file d'attente n'est pas

intéressante pour assurer l'équité : elle n'offre pas une sélection de rejet de paquets. Elle cause une perte sporadique de tous les paquets qui trouvent la file pleine. Augmenter le rapport entre les RTT de la source 4 et les autres sources en augmentant les délais apporte des améliorations de point de vue équité. On a observé que l'équité est meilleure pour de grandes tailles de tampon qui ne constitue pas une solution typique et intéressante puisqu'une grande taille de tampon a plus de coûts et cause des délais importants.

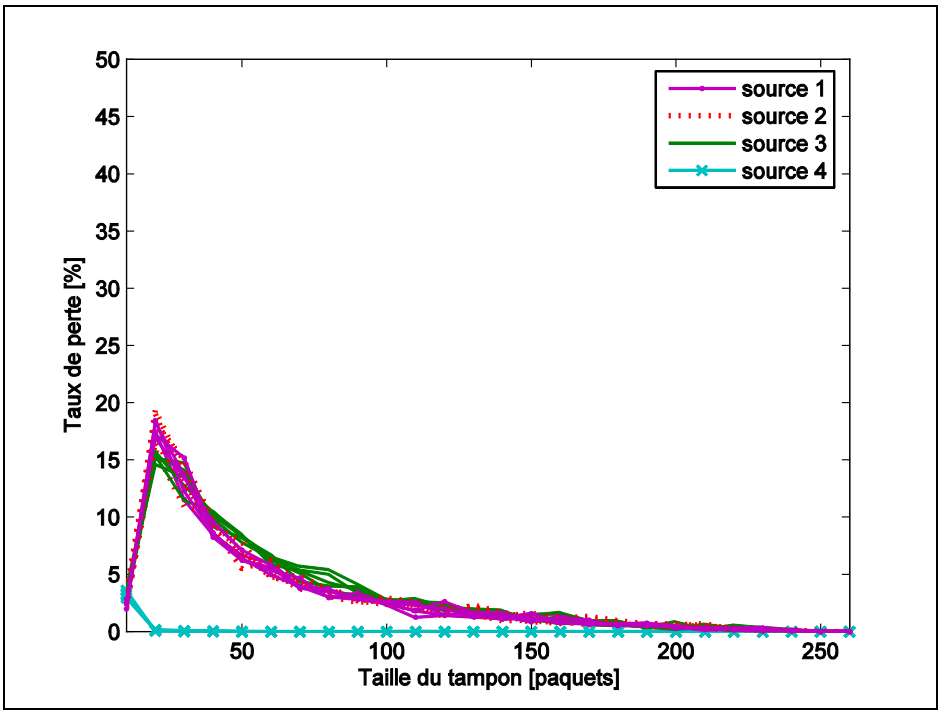

Figure 3.10 Taux de perte en fonction de la taille du tampon avec délai de liens de 20 ms.

3.4 Étude avec la gestion de file d'attente RED

3.4.1 RED : Random Early Discard

RED (Floyd et Jacobson, 1993) est un mécanisme de gestion active des files d'attente introduit par Floyd et Jacobson en 1993. L'apport de RED par rapport à *DropTail* est la

détection préemptive de la congestion. RED n'attend pas que le tampon soit plein pour détecter la congestion à travers la perte de paquets. Le principal objectif de RED est l'évitement de congestion en contrôlant la taille moyenne du tampon (Floyd et Jacobson, 1993). Parmi les autres objectifs derrière l'utilisation de RED que Floyd et Jacobson ont identifiés, on trouve :

- Éviter les pertes synchronisées : Avec *DropTail*, plusieurs connexions peuvent subir des pertes au même temps et par suite elles vont réduire simultanément leur fenêtre de congestion. Cette synchronisation de perte conduit à une diminution du débit total et une sous-utilisation de la capacité du lien. Au contraire, RED permet d'éviter ces synchronisations puisque l'effacement de paquets est aléatoire et il ne se produit pas en rafale.
- Éviter les biais contre le trafic en rafale : ce problème de biais contre le trafic en rafale existe avec l'utilisation de *DropTail*. Avec *DropTail* plus que le trafic d'une certaine connexion est en rafale, il est plus probable que la taille maximale du tampon sera dépassée au moment de l'arrivée des paquets de cette connexion (Floyd et Jacobson, 1993).
- Le contrôle de la taille moyenne du tampon : à travers ce contrôle, RED permet de contrôler et minimiser le délai.
- Équité : d'après les auteurs dans (Floyd et Jacobson, 1993), l'objectif d'assurer l'équité à travers RED n'est pas bien défini. Cependant, par l'intermédiaire de l'algorithme d'effacement aléatoire des paquets, RED essaye d'éviter qu'une ou plusieurs connexions ne subissent plus de pertes que les autres.

Pour réaliser ces objectifs, RED exécute un algorithme à chaque arrivée de paquet. L'algorithme permet de calculer la taille moyenne du tampon puis de la comparer aux deux seuils, le seuil minimal et le seuil maximal (*minimum threshold* et *maximum threshold*). Ces deux seuils sont définis d'avance. Une fois la taille moyenne calculée *Avgq*, RED la compare à ces deux seuils. Si *Avgq* est inférieure au seuil minimal le paquet est admis dans la file d'attente ; si elle est comprise entre le seuil minimal et le seuil maximal, RED calcule une

probabilité *p* selon laquelle le paquet sera soit marqué soit effacé Concernant le marquage de paquets, il y a principalement deux méthodes qui sont utilisées par RED pour signaler une congestion : soit l'effacement de paquets, soit l'application de la méthode ECN (*Explicit Congestion Notification*) qui permet d'ajouter un bit d'indication de congestion dans l'en-tête du paquet, au niveau IP ou TCP.

Donc comme on l'a déjà mentionné, l'exécution de l'algorithme RED inclut le calcul des deux variables *Avgq* et *p*. La variable *Avgq* est calculée comme suit :

$$Avgq \leftarrow (1 - w_q)\,Avgq + w_q.q \text{ (Floyd et Jacobson, 1993)}$$

w_q est une petite constante qui représente le poids de la file et elle est définie d'avance et q est la taille instantanée du buffer.

La probabilité *p* est calculée en fonction de *Avgq*. Quand *Avgq* varie entre le seuil minimal et le seuil maximal, *p* varie linéairement entre 0 et une valeur maximale max_p définie à l'avance :

$$p \leftarrow \max_p (Avgq - \min thresh) / (\max thresh - \min thresh) \text{ (Floyd et Jacobson, 1993)}$$

$\min thresh$: Le seuil minimal.

$\max thresh$: Le seuil maximal.

L'utilisation de RED exige alors le choix des paramètres suivants : $\min thresh$, $\max thresh$, $\max_p$ et w_q. Dans (Floyd et Jacobson, 1993), les auteurs ont montré la sensibilité des performances de RED à la valeur de ces dernières variables. Cependant, la définition des valeurs dépendra toujours de la nature du trafic, de la topologie et d'autres paramètres tels que le délai de la file d'attente souhaité.

3.4.2 Simulations et analyse des résultats:

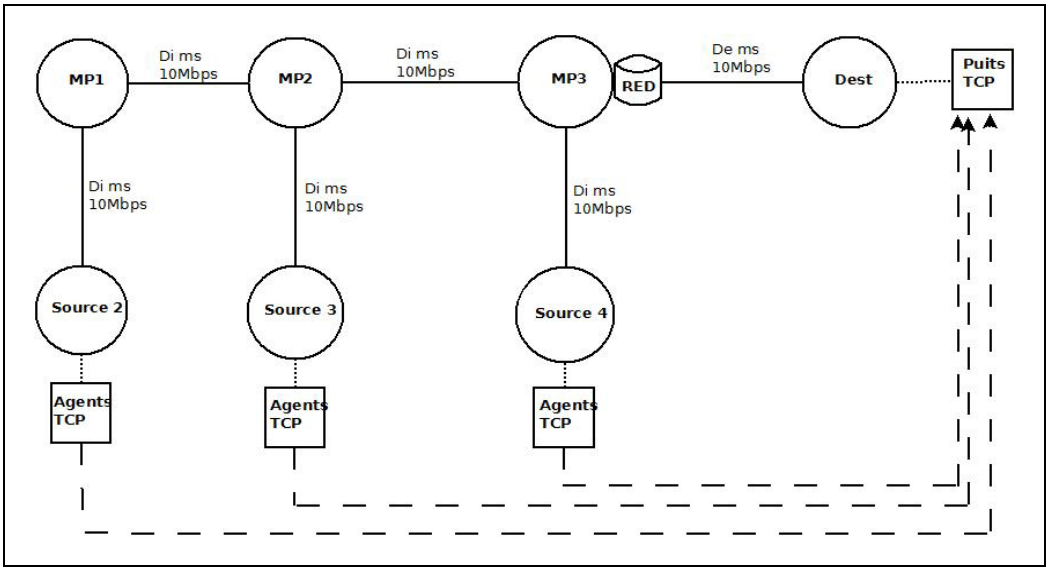

Figure 3.11 Topologie simulée dans le cas de RED.

La Figure 3.11 illustre la topologie simulée dans le cadre de RED. Il s'agit de la même topologie que celle utilisée avec *DropTail* sauf qu'on a éliminé la source 1 puisque son comportement est le même que celui de la source 2. Dans ce scénario aussi, le lien qui lie MP_3 et le serveur constitue un goulot d'étranglement. Le routeur MP_3 est muni d'une file d'attente RED. Chaque source transmet 5 flux TCP d'une manière continue, c'est-à-dire qu'il y a toujours des paquets TCP à transmettre. La capacité de tous les liens est de 10 Mbps. Pour la suite on appellera le délai du lien entre MP_3 et le serveur délai externe, D_e, alors que le délai de tous les autres liens sera nommé délai interne, D_i.

A travers ce scénario on essaie de voir comment se manifeste l'équité de TCP et comment se fait le partage de la bande passante. Le but est de voir la capacité de RED à améliorer l'équité de TCP et de voir l'influence des différents délais externes et internes ainsi que du paramètre *MaxThresh* de RED sur cette équité dans le cadre de tampons de petites tailles. Dans notre

scénario la taille maximale du tampon est de 10 paquets, pour éviter d'ajouter trop de délai par saut :

- 1er cas : Dans ce cas on considère que tous les délais, externes et internes, sont égaux. On étudie comment se fait le partage de la bande passante en illustrant le débit des différents flux en fonction du *MaxThresh*. Chaque courbe correspond à une valeur de délai (1 ms, 3 ms et 5 ms). Le but est donc de voir l'influence du délai et du *MaxThresh* sur l'équité.

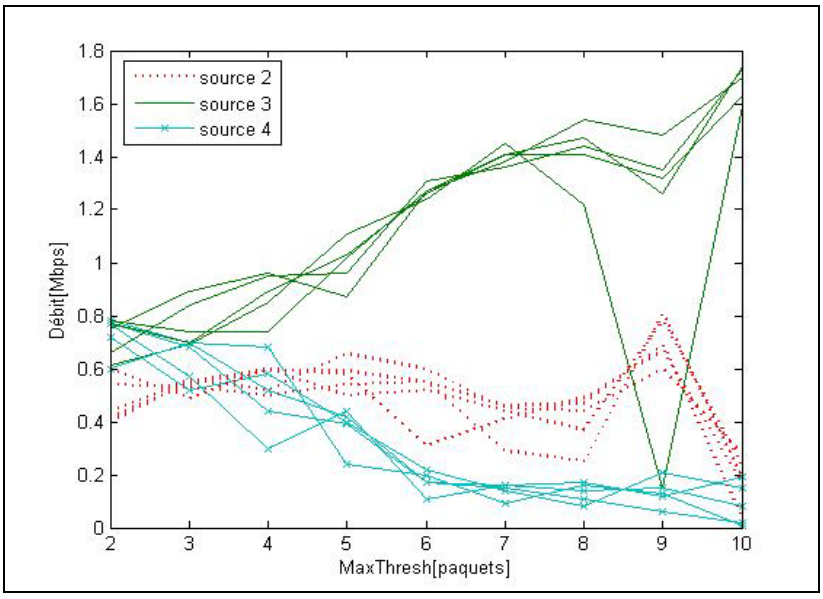

Figure 3.12 Débit en fonction du *MaxThresh* [D_i=1ms, D_e=1ms].

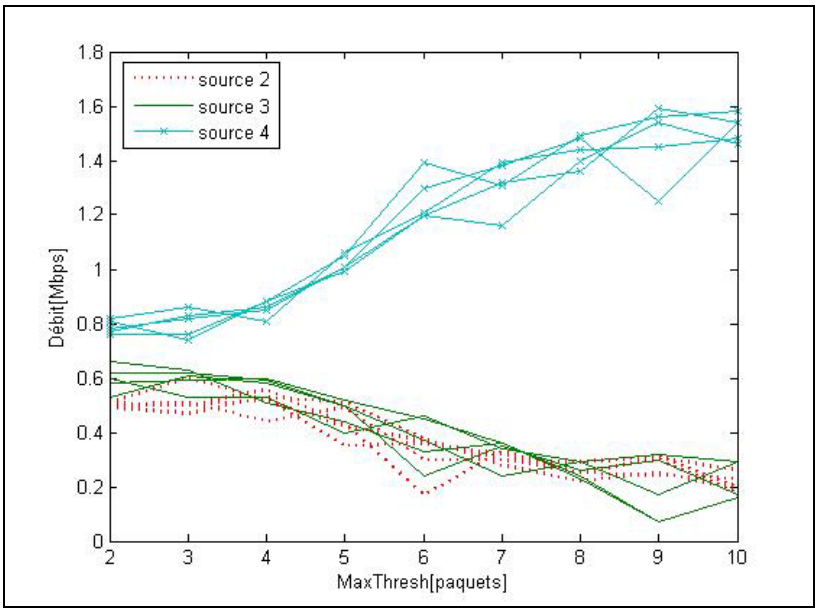

Figure 3.13 Débit en fonction du *MaxThresh* [D_i=3 ms, D_e=3 ms].

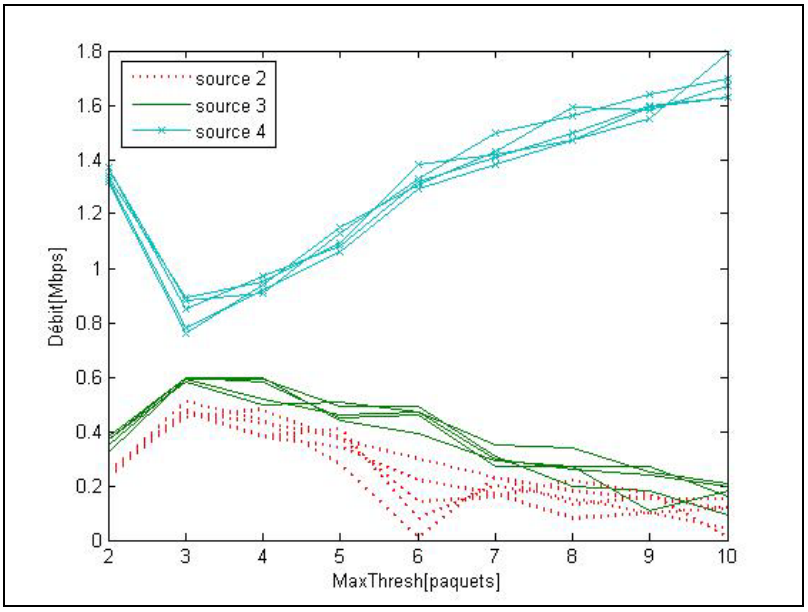

Figure 3.14 Débit en fonction du *MaxThresh* [D_i=5 ms, D_e=5 ms].

Les courbes dans la Figure 3.13 et la Figure 3.14 nous montrent que le comportement global est presque le même. Plus précisément, dans ces deux cas, on remarque que la source 4 qui est la source la plus proche du serveur s'empare de plus de bande passante que les deux

autres sources. Pour ces deux cas, d'une manière générale, la source ayant un moindre nombre de sauts vers le serveur profite de plus de bande passante. Par contre, pour un délai de 1 ms, comme représenté par la Figure 3.12, bien que la source 4 soit la source la plus proche du serveur, elle se voit perdre de la bande passante au profit de la source 3. Nous reviendrons plus loin sur ce phénomène. De point de vue du délai, on remarque qu'en augmentant le délai l'équité se détériore, surtout pour les petites valeurs de MaxThresh.

On remarque aussi que, pour les trois cas, la valeur de *MaxThresh* influe aussi sur l'équité. En effet, on peut distinguer une valeur de *MaxThresh* pour laquelle l'équité est meilleure. Dans les trois figures, cette valeur de *MaxThresh* est égale à 3. D'autre part, on remarque aussi qu'en augmentant la valeur de *MaxThresh* l'équité se détériore : pour la Figure 3.13 et la Figure 3.14, en augmentant la valeur de *MaxThresh*, c'est la source 4 qui s'empare de plus en plus de bande passante. Dans le cas de la Figure 3.12 où le délai est égal à 1 ms, c'est la source 3 qui en bénéficie. Normalement, la source la plus proche du serveur, donc celle qui a le RTT le plus faible, voit sa fenêtre de congestion de TCP croître plus rapidement que les autres. L'effacement des paquets avec RED étant aléatoire selon une probabilité qui augmente proportionnellement à la taille moyenne de la file d'attente, en observant la Figure 3.13 et la Figure 3.14, nous constatons que, malgré l'utilisation de RED, la source 4 qui est la plus proche du serveur gagne plus de bande passante que les autres avec l'augmentation de la valeur de *MaxThresh*. A partir de la Figure 3.15 qui correspond à la taille instantanée de la file d'attente pour une valeur de *MaxThresh* égale à 10, on constate que la file atteint sa capacité maximale (10 paquets) pendant la majorité de la simulation. Dans le cas où la valeur *MaxThresh* est égale à 2, on constate à partir de la figure 6 que la file atteint rarement sa capacité maximale. En faisant le lien avec la Figure 3.13, on pourra conclure qu'en augmentant la valeur de *MaxThresh*, on tend vers un comportement similaire à celui observé avec des files d'attente *DropTail*. Puisque pour une valeur de *MaxThresh* égale à 2, l'équité est meilleure que celle dans le cas d'une valeur de *MaxThresh* égale à 10. Donc, avec ces délais, 3 ms et 5 ms, à partir d'une certaine valeur de *MaxThresh*, on a le même comportement qu'avec des files d'attente *DropTail*. C'est-à-dire que les paquets des sources

2 et 3 vont trouver la file d'attente pleine et seront effacés pas seulement sous l'effet de RED mais suite au dépassement de capacité de la file (*Buffer overflow*).

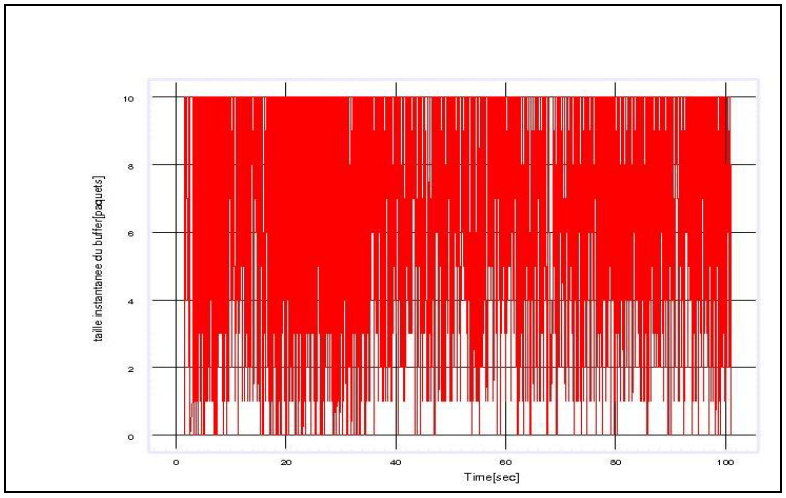

Figure 3.15 Taille instantanée du tampon [D_i=3 ms, D_e=3 ms, *MaxThresh*=10].

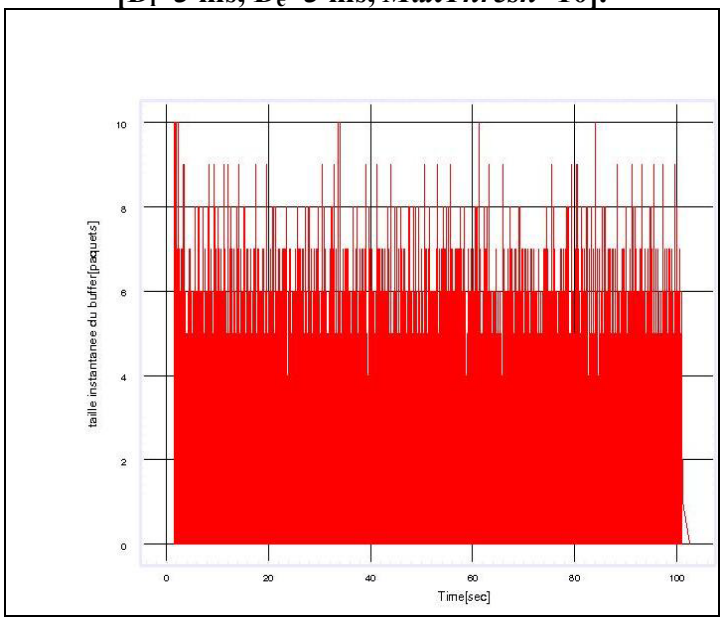

Figure 3.16 Taille instantanée du tampon [D_i=3 ms, D_e=3 ms, *MaxThresh*=2].

Pour le cas où le délai est de 1 ms, à partir de la Figure 3.12 on constate que l'allure générale est un peu similaire à celle de la Figure 3.13 et la Figure 3.14. Donc du point de vue de

l'équité, on constate qu'elle est meilleure pour des petites valeurs de *MaxThresh* et qu'elle se détériore en augmentant la valeur de *MaxThresh*. L'observation de la Figure 3.17 et la Figure 3.18 nous mènent à l'analyse déjà mentionnée concernant le cas où le délai est de 3 ms et de 5 ms. C'est-à-dire que, pour les valeurs de *MaxThresh* où la taille du tampon n'atteint pas la taille maximale, RED réussit à améliorer relativement l'équité à travers l'effacement des paquets aléatoirement selon une probabilité qui dépend de la taille moyenne du tampon. Mais pour les valeurs de *MaxThresh* où l'occupation du tampon atteint souvent la taille maximale, les pertes seront la plupart du temps dues au dépassement de la capacité du tampon, et par suite, on se retrouve à la même situation où on utilise *DropTail*.

Cependant, dans le cas où le délai est de 1 ms, le phénomène qui apparait et qui est différent des autres cas est que la source 4, étant la source la plus proche du serveur, perd de la bande passante au profit des autres sources comme le montre la Figure 3.12. La Figure 3.19 qui montre le taux de perte des flux des différentes sources en fonction du *MaxThresh* pour un délai de 1 ms, nous montre que les taux de perte des différentes sources est le même pour les plus petites valeurs de *MaxThresh* ce qui est conforme au niveau d'équité montré sur la Figure 3.12. En augmentant la valeur de *MaxThresh*, le taux de perte augmente pour la source 4 et diminue pour les deux autres sources. La Figure 3.20 et la Figure 3.21 qui illustrent l'évolution des fenêtres de congestion d'un flux de chacune des sources dans le cas d'une valeur de *MaxThresh* égale à 2 et 10 respectivement. On remarque remarquer que la fenêtre de congestion du flux TCP généré par la source 4 dans le cas de *MaxThresh* égal à 1 arrive à s'ouvrir et elle est presque au même niveau des autres sources. Pour le cas où *MaxThresh* est égal à 10 la fenêtre de congestion est constamment égale à 2. Donc un délai de 1ms est un délai qui va conduire à des pertes constantes pour la source 4.

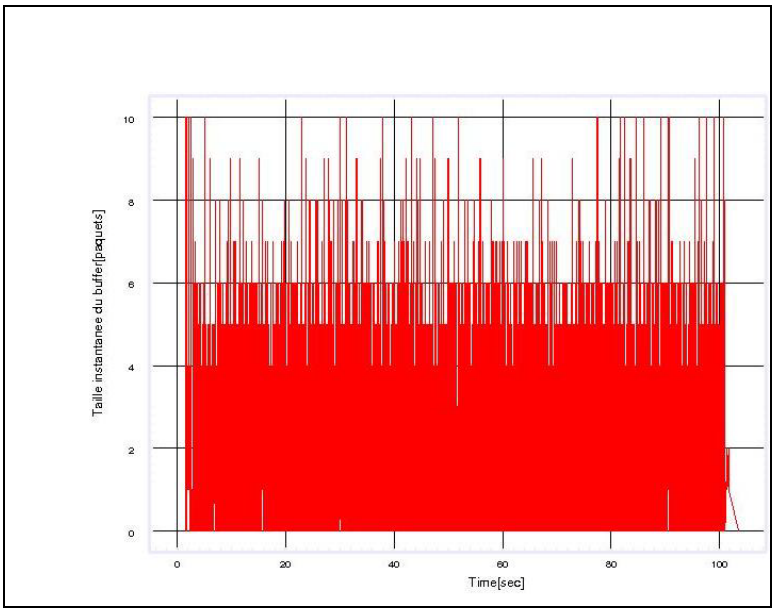

Figure 3.17 Taille instantanée du tampon [D_i=1 ms, D_e=1 ms, *MaxThresh*=2].

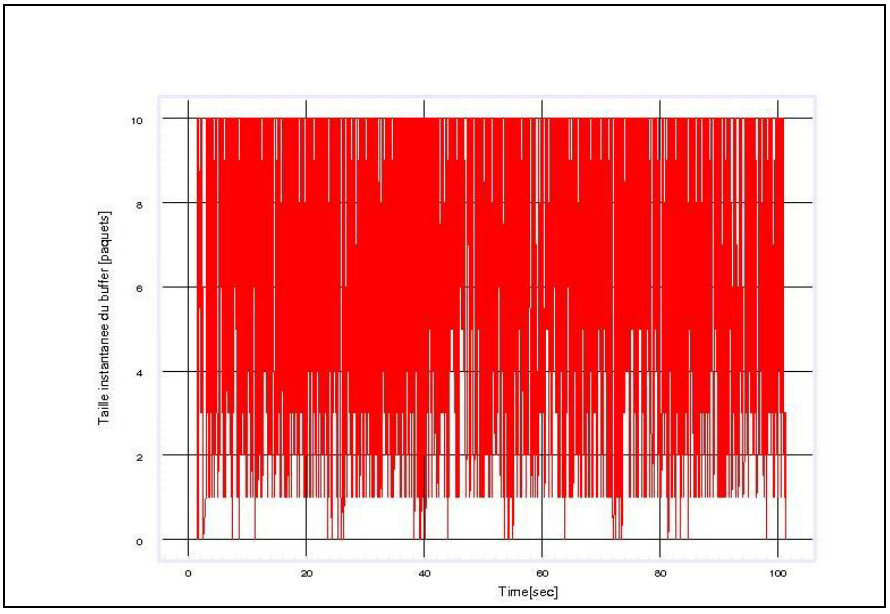

Figure 3.18 Taille instantanée du tampon [D_i=1 ms, D_e=1 ms, *MaxThresh*=10].

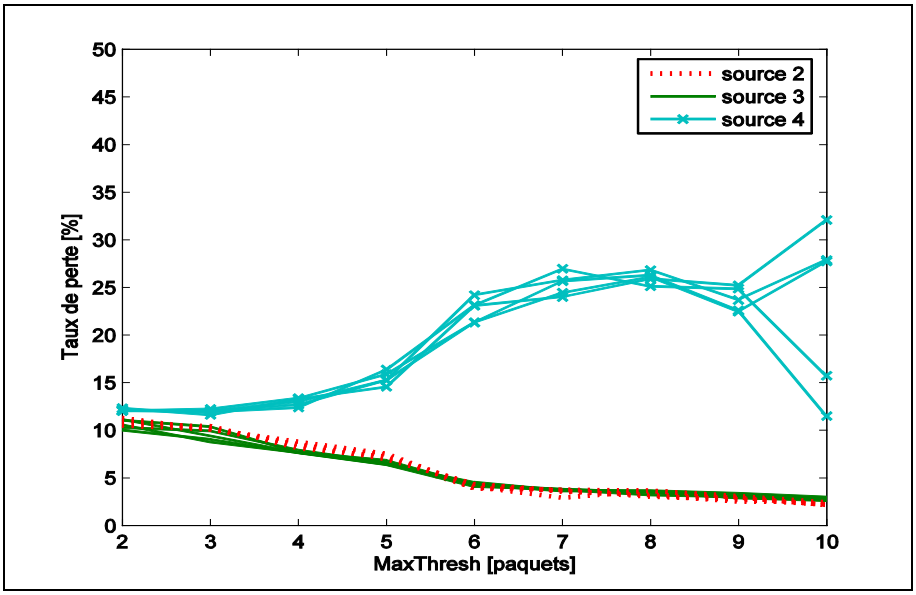

Figure 3.19 Taux de perte en fonction de *MaxThresh* [D_i=1 ms, D_e=1 ms].

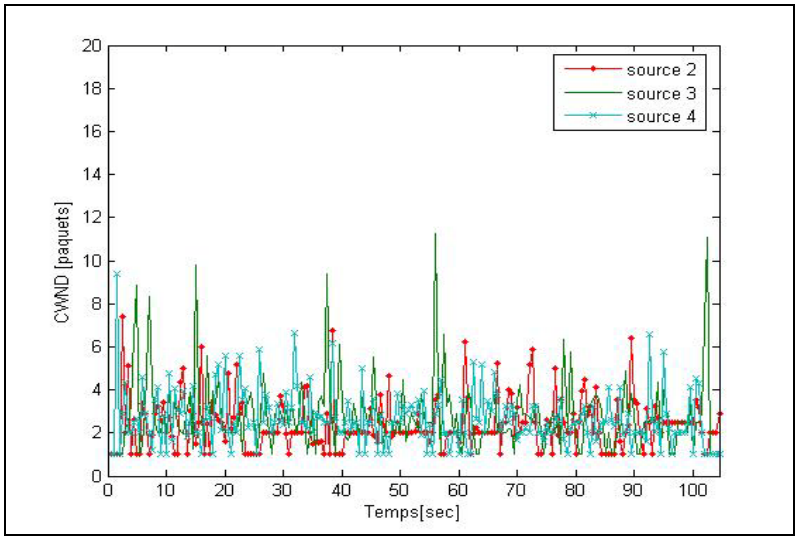

Figure 3.20 Évolution de la fenêtre de congestion [D_i=1 ms, D_e=1 ms, *MaxThresh*=2].

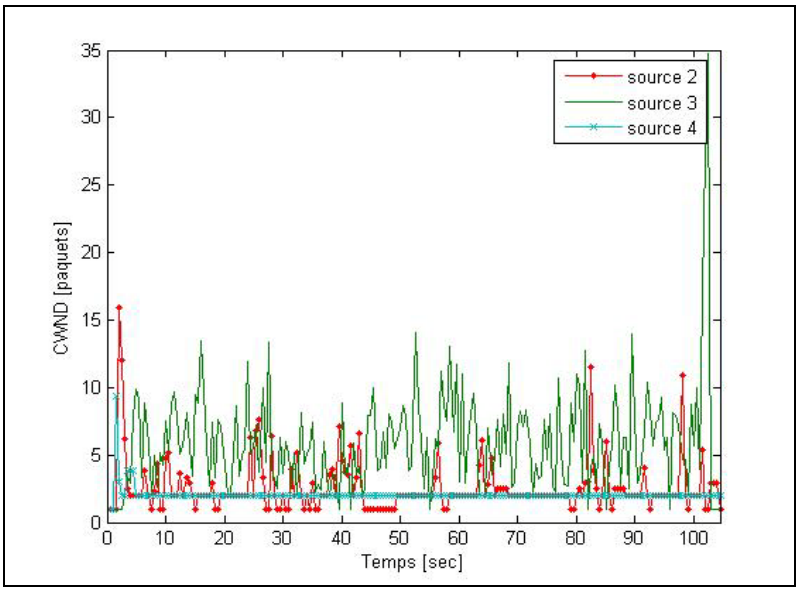

Figure 3.21 Évolution de la fenêtre de congestion
[D_i=1 ms, D_e=1 ms, *MaxThresh*=10].

- 2ème cas : Dans ce cas on désire voir l'effet du délai externe sur l'équité TCP. On fixe alors le délai externe à 50 ms et on fait l'étude pour les 3 valeurs de délais internes 1 ms, 3 ms et 5 ms.

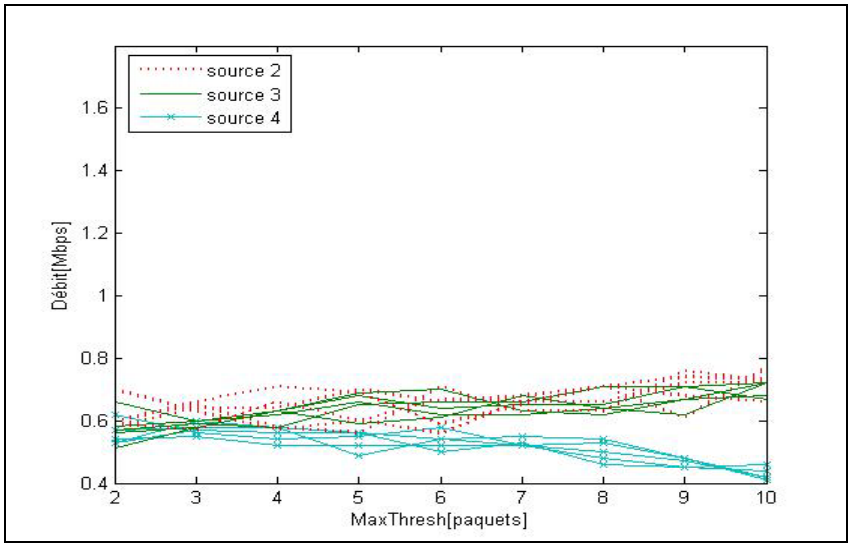

Figure 3.22 Débit en fonction du *MaxThresh*
[D_i=1 ms, D_e=50 ms].

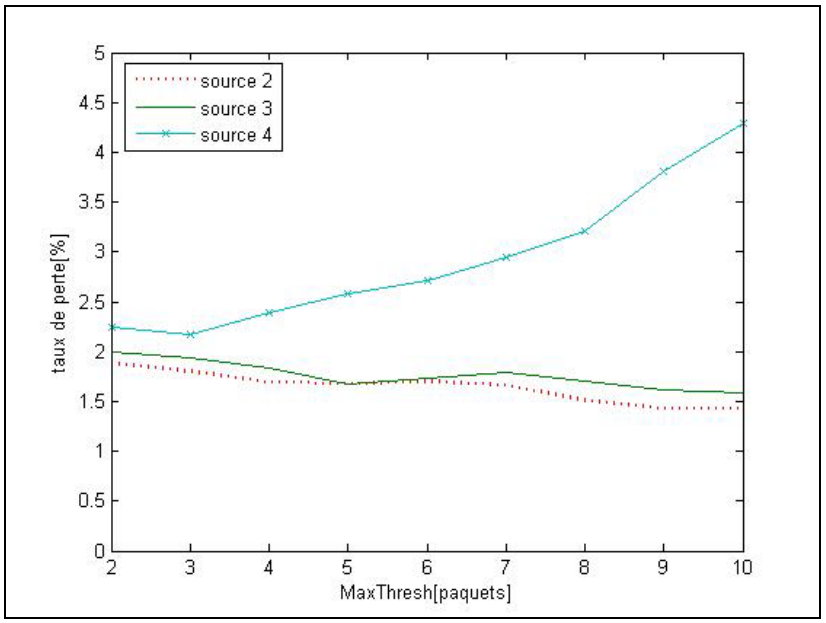

Figure 3.23 Taux de perte en fonction du *MaxThresh* [D_i=1 ms, D_e=50 ms].

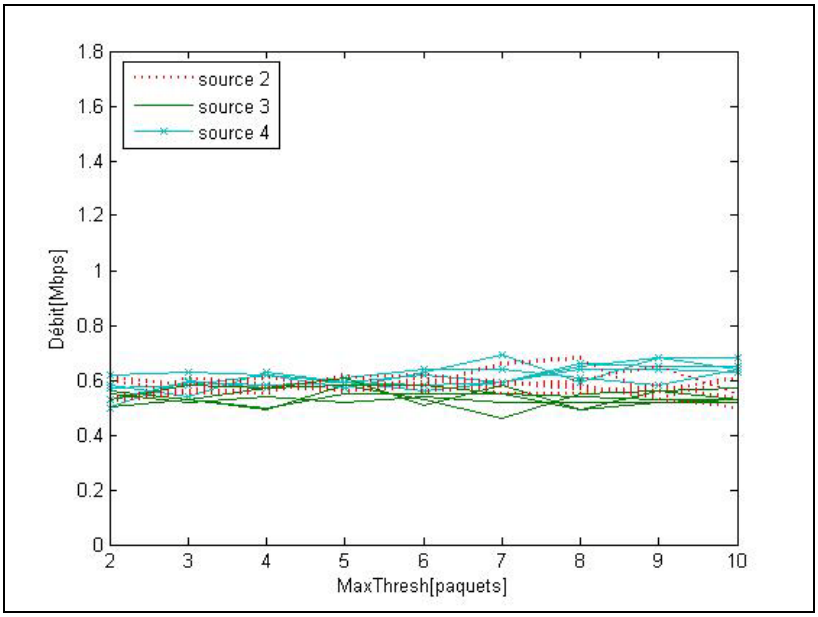

Figure 3.24 Débit en fonction du *MaxThresh* [D_i=3 ms, D_e=50 ms].

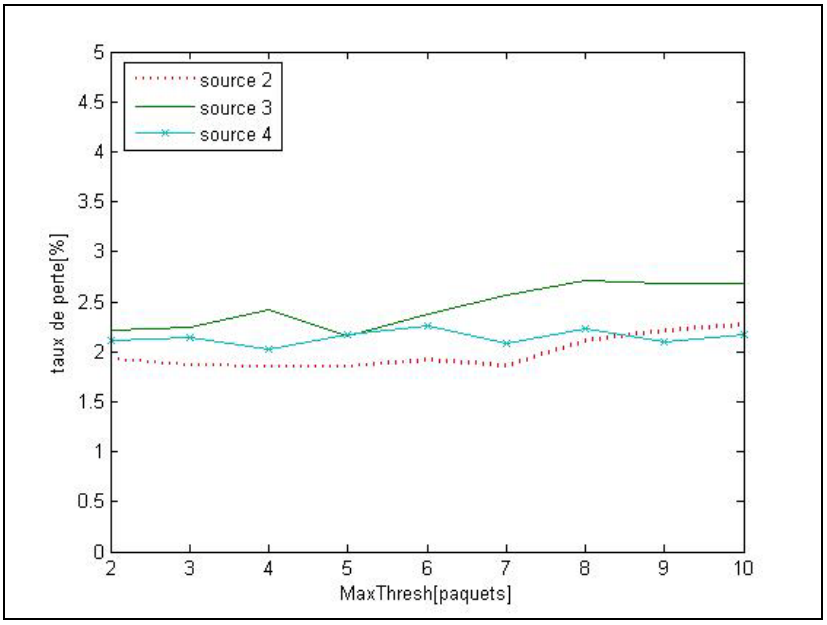

Figure 3.25 Taux de perte en fonction du *MaxThresh* [D_i=3 ms, D_e=50 ms].

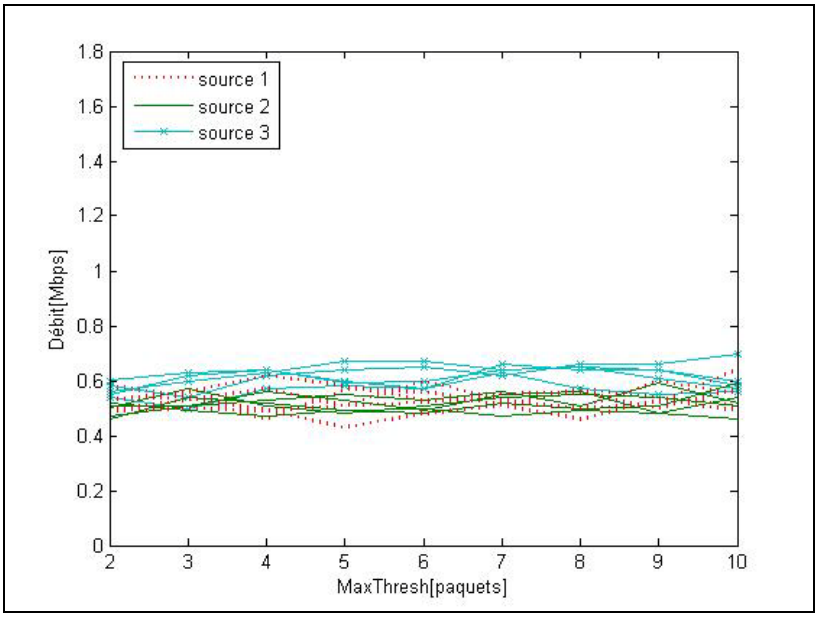

Figure 3.26 Débit en fonction du *MaxThresh* [D_i=5 ms, D_e=50 ms].

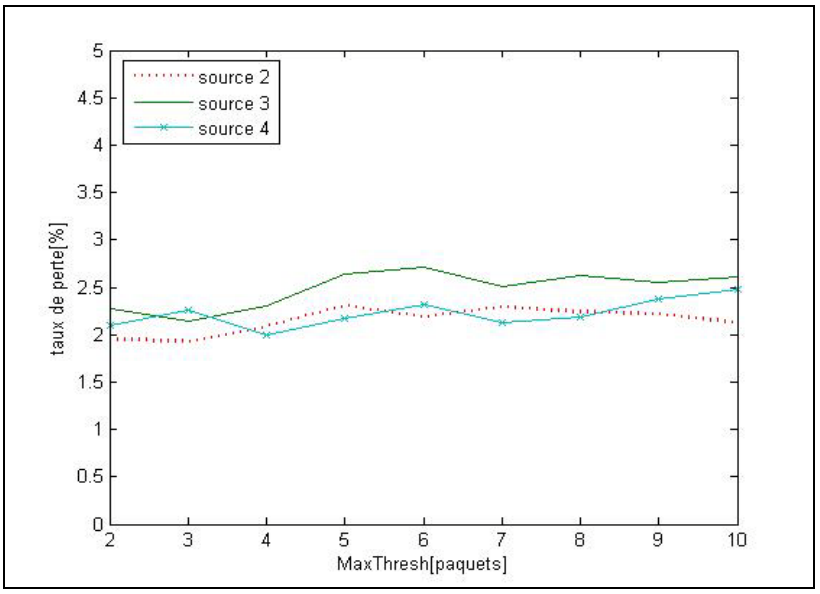

Figure 3.27 Taux de perte en fonction du *MaxThresh* [D_i=5 ms, D_e=50 ms].

A partir de la Figure 3.22, la Figure 3.24 et la Figure 3.26, on voit qu'en augmentant le délai externe, qui devient assez élevé par rapport aux délais internes, on a presque une équité entre les flux des différentes sources. Cette équité se réalise avec les 3 valeurs de délai internes à savoir 1 ms, 3 ms et 5 ms. Cependant, le degré d'équité varie en fonction des délais internes. Avec un délai de 1ms, à partir de la Figure 3.22, on remarque que l'équité commence à se détériorer à partir d'une certaine valeur de *MaxThresh* où la source 4 commence à perdre de la bande passante au profit des autres sources. Cette perte est due au taux de perte qui devient plus important pour la source 4 par rapport aux autres sources comme l'illustre la Figure 3.23. À partir de la Figure 3.24 et la Figure 3.26, on remarque que le passage du délai interne de 3 ms à 5 ms affecte légèrement l'équité. La question qui se pose à ce niveau est si RED assure cette équité ? Pour répondre à cette question voyons la Figure 3.28 et la Figure 3.29 ci-dessous qui illustrent la taille instantanée du tampon pour les valeurs de *MaxThresh* 2 et 10 respectivement, la valeur du délai interne étant de 3 ms.

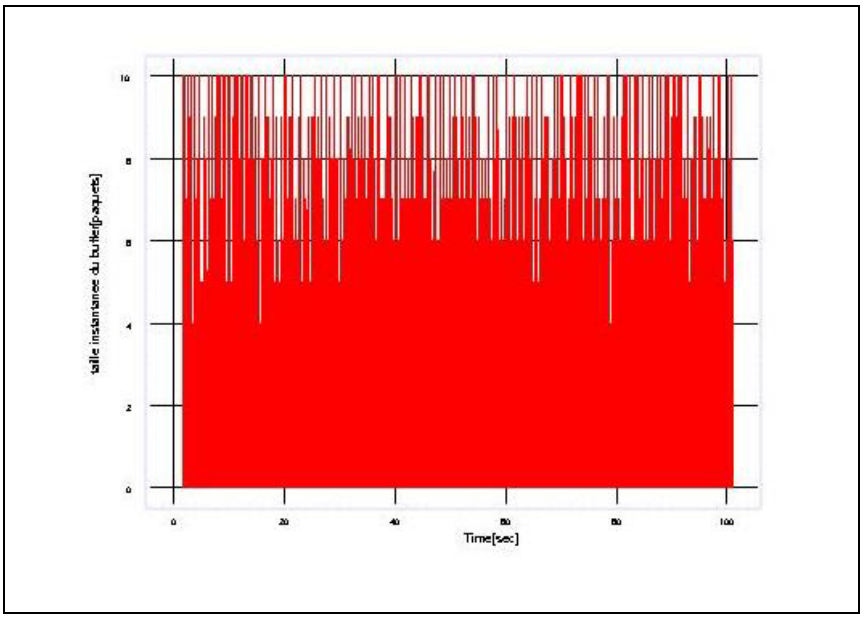

Figure 3.28 Taille instantanée du tampon [D_i=3 ms, D_e=50 ms, *MaxThresh*=2].

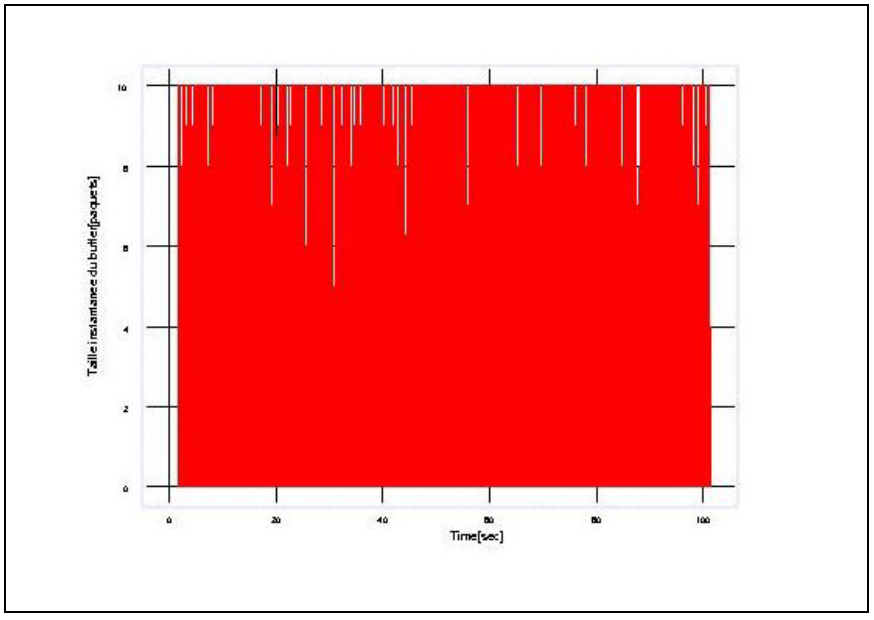

Figure 3.29 Taille instantanée du tampon [D_i=3 ms, D_e=50 ms, *MaxThresh*=10].

A travers les deux figures, on constate que le tampon est constamment rempli, surtout quand le *MaxThresh* est égal à 10. Donc les pertes à cause du dépassement de la taille du tampon

sont souvent présentes. Ces pertes ne sont donc pas relatives à la politique RED. Aura-t-on alors les mêmes résultats et les mêmes allures en utilisant un tampon de type *DropTail* ? Les résultats de la simulation en utilisant ce type de tampon de taille de 10 paquets et où le délai interne est de 3 ms sont illustrés dans le ci-dessous.

Tableau 3.1 Débit (Mbps) des flux des différentes sources

[D_i=3 ms,D_e=50 ms,*Droptail*]

	Flux 1	**Flux 2**	**Flux 3**	**Flux 4**	**Flux 5**
Source 2	0.58	0.55	0.57	0.63	0.56
Source 3	0.49	0.48	0.53	0.56	0.52
Source 4	0.62	0.56	0.67	0.62	0.66

Donc à partir du tableau ci-dessus, on remarque que le comportement est presque le même pour RED et *DropTail* dans le cas où le délai externe est de 50 ms et le délai interne est de 3 ms. Le fait d'avoir un délai externe élevé par rapport aux délais internes a permis que les différents RTTs correspondant aux différentes sources soient approximativement aussi grands l'un que l'aure et par suite la différence entre eux influe peu. Avec ce RTT assez grand, les fenêtres de congestion s'ouvrent plus lentement et presque de la même manière pour toutes les sources. Cette ouverture plus ou moins lente a permis d'éviter qu'une source qui est plus proche de la destination n'envoie beaucoup plus de paquets que les autres plus éloignées. Ainsi cela évite que les paquets des autres sources trouvent le tampon déjà plein à leur arrivée au goulot d'étranglement et par suite les pertes seront à peu près les mêmes. La Figure 3.25 et la Figure 3.27 nous permettent de remarquer que le taux de perte est presque le même pour toutes les sources, et que ce taux n'est pas très élevé.

Dans ce qui suit, on procède à des simulations où le délai interne est de 5 ms et les délais externes sont multiples de 5 ms pour s'assurer de l'influence du rapport D_e / D_i.

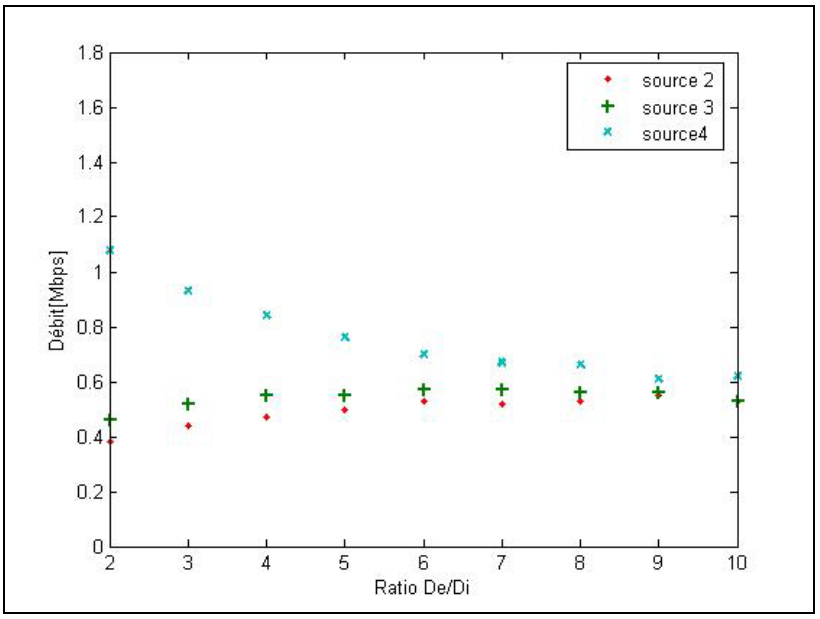

Figure 3.30 Débit en fonction du rapport D_e / D_i.

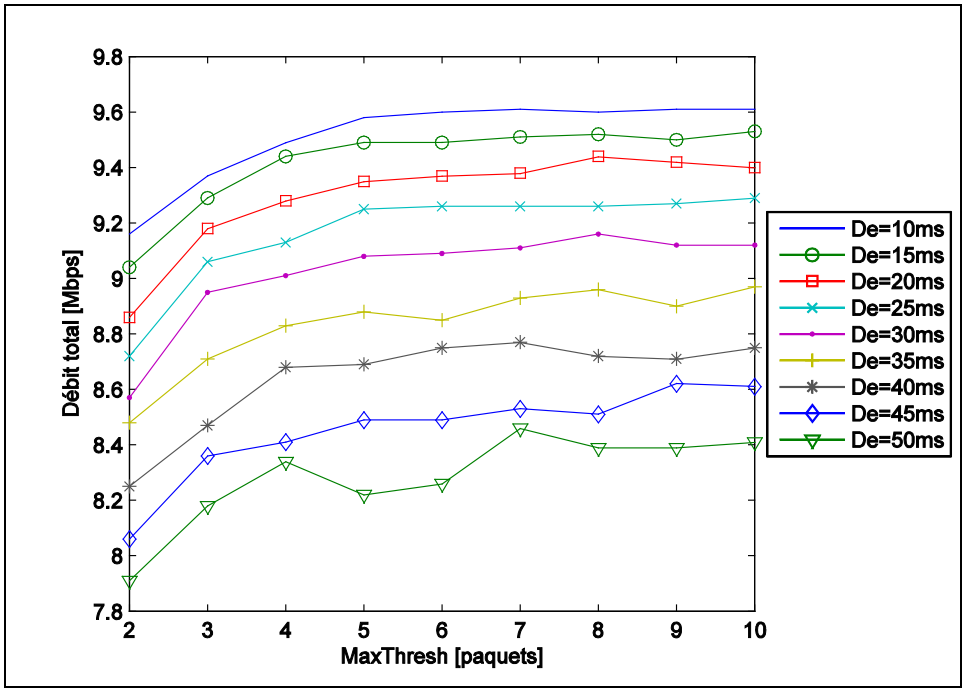

Figure 3.31 Débit total en fonction du MaxThresh.

La Figure 3.30 montre bien que l'équité s'améliore en augmentant le rapport D_e/D_i. Mais cette amélioration est obtenue au prix de l'utilisation du lien. Figure 3.31 nous montre que le débit total, l'utilisation du lien, diminue quand ce rapport augmente.

Conclusion

A l'issue de ce chapitre, on peut tirer plusieurs leçons concernant l'effet des délais et de la politique de gestion de files d'attente RED sur l'équité. En effet, on a pu constater que d'une manière générale et dans la plupart des cas, la source la plus proche du serveur est la source qui s'empare de plus de bande passante que les autres.

D'autre part, en augmentant le délai interne, l'équité se détériore. L'utilisation d'un délai externe assez élevé par rapport aux délais internes permet de minimiser l'influence de la différence entre les RTTs et par suite toutes les sources auront un taux de perte plus ou moins égal et on évite ainsi que les paquets des sources les plus éloignées trouvent le tampon déjà rempli par les paquets de la source la plus proche. Enfin, on a vu que l'utilisation de *DropTail* avec un délai externe de 50 ms et un délai interne de 3 ms donne le même niveau d'équité obtenu en utilisant RED et donc que la contribution de RED n'est pas toujours significative dans ce scénario.

Le débit de TCP est sensible au RTT et au taux de perte. Dans ce type de réseaux ou de topologie on peut avoir des sources qui auront moins de pertes et un RTT plus faible que les autres. L'utilisation de RED n'a pas empêché que certains flux monopolisent la file d'attente alors que d'autres sont privés de place. RED qui vise la détection prématurée de la congestion, et qui se base sur l'effacement des paquets selon une probabilité qui est une fonction croissante de la taille moyenne du tampon, ne permet pas de résoudre le problème d'équité puisqu'il n'y a pas de mécanisme d'effacement différencié.

CHAPITRE 4

DIFFSERV

4.1 Introduction

Après l'augmentation du nombre d'utilisateurs d'Internet et l'apparition de plusieurs variétés d'applications qui exigent différentes garanties de qualité de service, le service Best Effort qu'offre Internet, où toutes les applications subissent le même traitement, n'est plus alors suffisant. Des mécanismes qui doivent assurer des niveaux de QoS aux différents types d'application sont alors indispensables. Des recherches ont été menées pour trouver et ajouter des mécanismes permettant d'identifier les demandes des différents utilisateurs et de leur garantir le niveau de QoS qu'ils exigent en termes, par exemple, de garantie de bande passante, de latence, de gigue et de taux de perte.

L'architecture IntServ (Integrated Services) a été développée en 1993 par l'IETF avec l'objectif de prendre en charge la QoS sans toucher au fonctionnement de IP et garantir une QoS par flux. Il s'agit d'une architecture qui fournit une intégration de services afin, par exemple de supporter les applications temps réel ainsi que les applications non temps réel. IntServ est un mécanisme basé sur la réservation de ressources. Et il utilise pour cela une signalisation explicite (hors bande) à travers un protocole de réservation qui est le RSVP (Resource reSerVation Protocol) qui effectue et gère la réservation de ressources.
IntServ qui fournit un traitement par flux convient à des réseaux de petites tailles mais pour Internet, où il y a un très grand nombre de nœuds et par suite de flux, il présente des problèmes de mise en échelle (scalability). Pour palier aux problèmes et limites d'IntServ, et pour donner aux fournisseurs de services la possibilité de proposer différents types de services, l'IETF a développé en 1997 l'architecture DiffServ (Blake *et al.*, 1998). Cette architecture, contrairement à IntServ, fournit un traitement par agrégat de flux, une

signalisation implicite et elle affecte les tâches complexes aux routeurs de bordure pour alléger les routeurs du cœur du réseau.

4.2 Les principes de DiffServ

L'architecture DiffServ (Blake *et al.*, 1998) permet de rassembler les flux ayant les mêmes caractéristiques et exigences en termes de QoS dans la même classe. On parle alors d'un traitement d'agrégats de flux. Chaque agrégat de flux va correspondre à une classe de service. DiffServ permet alors d'attribuer un comportement spécifique à chaque classe de service suivant ses exigences en termes de QoS. La différenciation de services peut porter sur la priorité des paquets ou sur tout autre critère. Le modèle DiffServ permet de rassembler les flux dans des différentes classes de services, et chaque classe sera identifiée par un code. Donc tous les paquets d'une classe de service auront le même code et le traitement subi par le paquet dépend de ce code. Ce code sera présent dans le paquet IP et il s'agit du champ TOS pour la version IPv4, et *class of service* et *flow label* pour la version IPv6. Un des principes de DiffServ est qu'il n y a pas de réservation de bande passante ni de contrôle d'admission, mais plutôt ce qu'on appelle un SLA (Service Level Agreement) qui établit un contrat entre l'utilisateur et le fournisseur d'accès. DiffServ distingue aussi entre les routeurs de bordure et les routeurs du cœur du réseau. Le concept est d'affecter les tâches complexes aux routeurs de bordure et non pas aux routeurs du cœur afin d'alléger le réseau cœur.

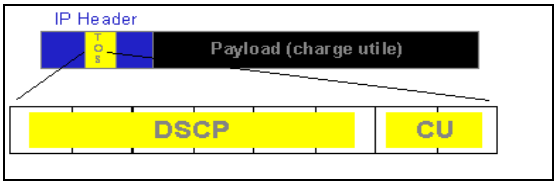

Figure 4.1 Le champ DSCP.

Le champ TOS, utilisé pour assurer le traitement différencié, est de 8 bits. Le DSCP utilise les 6 bits de poids fort et les deux autres bits ne sont pas utilisés. Les 3 premiers bits (de

poids fort) constituent les sélecteurs de classes CSC (Class Selector Code point). Les bits de 3 à 5 permettent d'étendre les CSC pour obtenir une granularité supplémentaire pour avoir 8 sous-classes par CSC. Comme on le verra plus tard, les routeurs du cœur traiteront les paquets en fonction de ce code DSCP selon un comportement spécifique appelé PHB (Per Hop Behavior). Donc chaque PHB est codé par un et un seul DSCP. Le PHB permet la différentiation de service. DiffServ a défini principalement 3 PHB :

- *Best Effort* (BE) : c'est la classe avec la priorité la plus basse et il s'agit du PHB par défaut. Le DSCP correspondant à ce PHB est 000000.
- *Assured Forwarding* (AF) : il est dédié à des services généraux et permet de garantir un acheminement des paquets IP avec une haute probabilité sans prendre en considération les délais. Le PHB AF est constitué de 4 classes de service, pour chacune une bande passante minimale est garantie. Il est à noter qu'il n y a pas de priorité entre ces classes. Par contre, chaque classe comprend 3 niveaux de priorité (*Drop Precedence*) selon que l'utilisateur respecte son contrat, le dépasse légèrement ou est largement en dehors. La notion de précédence permet de définir les priorités entre les paquets de la même classe. En cas de congestion dans une classe AF, les paquets de basse priorité sont rejetés en premier. Ainsi le service AF est composé de 12 PHB interdépendants qui sont identifiés par $AF_{i,j}$, où i est la classe (de 1 à 4) et j est la précédence (de 1 à 3). Les priorités sont généralement associées à des couleurs : rouge pour priorité haute, jaune pour une priorité moyenne et verte pour une priorité basse. Donc à son entrée le trafic est marqué par un marqueur (basé sur deux seaux à jeton par exemple) selon le contrat défini entre l'utilisateur et le fournisseur d'accès.

Tableau 4.1 les différents codes DSCP pour AF

Drop Precedence	**Classe 1**	**Classe 2**	**Classe 3**	**Classe 4**
Basse	AF11 001010	AF21 010010	AF31 011010	AF41 100010
Moyenne	AF12 001100	AF22 010100	AF32 011100	AF42 100100
Haute	AF13 001110	AF23 010110	AF33 011110	AF43 100110

- *Expedited Forwarding* (EF): ce PHB correspond au DSCP 101110 ; il est destiné aux applications temps réel. Il permet une garantie de bande passante avec des taux de perte, de délai et de gigue faibles. Le contrat établi porte sur un débit constant. Au contraire du AF, les paquets non conformes au contrat sont rejetés. Afin d'obtenir un délai faible et que le service soit performant, il faut qu'un pourcentage réduit du trafic total utilise ce PHB afin qu'aucun paquet EF ne soit rejeté dans le cœur du réseau, que les paquets soient mis dans une file séparée qui est servie avec une plus haute priorité (PQ), et que le débit d'entrée dans les nœuds du cœur soit inférieur au débit de sortie afin d'assurer une bande passante minimale.

4.3 Architecture DiffServ

Nous présentons ici l'architecture DiffServ (Blake *et al.*, 1998) et décrivons le rôle de chaque composant de cette architecture. L'architecture DiffServ distingue entre deux types de routeurs : les routeurs de bordure (Edge Router) qui sont les routeurs d'entrée et de sortie du domaine DiffServ et les routeurs du cœur (core router). Il est à noter qu'une région DiffServ

est un ensemble de domaines reliés à travers les routeurs de bordures. Le concept de DiffServ est d'attribuer les tâches complexes aux routeurs de bordure, alors que les routeurs du cœur auront la tâche de traiter les paquets selon le contenu du champ DSCP minimisant ainsi le temps de traitement de ces routeurs.

4.3.1 Routeur de bordure

Le routeur de bordure, comme le montre la Figure 4.2 est composé de différents modules. Chaque module est responsable de différentes tâches. Ce type de routeur est responsable de la classification des paquets (agrège les flots en classes de service), du conditionnement du trafic et du marquage des paquets (attribuer le DSCP) :

- Classification: cette opération est effectuée à l'entrée du paquet au réseau DiffServ. Elle s'effectue selon un ou plusieurs champs de l'en-tête IP.
- Mesure : cette opération permet de vérifier la conformité du profil du trafic entrant avec le contrat négocié. Après l'opération de classification, les paquets passent par le module *Meter* afin de vérifier et déterminer le niveau de conformité de ces paquets avec le contrat. Suite au résultat de la vérification de conformité, les paquets conformes ou *IN* seront passés pour être étiquetés, les autres, *OUT* seront soit éliminés (pour les applications interactifs qui ne supportent pas un délai grand) soit ils seront conditionnés (mise en forme) et après marqués. Ces derniers seront les premiers à être rejetés en cas de congestion.
- Conditionnement : les paquets non conformes et qui n'ont pas été rejetés, passent par le module *Shaper* pour leurs rendre conformes en retardant leur acheminement. Après être devenus conformes, ces paquets seront envoyés au module Marker pour être étiquetés.
- Marquage : il s'agit de la dernière étape avant d'éjecter les paquets dans le réseau. Cette opération effectuée par le module *Marker* permet d'étiqueter les paquets et leurs affecter le DSCP correspondant.

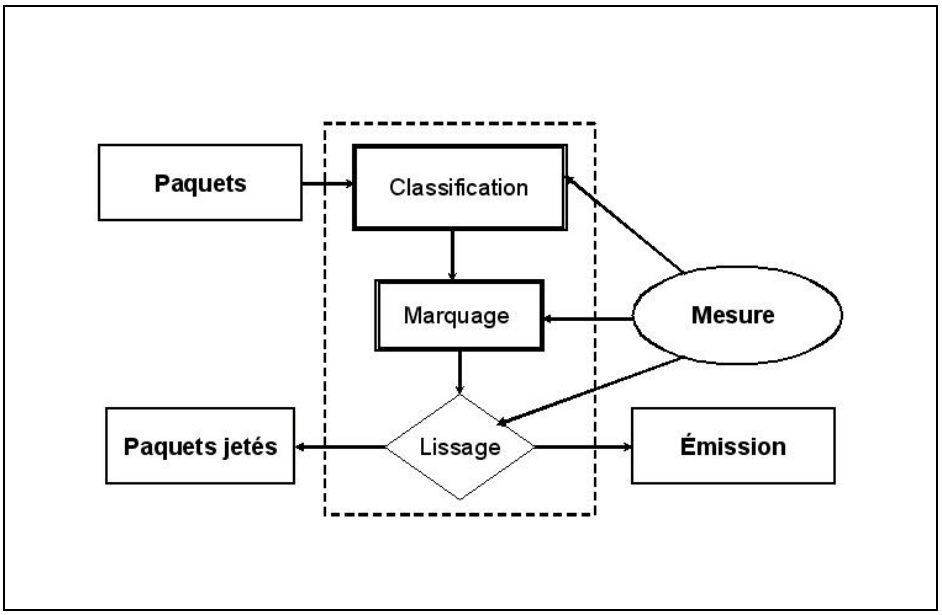

Figure 4.2 les différents modules du routeur de bordure.

4.3.2 Le routeur du cœur

Les tâches associées à ces routeurs sont plus simples que celles du routeur de bordure. Le routeur de cœur va traiter les paquets selon son PHB, qui se base uniquement sur le contenu du champ DSCP. Tous les paquets ayant le même DSCP doivent subir le même traitement. Ce type de routeur est responsable de deux tâches principales : la classification des paquets entrant et la gestion de la file d'attente. Il est constitué, comme le montre la Figure 4.3, d'un classificateur, un nombre de file d'attente, d'un ordonnanceur et d'un algorithme de gestion de file d'attente pour chaque file :

- La classification : cette opération est effectuée par le classificateur, elle est plus simple que celle effectuée au niveau des routeurs de bordure. Le but de cette opération est de différencier les paquets selon la valeur du DSCP. Le CSCP (Class Selector Code Point) permet d'identifier la classe de service correspondant à la catégorie du paquet alors l'autre partie du champ DSCP est utilisée par l'algorithme de gestion de file d'attente.

Chaque file d'attente caractérise une classe de service et ne recevra que les paquets qui sont conformes à ce service.

- L'ordonnancement : il a pour rôle de contrôler la distribution de ressources entre les différentes classes de service. Un des algorithmes d'ordonnancement (WRED, RIO, PQ, WRR) est utilisé pour effectuer cette tâche. Le choix d'un tel algorithme ou d'un autre dépend des besoins.

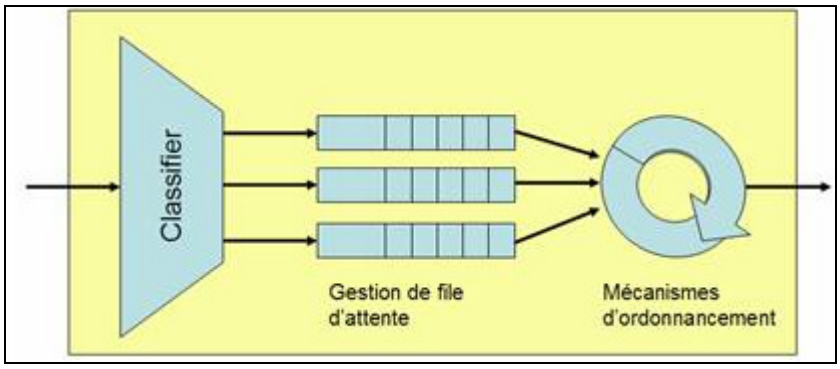

Figure 4.3 Les différents modules du routeur du cœur.

4.4 DiffServ et équité TCP :

Comme on a déjà vu à travers les résultats des simulations réalisées dans le chapitre précédent, les réseaux multi-saut, selon la topologie sur laquelle on a travaillé, présentent le problème d'équité entre les flux des différentes sources. D'une manière générale on a constaté que, plus un nœud est proche de la passerelle, plus il est favorisé en termes de bande passante. Bien que l'utilisation de RED permette d'améliorer les lacunes de *DropTail*, elle ne présente pas une bonne solution pour l'équité de TCP puisque RED efface aléatoirement les paquets et ne permet pas un effacement différencié. Dans le cadre de ce chapitre on va essayer d'appliquer Diffserv afin de voir s'il permet d'améliorer l'équité. Bien que DiffServ soit un mécanisme de QoS permettant de différencier entre classes de services en terme de leurs besoins en termes de qualité de service, on va essayer d'appliquer les principes de DiffServ afin d'aboutir à de meilleurs résultats en termes d'équité TCP. Ci-dessous on

présente les approches simulées. On a choisi deux grandes approches ou méthodologies pour les simulations.

4.4.1 Première approche

Ayant comme objectif d'assurer une équité entre les flux des 3 sources, nous allons à travers cette approche utiliser DiffServ pour réaliser cet objectif. Comme nous avons remarqué, la source 4 (edge 4) était la source qui procurait un débit supérieur aux autres puisque c'est la source la plus proche de la destination. Les deux sources les plus en aval par rapport à la source edge 4 sont caractérisées par un taux de perte supérieur à la source edge 4. Dans cette approche on va utiliser les principes du modèle DiffServ afin de différentier le traitement entre les trois sources. L'idée est d'affecter aux flux de chaque source un DSCP différent des autres. Et par la suite on va procéder à l'attribution d'une file à chaque DSCP. Réaliser l'équité passe alors par les paramètres attribués à chaque file ainsi que l'utilisation du bon ordonnanceur. La Figure 4.4 illustre la topologie simulée dans cette approche.

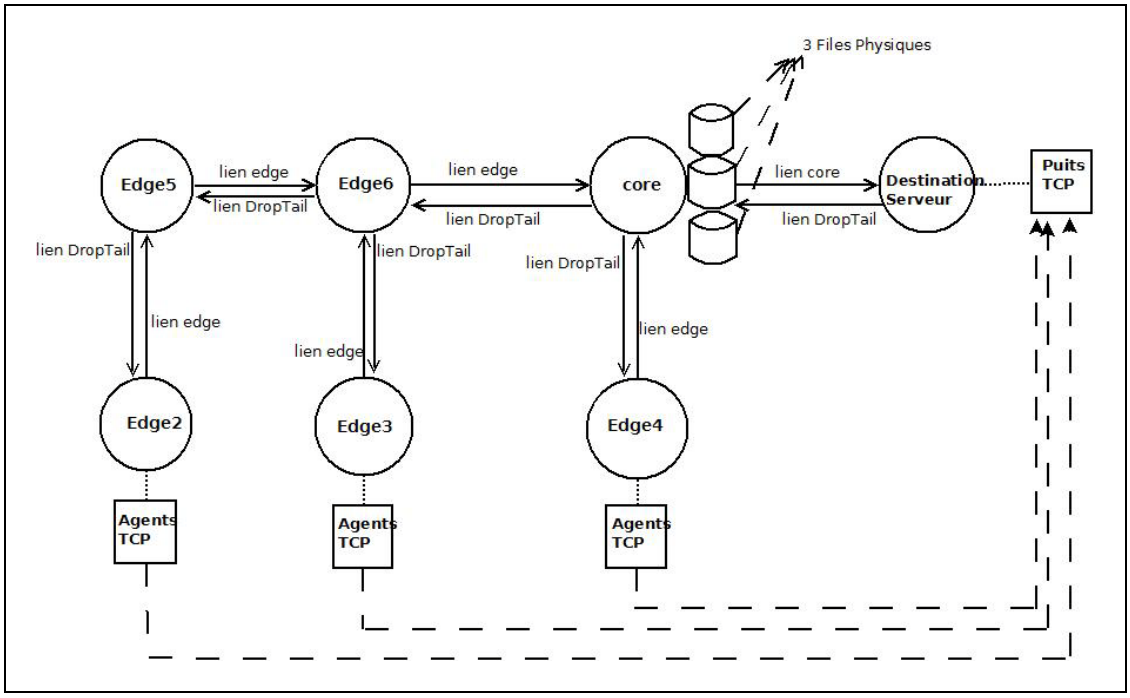

Figure 4.4 Topologie simulée avec DiffServ [3files d'attentes].

Le ci-dessous montre le résultat d'une simulation où on a fixé des paramètres RED plus sévères pour la file relative aux flux de la source 4 afin d'effacer d'avantage les paquets de cette source. L'algorithme d'ordonnancement utilisé est le WRR.

Tableau 4.2 Débit [Mbps] des flux des différentes sources

	4.5 Flux 1	**4.6 Flux 2**	**4.7 Flux3**	**4.8 Flux4**	**4.9 Flux 5**

4.10	**Source 1 (edge 2)**	0.67	0.62	0.67	0.65	0.61
4.11	**Source 2 (edge 3)**	0.68	0.65	0.63	0.68	0.64
4.12	**Source 3 (edge 4)**	0.64	0.60	0.68	0.68	0.58

Bien que cette approche permette d'aboutir à un résultat souhaitable en termes d'équité, comme le montre par exemple le résultat montré dans le tableau ci-dessus, qui est relatif à la configuration déjà présentée, elle semble être faible comme approche pour la raison suivante : dans notre cas on a seulement trois sources donc on aura besoin de trois DSCP différents et par suite de 3 files d'attente différentes. Mais dans un réseau de grande taille on a plusieurs sources et chaque source est à un nombre de sauts différents de beaucoup d'autres sources, en utilisant cette approche on aura besoin d'autant de files d'attente que de nombre de sauts différents.

4.12.1 Deuxième approche

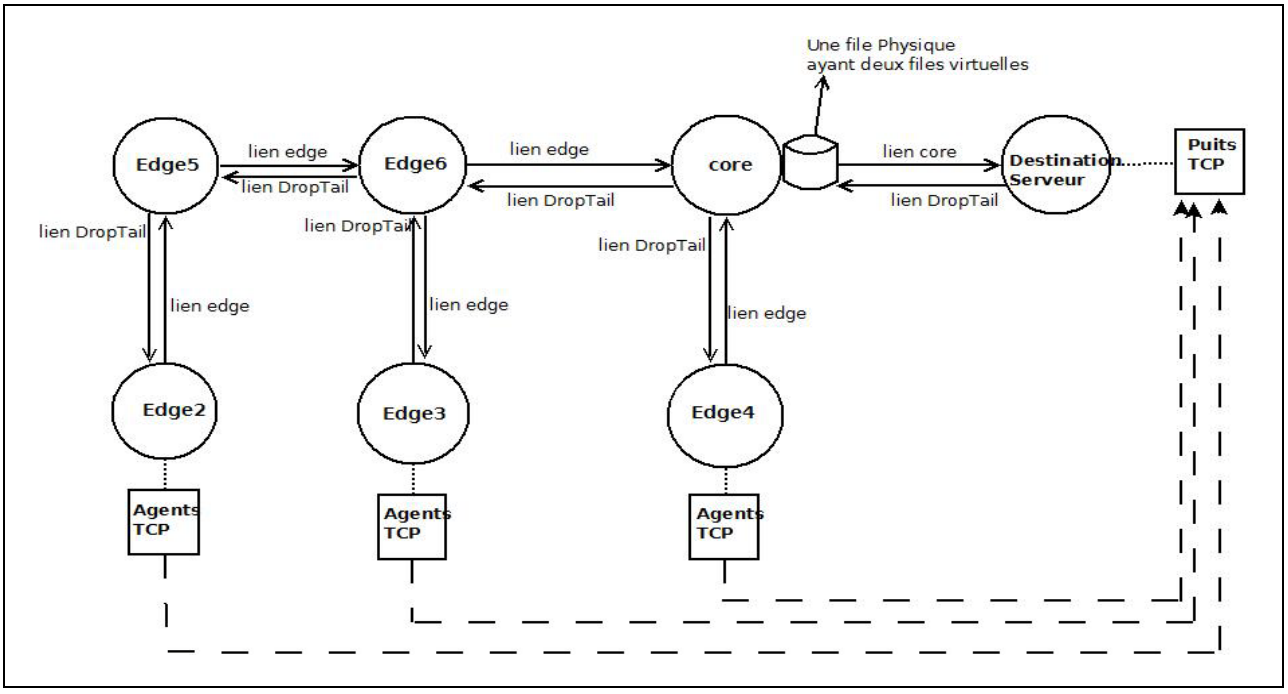

Figure 4.5 Topologie simulée avec DiffServ [1file d'attente].

Après avoir vu la faiblesse de la première approche, dans cette partie on va considérer une seule file d'attente ayant deux files virtuelles. Le *Policer* étant TSW2CM (*Time Sliding Window 2 Color Marker*) (David D. Clark, 1998), les paquets qui arriveront avec un taux inférieur au CIR (*Commited Information Rate*) seront des paquets *IN*, les autres des paquets selon le modèle RIO (David D. Clark, 1998), (N. Seddigh, 1998). RIO est une extension de RED pour le support de la QoS. Il fait partie des premiers algorithmes de gestion de file d'attente qui ont été proposés pour la différentiation de services, ainsi RIO joue un rôle important pour le contrôle de congestion d'une part et d'autre part pour assurer un traitement préférentiel aux paquets. Les paquets *IN* et *OUT* vont partager la même file. Cette file sera constituée de deux files virtuelles, la première pour les paquets *IN* et la deuxième pour les paquets *OUT*. RIO peut être considéré comme une combinaison de deux algorithmes RED de différents paramètres. Comme le montre la Figure 4.6, RIO utilise deux ensembles de

paramètres RED. Le premier ensemble {min_{thIN}, max_{thIN}, max_{pIN}} sera utilisé pour les paquets *IN* de la première file virtuelle. Le deuxième ensemble {min_{thOUT}, max_{thOUT}, max_{pOUT}} sera utilisé pour les paquets *OUT* de la deuxième file virtuelle. Le calcul de la probabilité d'effacement des paquets IN se base sur le premier ensemble de paramètres ainsi que l'occupation moyenne de la file d'attente des paquets *IN*, alors que la probabilité d'effacement des paquets *OUT* est basée sur les paramètres du deuxième ensemble ainsi que l'occupation moyenne totale incluant les paquets *IN* et *OUT*. RIO permet une différentiation de traitement entre les paquets IN et OUT. De cette manière, RIO permet de favoriser les paquets *IN* à travers la fonction de probabilité qu'il utilise. Le principe est d'effacer les paquets *OUT* en premier lieu. Afin d'assurer cette différentiation et de favoriser les paquets *IN*, les seuils des paquets *OUT* ainsi que leur probabilité maximale d'effacement doivent être inférieurs à ceux des paquets *IN*. Par exemple : $min_{thOUT} < min_{thIN}$; $max_{thOUT} < max_{thIN}$ et $max_{pOUT} < max_{pIN}$. Dans notre cas, tous les flux vont appartenir à la même classe de service. Notre but est alors de voir à quel point l'architecture DiffServ nous permet d'offrir une équité entre les flux TCP et quels sont les paramètres qui influencent cette équité. On va présenter deux cas d'étude : le premier suppose que le réseau soit surdimensionné. Dans ce cas la somme des CIR doit être inférieure à la capacité du lien goulot : $CIR_1+CIR_2+CIR_3 < C$. Par contre le deuxième va supposer que le réseau soit sous-dimensionné dans ce cas la somme des CIR doit être supérieure à la capacité de lien de goulot : $CIR_1+ CIR_2+ CIR_3 > C$. la capacité C du lien du goulot d'étranglement est égale à 10 Mbps dans notre cas.

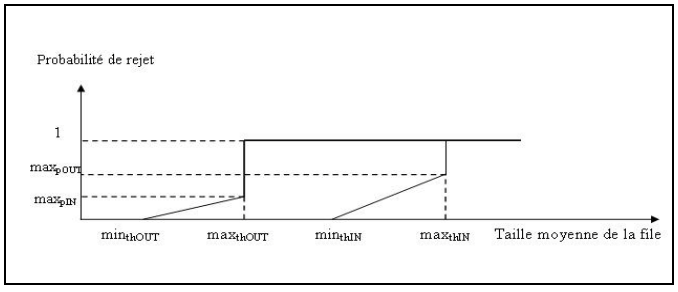

Figure 4.6 Les paramètres d'une file RIO.

1. Réseau surdimensionné

Dans cette partie, on va supposer que le réseau soit surdimensionné. Pour cela $CIR_1=CIR_2=CIR_3=$ 1 Mbps. La somme des CIR est alors 3 Mbps < C=10 Mbps. Dans ce type de réseau la bande passante totale demandée ou réservée par toutes les sources est bien inférieure à la capacité du lien de goulot et par suite il y aura de la bande passante en excès. On va commencer par étudier l'influence du RTT sur l'équité TCP dans le cadre de DiffSERV. La Figure 4.7 est relative au cas où on va modifier le délai du lien entre la source 4 (edge 4) et *core* augmentant ainsi le RTT des flux de la source 4.

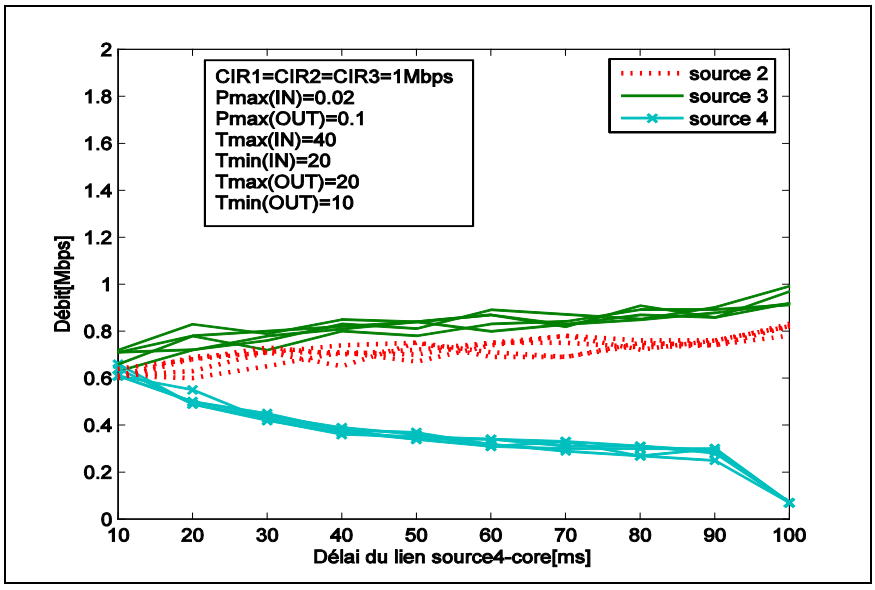

Figure 4.7 Débit en fonction du délai du lien edge4 → *core* dans le cas d'un réseau surdimensionné.

On remarque à partir de la Figure 4.7 que la source 4 s'empare de moins en moins de passante en augmentant son RTT. Toutes les sources ayant le même CIR, il semble alors que les paquets *OUT* sont à l'origine de cette inéquité. Cette source ayant le RTT le plus

faible, elle est plus agressive et génère plus de paquets OUT que celle ayant le RTT plus large et profite de la bande passante en excès.

Maintenant on va voir l'effet des paramètres RIO relatifs aux paquets *OUT*. Plus précisément on va modifier les seuils maximaux pour voir l'effet de les rendre plus sévères. Affecter des valeurs sévères aux seuils relatifs aux paquets *OUT* permet de les effacer d'avantage.

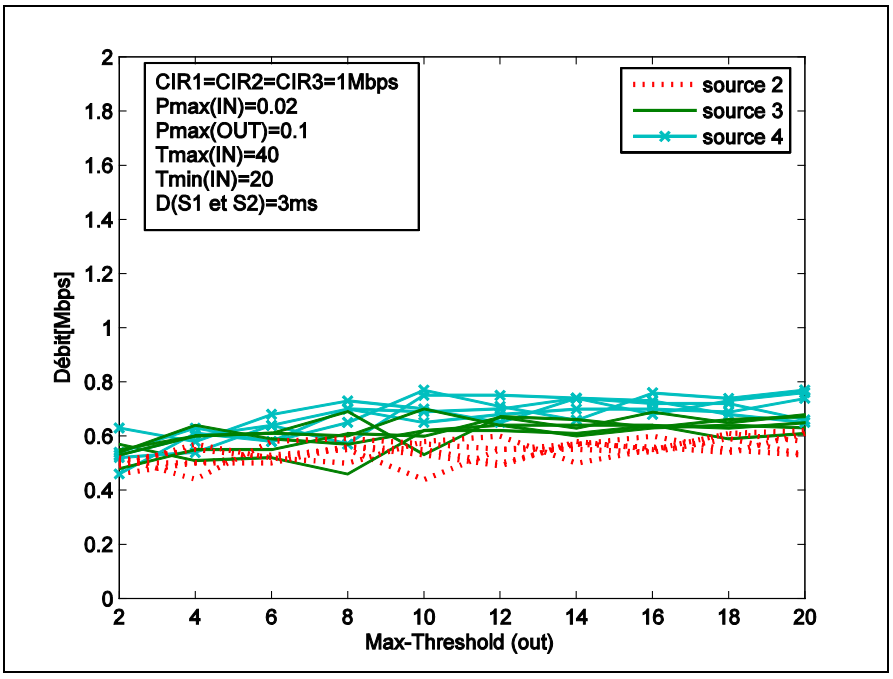

Figure 4.8 Débit en fonction max_{thOUT} (max_{pOUT} = 0.1) dans le cas d'un réseau surdimensionné.

À partir de la Figure 4.8 on observe un bon niveau d'équité même en modifiant les valeurs de min_{thOUT}, max_{thOUT}. Il est à noter que $max_{thOUT} = 2 * min_{thOUT}$. Toutefois on remarque que plus que les seuils sont sévères (min_{thOUT} et max_{thOUT} tendent vers zéro) plus l'équité est meilleure. Puisque toutes les sources ont réservé la même valeur de bande passante (elles ont le même CIR) donc le même taux de paquets IN, on s'attend à ce qu'elles reçoivent le même quota de bande passante. Ceci n'est pas le cas puisque la

source 4 ayant le plus faible RTT dispose de plus de bande passante. Le réseau étant surdimensionné, il y a alors une bande passante en excès. Tout ceci nous mène à conclure que la bande passante en excès est partagée non équitablement entre les sources. Les différentes sources ont réussi d'autre part à obtenir leur taux de bande passante désiré. Les paquets *OUT* sont les responsables de cette iniquité et ceci est justifié par le résultat illustré par la figure ci-dessus qui montre que pour des valeurs sévères des seuils relatifs aux paquets *OUT* on a une meilleure équité. Il est à noter aussi que toutes les sources arrivent à avoir leur niveau de bande passante souhaité. Traiter sévèrement les paquets *OUT* améliore l'équité mais cependant va causer une détérioration de l'utilisation de la bande passante. Ceci est montré par la Figure 4.9 ci-dessous qui illustre le débit total obtenu en fonction des seuils des paquets *OUT*.

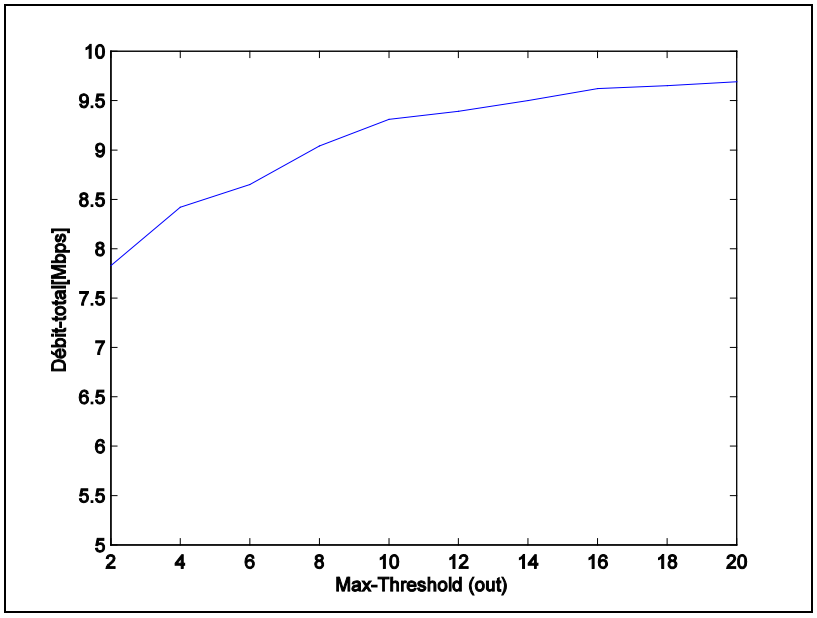

Figure 4.9 Débit total en fonction de MaxThresh dans le cas d'un réseau surdimensionné.

2. Réseau sous-dimensionné

Dans cette partie on fait la même étude mais dans le cadre dans un réseau sous-dimensionné qui est le cas le plus répandu puisque les fournisseurs de services tendent toujours à minimiser les coûts. Le réseau est dit sous-dimensionné dans notre cas si la somme des CIR est supérieure à la bande passante offerte. Dans notre cas tous les CIR sont de 3.5 Mbps ce qui fait que $CIR_1+CIR_2+CIR_3 = 10.5$ Mbps > C=10 Mbps. Comme dans le cas où le réseau est surdimensionné, nous allons voir l'effet du RTT sur l'équité et sur la performance des différents flux TCP. On rappelle que tous les délais des liens sont à 3 ms. La Figure 4.10 ci-dessous nous montre que plus le RTT de la source 4 augmente plus l'équité se détériore. D'autre part et de même que dans le cas où le réseau est surdimensionné, la différence entre les RTT joue un rôle important pour l'équité. On constate qu'en diminuant la différence entre les RTT l'équité s'améliore.

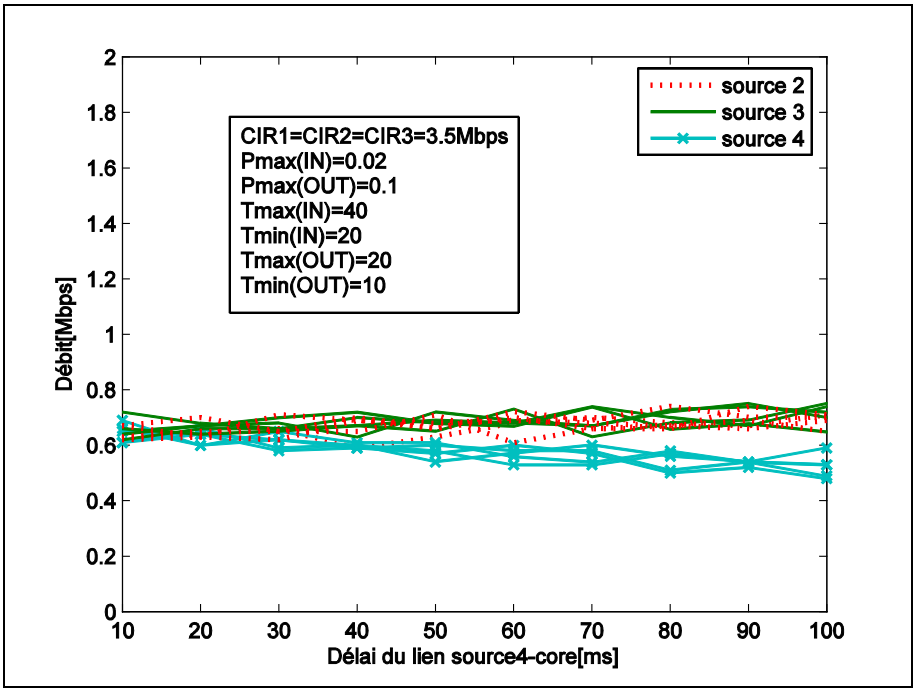

Figure 4.10 Débit en fonction du délai du lien edge4 → *core* dans le cas d'un réseau sous-dimensionné.

Comme dans le cas du réseau surdimensionné, on va traiter plus sévèrement les paquets *OUT*. Dans la Figure 4.11, on a modifié les seuils de la deuxième fille virtuelle, celle

relative aux paquets *OUT*. On observe que le niveau d'équité est bon d'une manière générale dans notre cas où tous les délais des liens sont à 3ms. Dans la Figure 4.12 et la Figure 4.13, on a encore traité les paquets OUT plus sévèrement en fixant la probabilité maximale d'effacement à 0.5 et 1 respectivement. Traiter encore plus sévèrement les paquets OUT a permis d'optimiser encore plus l'équité.

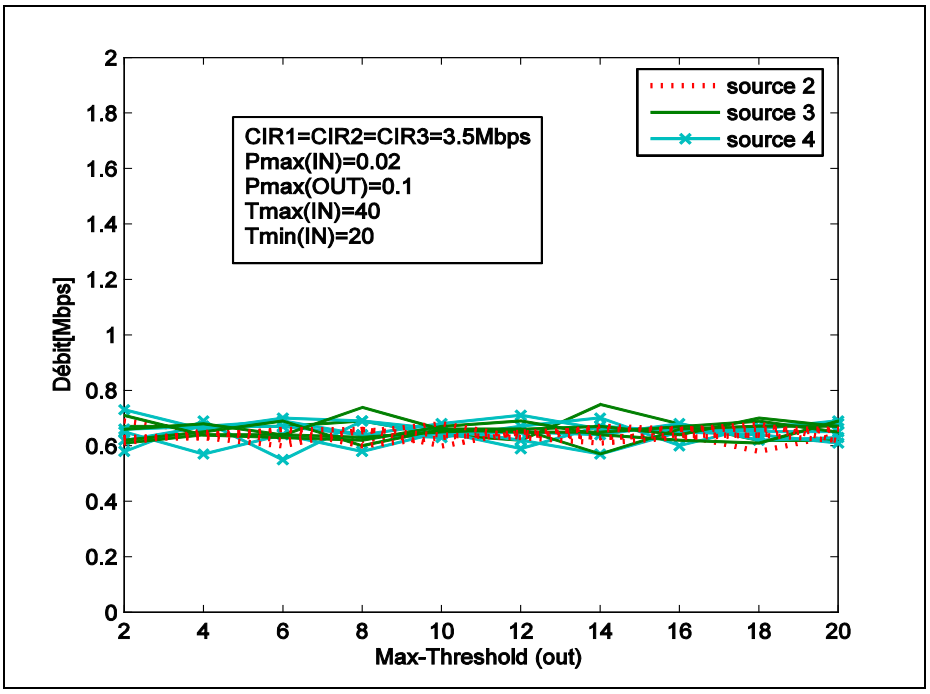

Figure 4.11 Débit en fonction max_{thOUT} (max_{pOUT} = 0.1) dans le cas d'un réseau sous dimensionné.

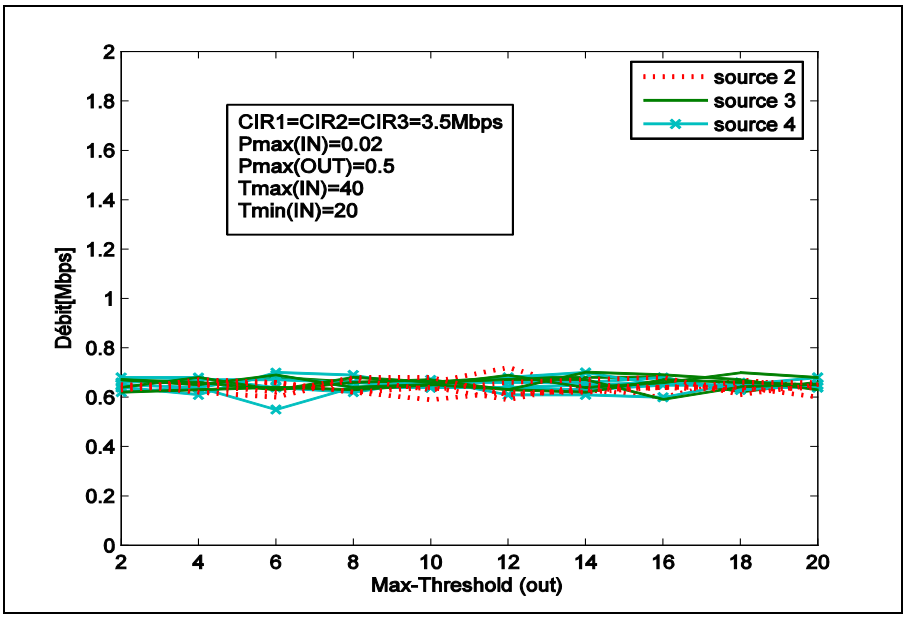

Figure 4.12 Débit en fonction max_{thOUT} (max_{pOUT} = 0.5) dans le cas d'un réseau sous dimensionné.

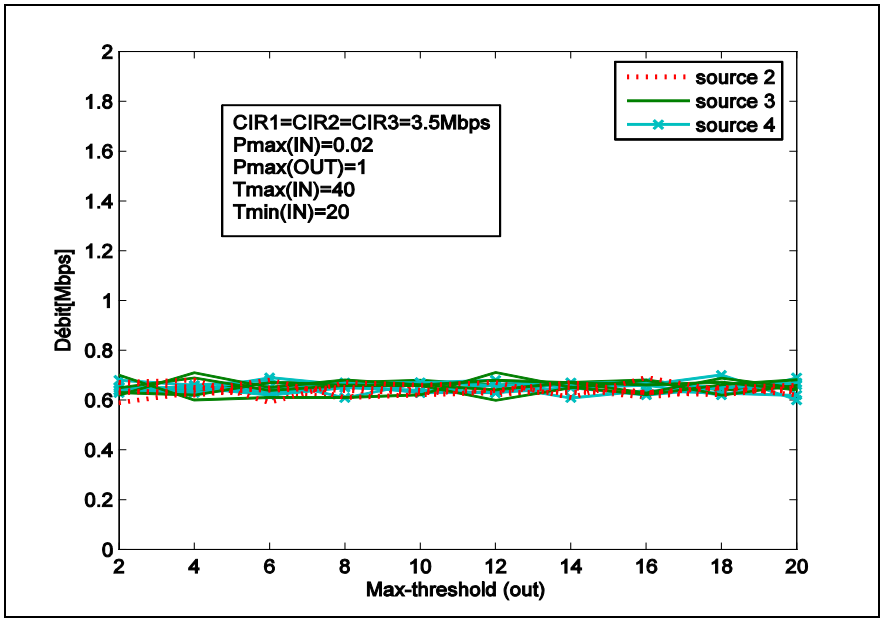

Figure 4.13 Débit en fonction max_{thOUT} (max_{pOUT} =1) dans le cas d'un réseau sous dimensionné.

Conclusion

A travers ce chapitre, on avait comme objectif de voir l'apport de DiffServ en matière d'équité à notre réseau. On a pensé à deux approches. La première consiste à consacrer une file d'attente aux flux de chaque source, c'est-à-dire attribuer un DSCP différent pour chaque source. L'objectif de cette approche est de différencier le traitement des paquets entre les différentes sources. Cette approche pose un problème d'échelle puisqu'on aura besoin d'autant de files d'attente que de nombre de sources différentes. Dans une deuxième approche, on a regroupé tous les flux dans la même classe de service. D'après les simulations on a constaté qu'on peut avoir une inéquité même avec l'utilisation de DiffServ. On a constaté aussi que les paquets *OUT* sont à l'origine de cette équité. La solution était alors de traiter ces paquets sévèrement afin d'améliorer l'équité. Mais d'autre part, on a remarqué aussi que l'équité dépend aussi de l'ampleur de la différence entre les RTT. À partir de la Figure 4.7 et la Figure 4.10, on constate que plus on augmente le RTT de la source 4, plus elle perd en bande passante. Ce qui fait que pour notre cas où on a considéré que les délais sont de 3 ms, DiffServ a réussi à fournir une bonne équité quelles que soient les valeurs de *MaxThresh*. DiffServ nous permet alors au moins de prévenir la famine en bande passante pour certaines sources tel qu'on a observé dans quelque cas dans le chapitre précédent et une bonne équité de TCP dans le cas où la différence entre les RTT n'est pas énorme. D'autre part on pourra utiliser DiffServ pour protéger TCP du protocole UDP, et pour assurer un certain niveau de QoS pour les applications utilisant UDP comme la voix sur IP, et ceci en utilisant deux classes de service. Toutefois, DiffServ est un mécanisme très fin à configurer, il y a beaucoup de paramètres à régler.

CHAPITRE 5

ÉTUDE EXPÉRIMENTALE

5.1 Introduction

Jusqu'ici, les résultats obtenus ont été relatifs aux simulations réalisées sur une topologie filaire. Comme on l'a déjà présenté dans le chapitre 2, l'environnement sans fil est différent du filaire. Les interférences, par exemple, vont induire des pertes de paquets qui vont être perçues de la part de TCP comme des congestions alors que ce n'est pas le cas, ce qui va détériorer la performance de TCP. Le caractère unidirectionnel des liens sans-fil provoque des contentions entre les paquets de données et les accusés de réception qui risquent d'être perdus. Cette perte influe aussi sur la performance de TCP. Cependant, ces problèmes peuvent être réduits avec l'utilisation des nœuds à radios multiples. L'utilisation de noeuds à deux radios permet l'utilisation par le point d'accès d'une radio pour l'accès client et l'autre pour le relais des paquets sur le lien de la dorsale. Ces deux radios vont opérer sur des fréquences différentes, et donc parallèlement, sans interférences (Bing, 2008). D'autre part, un nœud mesh peut avoir plusieurs voisins, et l'utilisation d'une seule radio pour le réseau mesh ou bien pour la dorsale fait en sorte que tous les nœuds mesh utilisent le même canal pour la connectivité et ainsi les communications parallèles sont impossibles ce qui conduit aux problèmes de contention et d'interférence réduisant ainsi la capacité du réseau. Dans un WMN multi-radio, chaque nœud est doté de plusieurs radios dédiées à la dorsale. Ainsi les liens vont opérer sur des canaux différents et indépendants (Bing, 2008), qui n'interfèrent pas entre eux (Draves, Padhye et Zill, 2004). L'utilisation des radios multiples rend flexible l'assignation de canaux afin de minimiser l'interférence. Comment assigner les canaux est l'objet de plusieurs études de recherche ayant pour objectif de trouver un algorithme d'assignation de canaux pour minimiser les interférences (Ramachandran *et al.*, 2006), (Raniwala, Gopalan et Chiueh, 2004). L'utilisation de radios multiples avec une assignation de canaux idéale permet de réduire les interférences et d'augmenter la capacité. D'autre part

elle permet de réduire la latence puisqu'elle réduit la contention. La performance des WMN à radios multiples tend alors à la similarité avec les réseaux filaires.

Le but de ce chapitre est d'étudier l'équité TCP dans un réseau sans fil multi-saut dans un environnement plus réel, à travers une étude expérimentale sur un banc de test qui est développé dans le laboratoire réseau de l'INRS. Cette étude nous permettra de voir jusqu'à quel point on pourra exploiter et valider les résultats obtenus suite aux simulations, quels autres paramètres pourraient influencer l'équité et ainsi de comparer le comportement entre le sans fil et le filaire.

5.2 Le banc de test

Le banc de test est situé à l'intérieur du laboratoire réseau de l'INRS. Il est constitué de 5 routeurs ou nœuds. Chaque nœud est une carte embarquée de type Alix3d2 avec des cartes mini-pci Wi-Fi de type Atheros AR5212 et Atheros AR5213. Chaque routeur exécute le système d'exploitation FreeBSD 8.0. La topologie expérimentale est une topologie en chaîne. Chacun de ces routeurs est fixe et muni de deux radios 802.11a/b/g, l'une pour communiquer avec son prédécesseur et l'autre pour communiquer avec son successeur. On utilise un routage IP et les routes entre la source et la destination sont statiques. Les interfaces fonctionnent sur les canaux 1, 6 et 11 qui sont orthogonaux pour minimiser les interférences. Le nœud MP4, qui représente la passerelle est connecté avec un câble Ethernet à la destination comme le montre la Figure 5.1 ci-dessous.

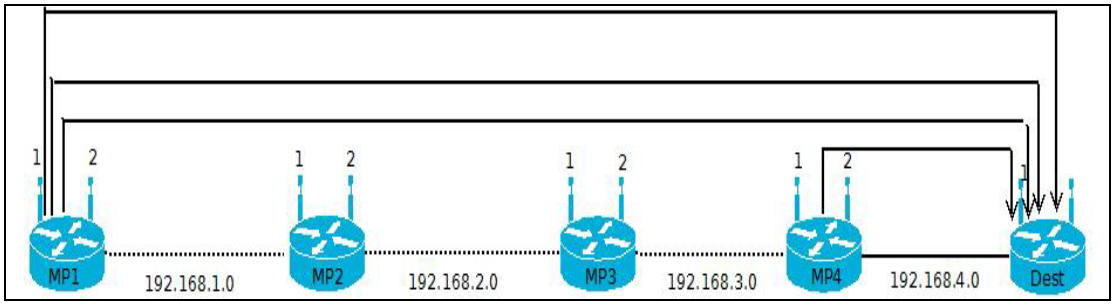

Figure 5.1 La topologie du banc de test.

Le taille du *MSS* est de 1460 octets et le taux de transmission est de 11 Mbps. nous avons utilisé les commandes *ifconfig/iwconfig* pour le contrôle des cartes sans-fil et l'outil iperf (*iPerf*) pour générer le trafic TCP ainsi que pour effectuer les mesures de performance.

5.3 Évaluation de performance

Dans ce qui suit on va présenter les différents scénarios expérimentés et les différents résultats obtenus.

5.3.1 Influence du nombre de sauts sur le débit TCP

Afin d'évaluer l'influence du nombre de sauts sur le débit TCP, on mesure le débit de la source MP1 dans chaque cas de nombre de sauts comme l'illustre la Figure 5.2. Dans chaque cas de nombre de sauts, on envoie du trafic TCP à l'aide de iperf de manière continue afin de saturer les liens. Les liens ont une bande passante de 11Mbps.

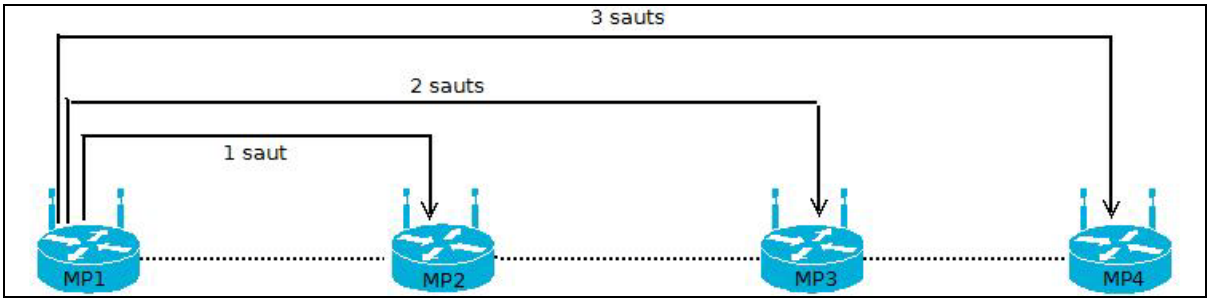

Figure 5.2 Nombre de sauts.

Le résultat de cette expérience est illustré par la Figure 5.3 ci-dessous. Cette figure nous montre que le débit de la source diminue quand le nombre de sauts augmente. En augmentant le nombre de sauts, le RTT augmente puisque le paquet va subir plus de contention et plus de

délai dans les files d'attente, et puisque le débit TCP est inversement proportionnel au RTT, il va ainsi diminuer. D'autre part en passant par plus de sauts, la probabilité de perte augmente ce qui va détériorer davantage la performance de TCP. La détérioration du débit TCP en augmentant le nombre de sauts est alors l'une des caractéristiques des réseaux sans fil multi-sauts.

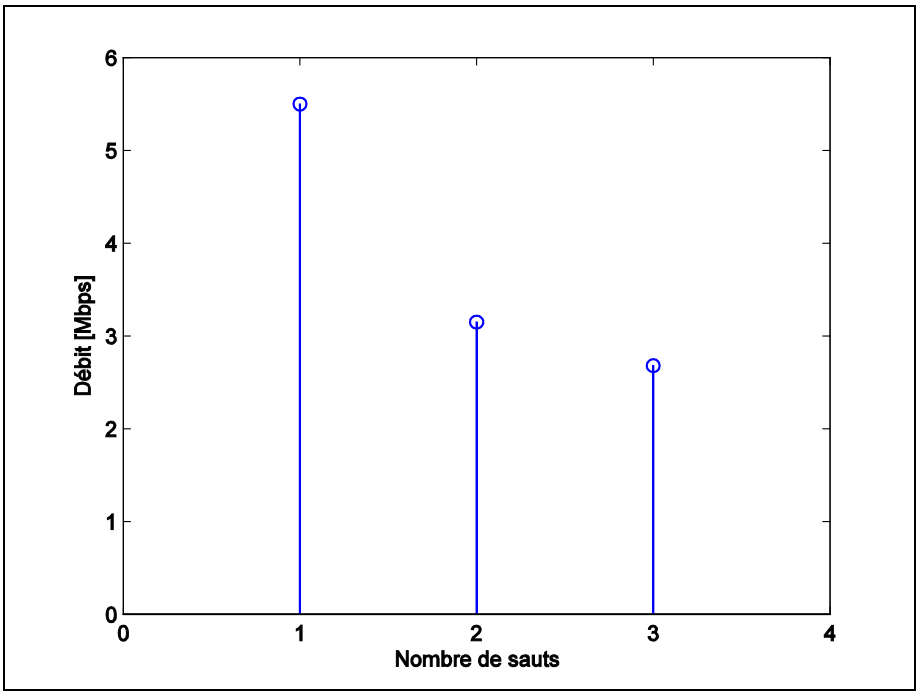

Figure 5.3 Influence du nombre de sauts sur le débit TCP.

5.3.2 Partage de débit et équité :

Pour voir le niveau d'équité entre les différentes sources, on lance l'expérience avec la topologie du banc de test illustrée par la Figure 5.1. On procède aux cas où chaque source émet 5 flux. La durée de chaque expérience est de deux minutes. L'expérience est réalisée dans le cas où la bande passante du lien Ethernet entre la passerelle et la destination est de 10 Mbps, afin de créer une saturation pour les liens sans-fil.

La Figure 5.4 nous montre que la source 4, qui la source la plus proche de la destination, s'empare de plus de bande passante que les autres sources qui sont plus en aval. Le comportement observé à travers cette figure et le comportement d'équité est similaire à celui observé avec les simulations. Ainsi la source qui est plus proche de l'autre par rapport à la destination est favorisée en termes de bande passante obtenue. Comme on l'a conclu à partir des simulations, la source la plus proche a un RTT et un taux de perte inférieurs aux autres sources et ainsi elle obtient plus de bande passante.

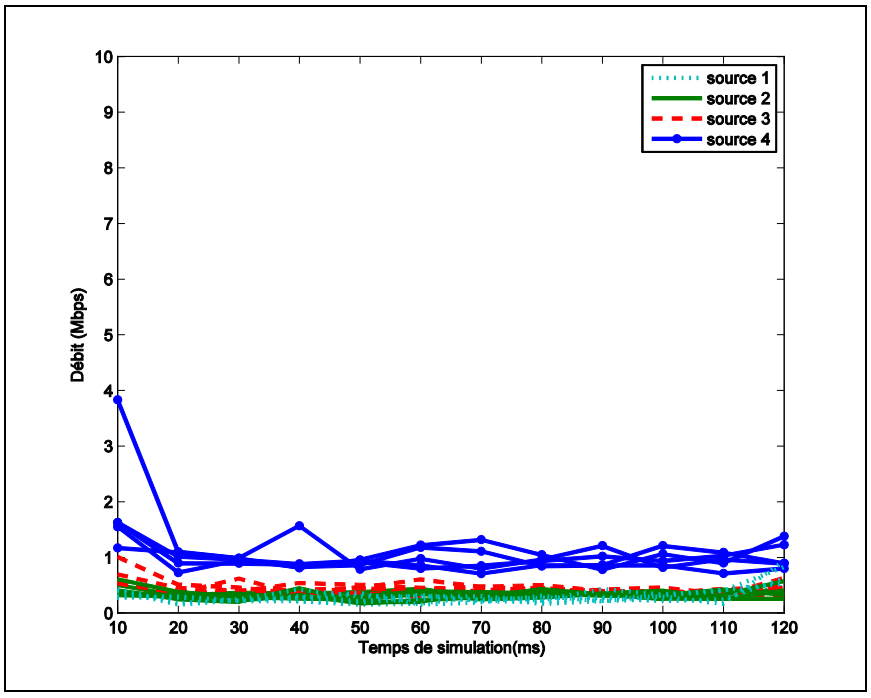

Figure 5.4 Partage de la bande passante entre les différentes sources.

5.3.3 Influence du délai externe

Afin de voir l'effet du délai externe qui est le délai du lien Ethernet entre la source 4 et la destination, on varie le délai de ce lien de 10 ms jusqu'à 100 ms avec un pas de 10 ms et on mesure le débit de chaque flux. La même expérience est réalisée avec une capacité du lien Ethernet de 10Mbps et de 100Mbps. Les résultats sont illustrés par la Figure 5.5 et la Figure

5.6 Influence du délai externe (cas 100Mbps).ci-dessous. A partir de la Figure 5.5, nous constatons que le délai externe affecte peu l'équité. Le lien étant congestionné, la source la plus proche de la passerelle se montre plus agressive et les paquets des autres sources ont plus de probabilité de rejet..Dans le cas où lien Ethernet a une capacité de 100Mbps, la Figure 5.6 nous montre que la source la plus proche perd en terme de bande passante en augmentant le délai, et le résultat est similaire à celui qui est observé dans les simulations. Ici, c'est le RTT qui pénalise la source 4 et bien sûr les autres sources sont déjà pénalisées. Améliorer l'équité passe encore par la baisse du taux d'utilisation du lien

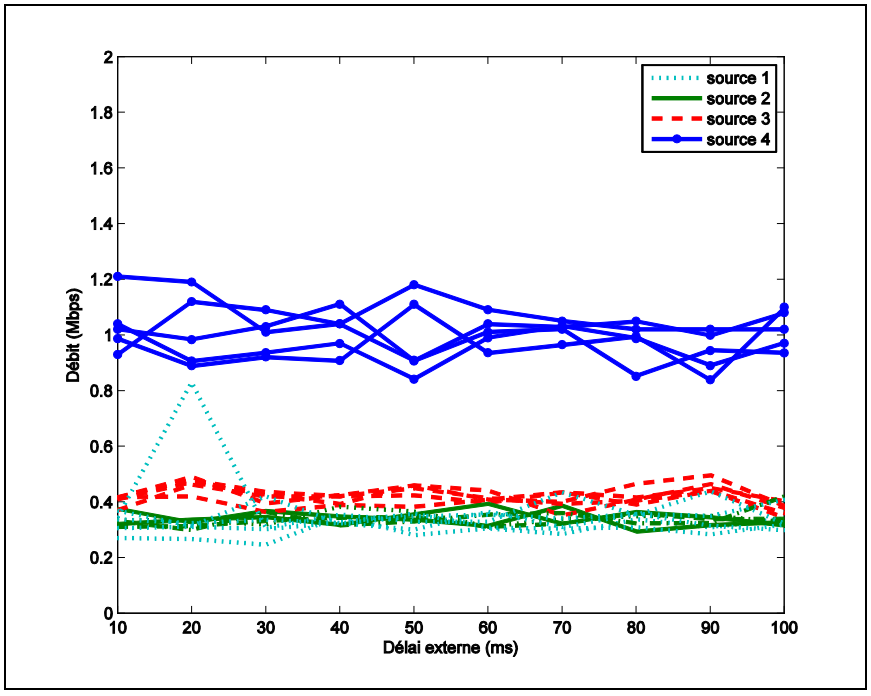

Figure 5.5 Influence du délai externe (cas 10Mbps).

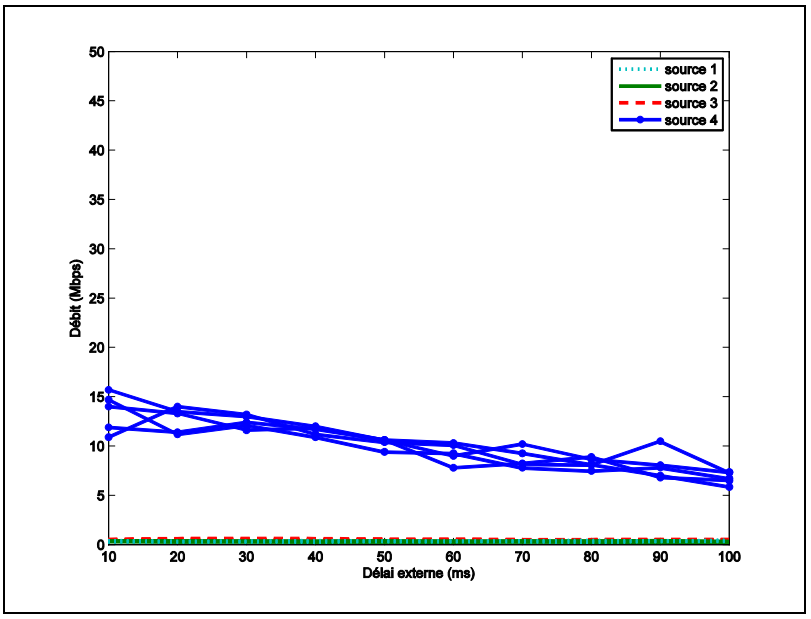

Figure 5.6 Influence du délai externe (cas 100Mbps).

5.3.4 Contrôle de la bande passante du lien du goulot d'étranglement

Étant donné que la source la plus proche de la destination est la source la plus agressive et que sa fenêtre de congestion s'ouvre plus rapidement et, par suite, qu'elle s'empare de plus de bande passante qu'une autre source plus en aval par rapport à la destination, notre idée est d'essayer de contrôler la capacité du lien de goulot et de réserver en quelque sorte la même part de la bande passante entre les flux des différentes sources. Le contrôle va se faire au niveau de l'interface d'entrée de la destination. Pour assurer ce contrôle, nous avons utilisé *ipfw*. *Ipfw* est un programme de pare-feu IP et de conditionnement de trafic. Il permet de dresser une liste de contrôle et d'accès et de traiter par suite les paquets selon cette liste. *Ipfw* est aussi l'interface utilisateur du conditionneur de trafic dummynet qui va nous permettre de manipuler le trafic des différentes sources pour qu'il soit conforme à la spécification ou les polices déjà définies. Les paquets du même flux sont transmis à un « pipe » qui permet d'émuler un lien ayant une certaine bande passante, un certain délai de propagation et un certain taux de perte. Les paquets sont ensuite transmis à un ordonnanceur WF2Q+ (Worst-

case Fair Weighted Fair Queueing). En pratique, l'objet « pipe » permet de mettre une sévère limite pour la bande passante qu'un flux peut utiliser alors que l'ordonnanceur permet de définir comment la bande passante disponible sera partagée entre les flux. Dans notre expérience, on va utiliser alors ipfw pour fixer un débit égal à tous les flux des différentes sources. Ipfw est configuré à la destination où le contrôle va se faire. On effectue les tests en commençant par réserver 1Mbps pour chaque source puis à chaque nouveau test on augmente cette réservation jusqu'à la saturation du lien Ethernet qui a une capacité de 10 Mbps, et on mesure le débit au niveau de TCP.

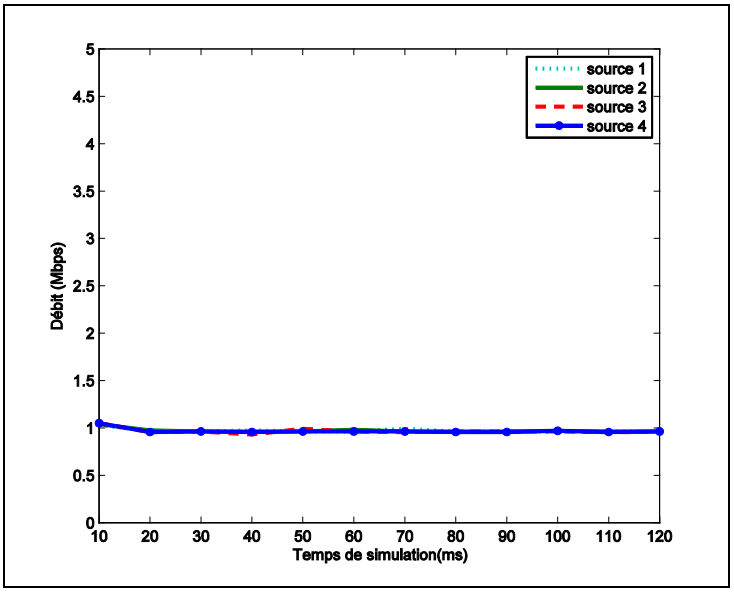

Figure 5.7 Débit TCP des différentes sources avec une réservation de 1Mbps pour chaque source.

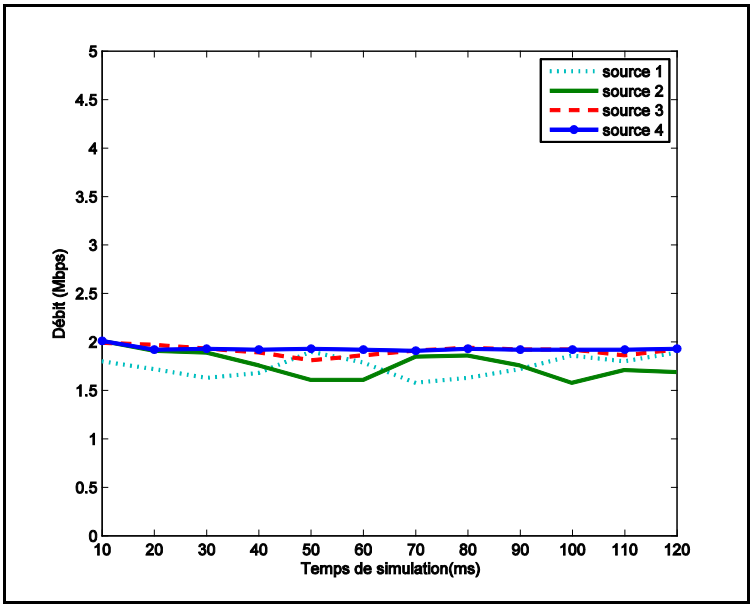

Figure 5.8 Débit TCP des différentes sources avec une réservation de 2 Mbps pour chaque source.

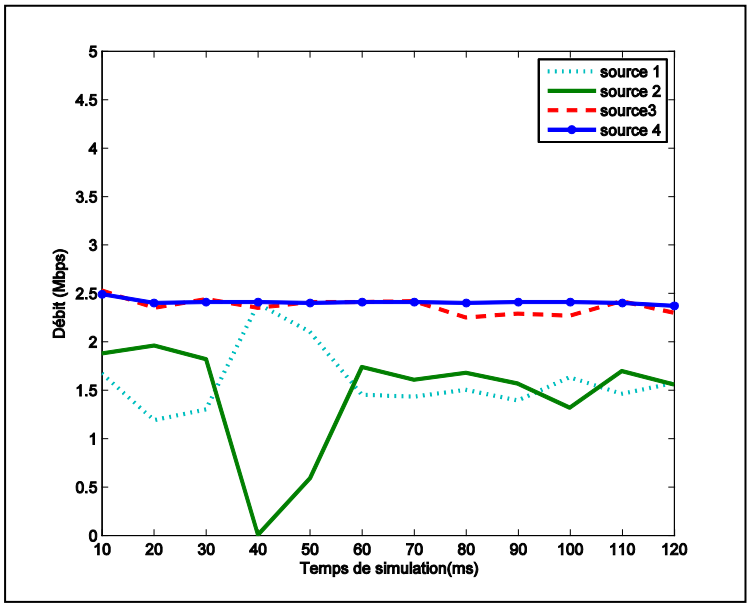

Figure 5.9 Débit TCP des différentes sources avec une réservation de 2.5 Mbps pour chaque source.

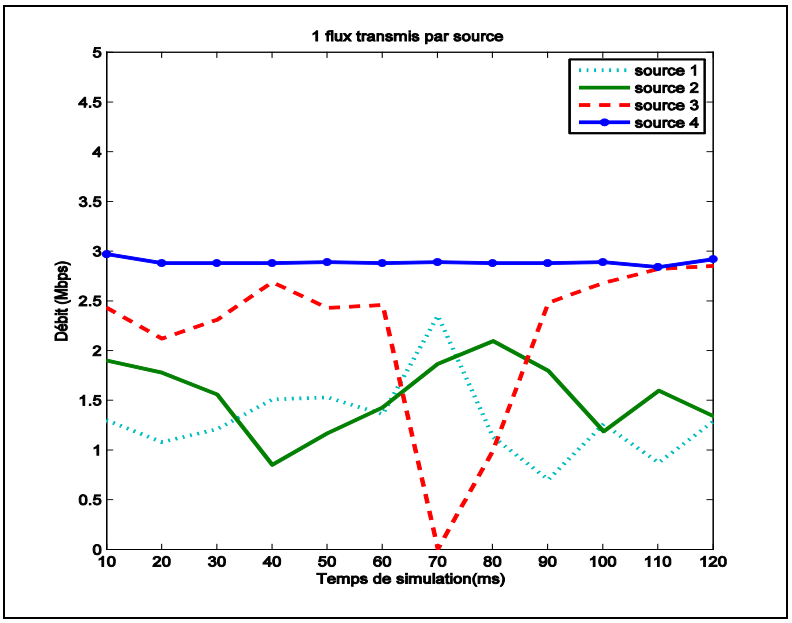

Figure 5.10 Débit TCP des différentes sources avec une réservation de 3 Mbps pour chaque source.

A partir de la Figure 5.7 on constate qu'en fixant un débit de 1 Mbps pour toutes les sources, on a obtenu une équité presque parfaite. Ce contrôle ou cet étranglement a permis que la bande passante soit partagée équitablement entre les différents flux. Dans ce cas de figure le lien Ethernet de capacité de 10 Mbps n'a pas été saturée et ainsi non seulement toutes les sources ont obtenu une part égale de la bande passante, mais le partage a pu se faire équitablement dans le cas où la bande passante est disponible.. En effet, cette équité est obtenue au prix du taux d'utilisation du lien puisque à peu prés 50% de la bande passante à été inutilisée et perdue. En augmentant le débit réservé à 2 Mbps pour chaque source, on remarque à partir de la Figure 5.8 que l'équité commence à se détériorer mais elle est toujours acceptable. Les deux sources 3 et 4 qui sont les plus proches de la destination ont presque le même niveau de débit et commencent à prendre l'avantage en termes de bande passant par rapport aux autres sources. On remarque aussi que le débit TCP obtenu est presque équivalent aux débits limites. Ce comportement reste presque le même en débit de 2.5 Mbps comme nous le montre la Figure 5.9. A partir de ce débit de 2.5 Mbps, les sources 1 et 2 commencent à perdre en termes de bande passante. Pour un débit de 3 Mbps, la Figure

5.10 nous montre que la source 4 devient plus agressive que les autres. Avec quatre sources à une bande passante réservée ou allouée de 3 Mbps, le lien Ethernet de 10 Mbps sature et le phénomène de congestion en résulte. Comme on l'a conclu à travers les simulations, quand il y a une congestion, plus une source est en aval, plus ses pertes seront supérieures aux autres. Ainsi la source 4 connait moins de pertes que les autres sources et la source 3 à son tour moins de pertes que les sources 1 et 2 et ainsi elle a un meilleur débit qu'elles comme le montre la Figure 5.10.

Donc, un contrôle et une allocation égale de la bande passante entre les différents flux nous permet une équité parfaite pourvue que la bande passante totale allouée pour toutes les sources ne dépasse pas à peu près 90 % de la capacité du lien goulot.

Conclusion :

Le banc de test nous a permis d'observer le comportement et l'équité TCP pour un réseau multi-saut sans fil dans un environnement réel. On constate que le comportement est similaire à celui observé dans nos simulations, entre autres le rejet de paquets d'une manière non équiprobable au niveau de la passerelle, en cas de congestion, est l'une des principales causes de l'iniquité. En effet dans un environnement sans-fil, l'effet du délai externe n'était pas aussi consistant que celui qu'on a constaté lors des simulations filaires. D'autre part, nous avons réussi à obtenir une équité parfaite en utilisant *ipfw* pour réserver une bande passante égale au niveau de la destination pour tous les flux pourvu que la bande passante totale allouée ne dépasse pas 90 % de la capacité du lien du goulot.

Ces résultats valident les observations obtenues lors des simulations. Entre autres nous avons réussis à appliquer des mécanismes analogues à ceux de DiffServ en utilisant *ipfw* et améliorer l'équité TCP d'une manière considérable, mais au détriment de l'utilisation optimale de la bande passante.

CONCLUSION

Dans ce mémoire, on a procédé à une étude et une analyse de l'équité TCP dans les réseaux sans fil multi-saut en se concentrant sur les réseaux maillés sans fil. Nous avons procédé à la détermination des différents facteurs qui influent sur l'équité dans le cas de l'utilisation du mécanisme de gestion de file d'attente *DropTail* et RED au niveau de la file d'attente de la passerelle. D'une manière générale, les sources les plus en aval de la passerelle souffrent pour avoir de la bande passante et la source qui est plus proche risque de créer une famine en termes de bande passante pour elles. En utilisant *DropTail*, améliorer l'équité passe à travers des files d'attente de grande taille ce qui est une solution non intéressante. RED qui peut être une solution pour l'équité quand il est déployé dans le cœur de réseaux, permet de résoudre quelques problèmes de *DropTail.* Toutefois il ne constitue pas une solution idéale pour l'équité dans notre cas puisque il efface aléatoirement les paquets et ne fournit pas un traitement différencié des paquets. D'autre part on a constaté que le rapport, délai externe/délai interne, joue un rôle important pour l'équité. En augmentant ce rapport, on améliore l'équité puisque cette augmentation permet de réduire l'agressivité de la source la plus proche de la passerelle.

Dans une deuxième partie nous avons proposé l'utilisation des mécanismes de DiffServ. Une première approche était de considérer une classe pour les flux de chaque source, toutefois cette approche n'est pas intéressante de point de vue du passage à l'échelle puisqu'elle requiert autant de files d'attente que des différents nombres de sauts dans le réseau. La deuxième approche consistait à considérer une seule classe de trafic et à traiter différemment les paquets qui sont dans le profil et les paquets qui sont hors profil. Cette solution nous permet d'éviter le problème de famine pour les sources les plus en aval et d'avoir un bon résultat en termes d'équité en considérant les délais qui sont propres aux réseaux maillés sans fil.

Enfin, les expériences réalisées sur le banc de test nous ont permis d'analyser le problème d'équité TCP dans les réseaux sans fil multi-saut dans un environnement réel. On a constaté le même comportement que celui obtenu au cours des simulations. D'autre part, pour valider les résultats obtenus en utilisant les principes de DiffServ, nous avons utilisé le programme ipfw pour exercer un contrôle de partage de bande passante au niveau de la destination. Nous avons ainsi réussi à atteindre une équité parfaite et le banc de test nous a permis alors de valider les résultats de simulation.

D'autre part, les résultats de l'analyse obtenus à l'issue de ce mémoire soulèvent les perspectives suivantes :

Le débit TCP étant très sensible au RTT et au taux de perte, l'équité ne peut se réaliser naturellement dans un réseau d'accès où le nombre de sauts, et donc les délais, est variable. Une gestion adaptative de files d'attente en fonction de la position de la source par rapport à la passerelle pourrait être une solution à ce problème pour apporter un traitement différencié des paquets des différentes sources basé sur l'adaptation de la probabilité de rejet d'un paquet au nombre de sauts effectués par ce paquet. Le champ TTL (Time To Live) de l'entête IP pourrait être exploité pour extraire cette information.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Akyildiz, Ian F., Xudong Wang et Weilin Wang. 2005. « Wireless Mesh Networks: a Survey ». *Computer Networks and ISDN Systems*. p. 445-487.

Allman, M., V. Paxson et W. Stevens. 1999. *TCP Congestion Control: RFC2581*. Internet Engineering Task Force.

Anna Zakrzewska, Leszek Koszałka, Iwona Poźniak-Koszałka. 2008. « Performance Study of Routing Protocols for Wireless Mesh Networks ». In *19th International Conference on Systems Engineering*.

BelAir Networks. <http://www.belairnetworks.com/products/>.

Bing, Benny. 2008. *Emerging Technologies in Wireless LANs Theory, Design and Deployement* 2008. Cambridge.

Blake, Steven, David Black, Mark Carlson, Elwyn Davies, Zheng Wang et Walter Weiss. 1998. *An architecture for differentiated services: RFC 2475*. Internet Engineering Task Force.

David D. Clark, Wenjia Fang. 1998. « Explicit allocation of best-effort packet delivery service ». *IEEE/ACM TRANSACTIONS ON NETWORKING*. p. 362–372.

Draves, Richard, Jitendra Padhye et Brian Zill. 2004. « Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks ». In *ACM Mobicom*.

Ekram Hossain, Kin K. Leung 2008. *Wireless Mesh Networks Architectures and Protocols*. Springer.

Firetide. <http://www.firetide.com/innerContent.aspx?taxid=8&id=2578&linkidentifier=id&itemid=2578>.

Floyd, S., et T. Henderson. 1999. *The NewReno Modification to TCP's Fast Recovery Algorithm: RFC 2582*. Internet Engineering Task Force.

Floyd, Sally, et Van Jacobson. 1993. « Random Early Detection Gateways for Congestion Avoidance ». *IEEE/ACM Transactions on Networking*. p. 397-413.

Hari Balakrishnan, Venkata N. Padmanabhan, Randy H. Katz. 1997. « The Effects of Asymmetry on TCP Performance ». In *3rd ACM/IEEE Intl. Conference on Mobile Computing and Networking* (Budapest, Hungary, Sept. 1997).

iPerf. <http://dast.nalanr.net/Projects/Iperf>.

James F. Kurose, Keith W. Ross. 2003 *Computer Networking: A Top-Down Approach Featuring the Internet* Addison Wesley.

Jiwei Cen, Yeng-Zhong Lee, Daniela Maniezzo, Mario Gerla. 2006. « Performance Comparison of AODV and OFLSR in Wireless Mesh Networks ». In *IFIP Fifth Annual Mediterranean Ad Hoc Networking Workshop*.

Liu, Chunlei, Fangyang Shen et Min-Te Sun. 2007. « A Unified TCP Enhancement for Wireless Mesh Networks ». In *Parallel Processing Workshops, 2007. ICPPW 2007. International Conference on* (10-14 Sept. 2007). p. 71.

LUCEOR. <http://www.luceor.com/rubrique.php3?id_rubrique=89>.

Mathis, M., J. Semke, J. Mahdavi et T. Ott. 1997. « The macroscopic behavior of the TCP congestion avoidance algorithm ». *ACM SIGCOMM* p. 67-82.

Microsoft. <http://research.microsoft.com/en-us/projects/mesh/>.

Moraru, B., F. Copaciu, G. Lazar et V. Dobrota. 2003. « Practical Analysis of TCP Implementations: Tahoe, Reno, NewReno ». In *Proceedings of RoEduNet International Conference: Networking in Education and Research*.

N. Seddigh, B. Nandy, P. Pieda, J. Hadi Salim, A. Chapman. 1998. « An Experimental Study of Assured Services in a DiffServ IP QoS Network ». *Proceedings of SPIE Symposium on Voice, Video and Data Communications*. p. 217-219.

Network Simulator. <http://www.isi.edu/nsnam/ns/>.

NORTEL.
<http://www2.nortel.com/go/solution_content.jsp?parId=0&segId=0&catId=W&prod_id=47160>.

Postel, Jon. 1981. *Transmission Control Protocol : RFC 793*. Internet Engineering Task Force.

Ramachandran, Krishna N., Elizabeth M. Belding, Kevin C. Almeroth et Milind M. Buddhikot. 2006. « Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks ». In *INFOCOM*.

Raniwala, A., K. Gopalan et T. Chiueh. 2004. « Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks ». *Mobile Computing and Communications Review*. p. 50-65.

S.G, Colombo., Mandorino. T et Paltenghi. G. 2008. « TCP optimization in wireless mesh backhaul networks ». In *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium* (7-9 Mai 2008). p. 42-45. In *INSPEC*.

Strix SYSTEMS. <http://www.strixsystems.com/products/default.asp>.

TROPOS Networks. <http://www.tropos.com/products/routers.html>.

Tung, Li-Ping, Wei-Kuan Shih, Te-Chung Cho, Y.S. Sun et Meng Chang Chen. 2007. « TCP Throughput Enhancement over Wireless Mesh Networks ». *Communications Magazine, IEEE,* vol. 45, p. 64-70.