

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane MIRA de BEJAÏA

Faculté des Sciences Exactes

Département d'Informatique



Mémoire de fin d'études en vue d'obtention du diplôme de Master

En Informatique

Option : Administration et Sécurité des Réseaux

### Thème

# Implémentation d'un Proxy MP-TCP au niveau Hyperviseur

Réalisé par : Mr. GUERMOUZ Amine

Devant le jury composé de :

President: M<sup>r</sup>. SAADI Mustapha. MAA UAM Bejaia.

Examineur : M<sup>r</sup>. KHENOUS Hahemi . MAA UAM Bejaia.

Promoteur : M<sup>r</sup>. BOUKERRAM Abdallah. Pr UAM Bejaia

Année universitaire 2013-2014



## Remerciements

*"L'éducation n'est, en somme, que l'art de révéler à l'être humain le sens intime qui doit gouverner ses actes, préparer l'emploi de ses énergies et lui communiquer le goût et la force de vivre pleinement."*

*Henry Bordeaux (1870-1963)*

*Ma profonde gratitude et mon éternelle reconnaissance*

*À mes parents,*

*Pour leur amour et leur abnégation qu'ils ont dévotieusement consentis pour que nous puissions un jour être dignes de leurs immenses sacrifices et de leurs espoirs.*

*À mes Professeurs, Spécialement à Mr BOUKERRAM Abdallah*

*Pour leur patience et leur disponibilité permanente et sans faille qui nous permet d'emprunter cette voie si noble et interminable, privilège sublime qu'est la quête du savoir. Spécialement à Mr BOUKERRAM Abdallah qui a accepté de m'encadrer, ainsi qu'aux membres de jury qui ont acceptés d'examiner mon travail.*

*À mes amis*

*Instigateurs malgré eux de cette émulation, seule force interne et mystérieuse qui anime en nous ce désir et cette ambition de parvenir à des sommets qui feront la fierté de nos parents, de notre université, et de notre pays.*



## Dédicaces

*C'est avec un cœur ouvert et une immense joie que je dédie ce modeste travail à :*

*Mes très chers, respectueux et magnifiques parents que j'aime le plus au monde, mon cher père qui a toujours su me raisonner et m'orienter dans mes choix, ma maman chérie qui a toujours été là pour moi, qui a su me redonner raison quand je l'ai perdu, me redonner espoir quand j'en n'en avais plus, et qui a toujours su dessiner un sourire sur mon visage même quand tout aller vraiment mal, je leurs dédie ce travail car j'en suis sûre qu'ils le méritent même un peu plus que moi.*

*A mon frère Wassime et ma sœur Lydia.*

*A mes amis et à mes collègues (laifa, Fares, Mohamed, Merhrez, hamou, Remdane, Abderhman EL harrachi, Sabiha, Salima, Amel et Ferroudja...)*

*A tous ceux qui me sont chers, ...*

# Table des matières

<b>Tables des matière</b> .....	i
<b>Listes des figures</b> .....	ii
<b>Listes des tableaux</b> .....	iii
<b>Introduction Générale</b> .....	1
<b>Chapitre I - Généralité sur linux</b> .....	2
Introduction .....	2
I. Définition .....	2
II. Bref historique de linux .....	2
III. Les caractéristiques de linux.....	3
IV. Applications de linux.....	3
V. Architecture de linux .....	4
V.1. Les composants du système de linux .....	4
V.2. Les avantages de l'architecture de linux .....	6
VI. La notion du noyau et de distribution.....	6
VI.1 La notion du noyau .....	6
VI.2 Les distributions.....	7
VII. Les partitions sous linux .....	9
VII.1 Définition.....	9
VII.2 Comment partitionner.....	9
VII.3 Les partitions de linux .....	10
Conclusion .....	11
<b>Chapitre II - Les serveurs proxy</b> .....	12
Introduction : .....	12
I. Définition d'un proxy :.....	12
II. Le fonctionnement d'un proxy :.....	13
III. Les principaux objectifs des serveurs proxy : .....	13
IV. Le rôle du proxy: .....	13

V. Les types de proxy :	14
VI.Les dangers de proxy :	14
Conclusion :	14
<b>Chapitre III - Etude et Conception d'un Proxy multipath – tcp au niveau Hyperviseur</b>	<b>15</b>
Introduction :	15
I. Description :	15
I.1. Objectifs :	15
I.2. Environnement :	16
II.Contexte technologique :	16
II.1. Etat de l'art :	17
II.2. Descisions de conception :	19
III.Analyse :	19
III.1. Fonctionnement du Protocole TCP :	19
III.1.1 La vie d'une connexion TCP :	20
III.2. Fonctionnement de MP-TCP :	21
III.2.1 Eléments important utilisé par MP-TCP :	22
III.2.2. Initialisation :	23
III.2.3. Echange de données :	23
III.2.4. Les problèmes potentiels de MP-TCP :	24
IV.Hyperviseur KVM :	25
IV.1. Architecture KVM :	27
IV.2. But et prérequis :	29
Conclusion:	29
<b>Chapitre IV - Installation et implémentation d'un Proxy multipath – tcp au niveau Hyperviseur</b>	<b>30</b>
Introduction :	30
I.Plan de développement :	31
I.1 Installation et manipulation de KVM :	31
I.2 Identification de codes à modifier dans KVM :	31
I.3 Implémentation du Proxy MP-TCP au niveau de l'hyperviseur KVM :	31
I.4 Test et comparaison des résultats :	32
II. Installation de KVM :	32
III.Les étapes d'installation de MPTCP	37

IV.Implémentation de la middle-box MP-TCP au niveau de KVM .....	42
V.Configuration Proxy MP-TCP :.....	43
VI.Difficultés à surmonter lors de la conception.....	44
VII.Etat d'avancement :.....	45
Conclusion :.....	45
<b>Conclusion Générale</b> .....	<b>46</b>
<b>Références bibliographiques électroniques</b> .....	<b>48</b>
<b>Annexes</b> .....	<b>51</b>

## Listes des figures

Figure 1 : Les composants de Linux. ....	6
Figure 2 : la position de Proxy. ....	12
Figure 3 : modèle de communication (hyperviseur – serveur).....	15
Figure 4 : connexion mobile – internet. ....	16
Figure 5 : chemins redondants vers les mêmes serveurs.....	17
Figure 6 : poignée de main TCP.....	20
Figure 7 : fermeture de connexion TCP.....	20
Figure 8 : modèle avec couches (MP-TCP). ....	21
Figure 9 : poignée de main MP-TCP. ....	23
Figure 10 : Architecture global. ....	26
Figure 11 : Architecture KVM. ....	27
Figure 12 : Imprimé écran – groupe KVM. ....	34
Figure 13 : Imprimé écran – ajout de l'utilisateur au groupe KVM. ....	34
Figure 14 : Imprimé écran – création d'une nouvelle machine.....	35
Figure 15 : Imprimé écran – machine Xp créé.....	36
Figure 16 : Imprimé écran – ajout de la 2e machine (Ubuntu). ....	36

## Liste des tableaux

Tableau 1 : Les répertoires de Linux et leurs contenus.....	10
---	----



Les réseaux sont devenus multiples, les appareils mobiles ont des interfaces multiples, les "Data center" ont des chemins redondants.

Le Cloud ainsi que le multi-homing sont devenus les normes pour les grandes fermes de serveurs, tandis que TCP n'utilise encore qu'un seul chemin. On essaye alors d'implémenter des solutions pour gérer le trafic et les connexions dans un environnement tel que le Cloud.

Est-il possible aujourd'hui, d'étendre le protocole TCP pour lui permettre de recourir à plusieurs chemins d'accès pour les applications courantes sur Internet ? La réponse est positive, si nous examinons soigneusement la conception de Multipath TCP, dans le noyau Linux et si nous évaluons sa performance.

Nos mesures portent sur les algorithmes nécessaires pour utiliser plusieurs chemins ayant des caractéristiques différentes, au niveau d'un Hyperviseur. Ces chemins sont utiles pour envoyer et recevoir des données et réorganiser les segments tout en gardant une transparence au niveau des machines virtuelles.

Dans le premier chapitre, des généralités sur linux sont présentées. Le second chapitre est consacré aux serveurs proxy, ses principes et ses dangers. la conception d'un proxy multipath-tcp au niveau Hyperviseur est proposée en troisième chapitre, ainsi que le fonctionnement de MP-TCP et les éléments qu'il utilise. L'architecture des KVM est également présentée, le quatrième chapitre est consacré pour les implémentations du projet. Une conclusion générale reprenant les points forts de ce travail suivie de perspectives futures vient clôturer ce mémoire.

## Introduction

Ce chapitre présente les principaux points théoriques du fonctionnement d'un système d'exploitation appelé linux. Il est probablement la réalisation la plus importante parmi les gratuits depuis 'Space-war' ou plus récemment 'Emacs'. Il fait partie de la plus grande famille du système d'exploitation Unix, il est son clone. Destiné aux ordinateurs de processeur Intel 80386 et 80486, il supporte une large gamme de programmation.

Multi usager d'Unix : il peut transformer n'importe quel PC 386 ou 486 en station de travail, car il apporte toute la puissance d'Unix. Des sociétés commerciales installent des réseaux entiers de machines tournant sous linux destinées à gérer les données financières ou médicales des télécommunications et bien d'autres choses. Dans le monde entier, des universités utilisent LINUX, pour l'étude des systèmes d'exploitation, des partitions d'informatique qu'ils utilisent chez eux pour programmer, produire, etc...

Sa particularité est qu'il est quasi gratuit (open source), développé par un groupe de volontaires de départ sur Internet, ils s'échangent du code, notifiant les bogues et corrigeant les problèmes selon un modèle de développement ouvert à tous , donc tout le monde est invité à participer : il suffit juste de posséder de bonne base de programmation.

## I. Définition

Un système d'exploitation est un ensemble de programmes et de sous programmes qui ont pour rôle de gérer et de piloter le matériel. C'est l'intermédiaire entre les programmes d'application et le matériel. Linux met ses fonctionnalités à la disposition des programmes d'application sous forme d'appel système (en langage C).

## II. Bref historique de linux

Comme on l'a déjà mentionné Linux est une version d'Unix gratuite et libre, diffusable développée à l'origine par Linus torvalds.

Il a été au départ un projet de loisir pour cet étudiant finlandais, et fut inspiré de minix écrit dans un but pédagogique par Andrew Tanenbum.

# Chapitre I. Généralités sur linux

---

En 1992, la toute première version fait son apparition avec l'intégration des développements libre des outils présents sous Unix, de nouveaux outils pour Linux apparaissent avec une vitesse vertigineuse.

La coopération de beaucoup de personne avec torvalds, via internet lui a permis d'arriver à enrichir le noyau, car ces développeurs ont implémenté de nombreuses fonctionnalités qui font de linux aujourd'hui un système Unix complet et efficace.

## III. Les caractéristiques de linux

- Linux est un système robuste, puissant, souple et efficace car il utilise peu de ressources.
- Il est multitâche et multi utilisateurs (comme toute autre version d'Unix). Cela signifie que plusieurs utilisateurs peuvent travailler sur la même machine en même temps et lancer des processus pour s'exécuter en parallèle.
- Il est multi plateformes ; c'est-à-dire qu'il fonctionne sur différents processeurs.
- Il supporte en standard le protocole TCP/IP.
- Son originalité est qu'il est libre, chacun peut y apporter ses propres modifications puisque son code source est fourni avec des fichiers binaires.
- Aucune société n'est en charge du développement de linux : ceci est considéré comme le plus grand inconvénient tel qu'en cas d'une panne ou d'un problème, il n'existe pas de service après-vente.
- C'est un système à part entière, qui est difficile à faire cohabiter avec un autre système pour un novice.

## IV. Applications de linux

Si l'utilisation d'un système d'exploitation est un amour en soi, ce n'est pas la raison pour laquelle de nombreuses personnes utilisent un ordinateur. La plupart d'entre nous utilisent un ordinateur pour son travail.

Linux offre des milliers d'applications disponibles aujourd'hui. Ces applications comprennent des programmes pour tableurs, base de données, traitement de texte, création d'application avec une variété de langages informatique et des logiciels de télécommunication pour nous connecter sur les services on-line.

Linux est également fourni avec une vaste gamme de jeux, textuelle et graphique.

## V. Architecture de linux

### V.1. Les composants du système linux

Linux se compose de la couche logicielle suivante [1,2] :

#### - Le noyau (kernel)

Il prend en charge la gestion du matériel ainsi que la gestion de certaines fonctionnalités de haut niveau, c'est pour cela qu'on pourrait craindre une faible fiabilité du Système linux.

Il n'est constitué que d'une base relativement petite et très bien testée, à laquelle beaucoup de pilotes et de périphériques ont été ajoutés.

#### - Les programmes en ligne de commande et le Shell

Shell qui signifie environnement utilisateur est capable de lire des commandes saisies au clavier, de les exécuter et d'afficher leurs résultats à l'écran.

Mais le Shell est bien plus que ça, car il peut être programmé et gérer les processus (en arrêter, en lancer un autre, etc.).

Il existe plusieurs Shell, et le plus utilisé sous linux est le Shell bash qui est développé par la free software. Les programmes que l'on peut directement injectés dans le Shell sont ceux que l'on appelle les programmes « ligne de commandes ». Ces programmes sont encore une fois, développés soit par free software foundation, soit par des bénévoles. Toutes ces commandes sont compatibles à Unix.

## - Serveur X window

Bien entendu, les commandes sont essentielles pour pouvoir utiliser le système linux, mais elles sont assez rébarbatives et peu d'utilisateurs acceptent de s'en contenter. C'est pour cela qu'une couche graphique a été développée pour introduire une interface graphique plus conviviale : XWINDOW. Cette couche logicielle est constituée de plusieurs composants, dont la base est le serveur X.

## - Le gestionnaire de fenêtre et le gestionnaire du bureau

- **Le gestionnaire de fenêtre (window manager) :**

Il prend en charge la gestion des fenêtres principales, des applications graphiques sous X et des décorations de fenêtre du premier niveau (fenêtres principales, des programmes).

Par exemple affichage des bords, la barre de titre, les boutons de réduction ... etc.

- **Le gestionnaire du bureau :**

Il existe également des environnements graphiques complets qui, en plus de fournir un gestionnaire de fenêtre, fournissent la plupart des outils classiques qui comprennent des éditeurs des outils de configuration, des navigateurs internet comme ils fournissent aussi un cadre standard pour les applications graphiques qui permet d'améliorer l'intégration des diverses applications entre elles, c'est la raison pour laquelle on appelle ces environnements des gestionnaires du bureau.

## - Les applications Xwindow

Cette couche se situe tout au-dessus de toutes les autres déjà citées. Ces applications sont diverses (traitement de texte, logiciels de dessin...), quelques-uns sont des «font-end» application en ligne de commande c'est-à-dire des interfaces graphiques pour des programmes non graphiques.

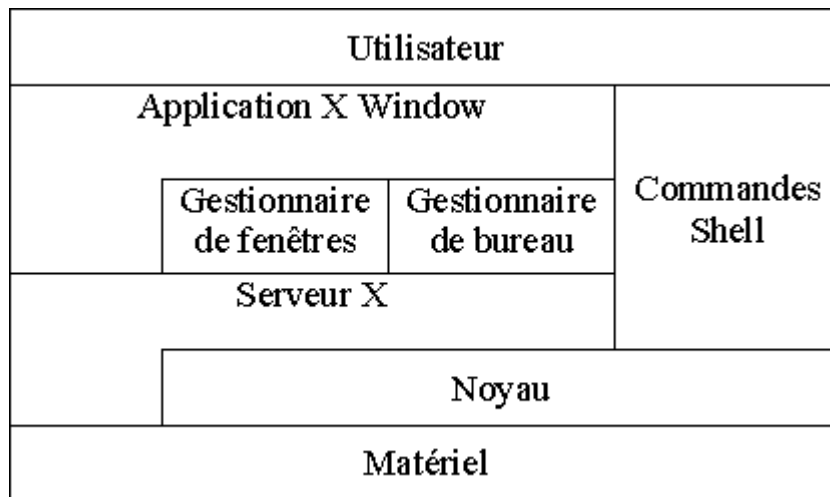


Figure1 : Les composants de Linux [3]

## V.2. Les avantages de l'architecture de linux

- Linux est très structuré, donc simple à utiliser, à configurer, à maintenir et à développer.
- Linux est très stable, car les composants de haut niveau n'interfèrent pas sur les composants de bas niveau.
- Il est facile à personnaliser, puisque l'utilisateur a le choix des outils à chaque niveau.
- Il a la particularité d'être parfaitement modifiable, puisque si l'on sait programmer, on peut personnaliser les composants du système ou rajouter à tous les niveaux.

## VI. La notion du noyau et de distribution

### VI.1. La notion du noyau

C'est l'assemblage d'un ensemble de logiciel autour d'un noyau linux afin de fournir un système clé en main.

Le noyau d'une distribution peut être mis à jour afin de permettre la prise en compte de matériel récent, toutefois cette manipulation consistant à recompiler le noyau est délicate car elle nécessite un certain niveau de connaissance du système et du matériel.

La plus part des distributions proposent également une installation graphique qui leur est propre ainsi qu'un système de gestion de paquetage permettant d'installer automatiquement des logiciels en gérant les dépendances.

## VI.2. Les distributions

Certaines distributions favorisent la sécurité, d'autre des interfaces graphiques pour les débutants. Les plus connues sont [1,2]:

- **Debian**

Distribution non commerciale régie par le *contrat social Debian*. Elle se distingue par le très grand nombre d'architectures soutenues et, de ce fait, par son cycle de développement relativement long, gage d'une certaine stabilité.

- **Suse**

Développée aujourd'hui en deux versions dont **SuSE Linux Enterprise** pour les besoins d'entreprise (avec de nombreuses certifications matérielles et logicielles). Basé à l'origine sur *Slackware*, *SuSE Linux* puis *SuSE linux Enterprise* est la plus ancienne distribution commerciale encore existante.

- **Knoppix**

Distribution live CD dérivée de Debian. L'installation sur disque dur est possible.

- **Slackware**

Une des plus anciennes distributions encore en développement.

# Chapitre I. Généralités sur linux

---

## - **Mandriva**

Distribution française, dérivée de celle de RedHat, conçue pour être facile d'installation et d'usage. Elle est très appréciée par les débutants. Elle propose aussi un service d'assistance commercial important. Initialement elle s'appelait Mandrake Linux. Elle est aujourd'hui abandonnée.

## - **Ubuntu**

Distribution commerciale fournie par l'entreprise Canonical et dérivée de *Debian*. Orientée grand-public à l'origine, elle existe maintenant en version entreprise. Ses versions stables sont éditées suivant un calendrier strict, suivant un cycle de 6 mois pour des versions dites «intermédiaires» et tous les deux ans pour des versions «supportées sur le long terme» (durée de vie de 5 ans). Ubuntu est disponible en live DVD. C'est une distribution qui est devenue très populaire en peu de temps.

## - **Fedora**

Version communautaire de *Red Hat*. Elle se veut particulièrement à la pointe de la technologie d'où la prise en charge courte (13 mois) et un cycle de développement rapide.

## - **Corel Linux**

**Corel** est un éditeur de logiciels qui a son siège à Ottawa, Ontario, Canada. L'entreprise a été fondée par Michael Cowpland en 1985, et était prévue pour être un laboratoire de recherches.

Corel linux -Debian-basée sur Linux. Le code source a été vendu à Xandros en 2001

## - **Red Hat**

C'est une version commerciale, mais qui est disponible également par téléchargement sur les sites FTP.



Cette distribution se distingue par son sérieux, sa simplicité d'installation, et surtout par son extrême convivialité, de plus, la mise à jour du système est gérée par un système de paquetage.

## VII. Les partitions sous linux

### VII.1. Définition

Le partitionnement est la division du disque dur en section, les partitions peuvent contenir des fichiers programmes, d'autres systèmes d'exploitation et des données.

Les premiers systèmes d'exploitation comme le Dos ne pouvait accéder qu'à un espace limité sur les disques durs, puis les constructions de disque dur commencèrent à augmenter l'espace sur leurs disques durs plus rapidement que le système d'exploitation ne réussissait à accéder à cet espace supplémentaire. Les systèmes d'exploitation peuvent surmonter ce problème en permettant à l'utilisateur de diviser d'autres systèmes ou des données.

Linux n'utilise pas des lettres pour définir les partitions comme le MS-DOS par contre il utilise les noms des répertoires sur différentes partitions, voir sur différents disques durs et même des systèmes sur différentes partitions.

### VII.2. Comment partitionner

Si le disque dur est vierge alors l'installation se lance directement sinon si le disque est vierge et on veut installer aussi Windows on commence par partitionner avec fdisk sous Dos ou avec micro-linux puis on installe Windows sur la première partition puis Linux. Si le disque contient déjà Windows on commence par défragmenter Windows ce qui aura pour effet de rassembler au début du disque tout Windows et libérer de la place pour partitionner et sauvegarder toutes les données.

## VII.3. Les partitions de linux

Presque toutes les distributions obligent l'utilisation d'une partition de **swap** qui servira à écrire lorsque la mémoire vive est débordée.

Les répertoires	Le contenu de chaque répertoire
/	C'est la partition ou se trouve la racine plus en moins grande selon ce qu'on mettra dessus sa saturation est dangereuse pour le système, tout les répertoires se placeront ici, si on leur spécifie pas de partition spécifique.
/bin	Contient des exécutable essentiels au système employé par tous les utilisateurs, les commandes système.
/boot	Contient des fichiers permettant à linux de démarrer.
/home	Contient les données. L'intérêt de les séparer physiquement et de pouvoir par exemple réinstaller Linux sans toucher aux données.
/var	C'est une partition très utile, le système écrit régulièrement ici les données.
/dev	Contient une multitude de fichiers dits spéciaux (pilote des périphériques).
/etc	Contient tous les fichiers de configuration de Linux. On y retrouve par exemple le fichier: / etc / passwd, qui définit les mots de passe des utilisateurs.
/sbin	Contient les binaires du système. On y trouve par exemple la commande shutdown qui permet d'arrêter l'ordinateur.
/lost+found	C'est le répertoire des fichiers perdus. Les fameux fichiers qui, du fait d'erreur disque, se retrouvent sans chemin d'accès .le binaire fsck, qui est lancé régulièrement au démarrage de Linux, se charge de les détecter et de les stocker dans le répertoire / lost+found.
/tmp	C'est un répertoire accessible par tous les utilisateurs du système, il permet de ne pas encombrer son répertoire personnel par des fichiers que l'on souhaite de toute manière détruire ou modifier.
/var/spool	Répertoire des fichiers qui servent de file d'attente. Par exemple, les files d'attente de l'imprimante se trouvent sous ce répertoire. Les données à imprimer, envoyer, sont stockées dans ces files d'attentes jusqu'à ce qu'elles soient traitées.

# Chapitre I. Généralités sur linux

---

/usr	Contient tout ce qui concerne les binaires utiles à tous les utilisateurs et quelques commandes d'administration.
/usr/include	Contient tous les « headers » nécessaires à la programmation.
/usr/lib	Contient toutes les bibliothèques nécessaires au fonctionnement des logiciels. (comme par exemple la bibliothèque C ou C++ ou tcl/tk).
/usr/local	On y met ce que l'on veut, mais surtout les fichiers d'usage local.
/usr/bin	Contient les binaires disponibles pour les utilisateurs et les scripts.

**Tableau 1 : Les répertoires de Linux et leurs contenus [4]**

## Conclusion

Ce chapitre donne les grandes lignes de Linux, avec ses commandes principales pour son bon fonctionnement. Au vu de ses qualités dont l'impact dans le domaine des serveurs, est d'une importance capitale, le système Linux est devenu une alternative incontournable pour les postes de travail de type bureautique ou pour l'hébergement d'applications métiers des entreprises. Ce travail nous a permis de maîtriser et de se familiariser avec l'installation et la configuration matérielle et logicielle des systèmes sous linux.

### Introduction

Ce chapitre présente les principaux points théoriques des serveurs proxy. Proxy est l'une des réalisations les plus importantes dans le monde de la sécurité et de la gestion informatique.

« *Mais pourquoi faire simple et bon marché, quand on peut faire cher et compliqué ?* » C'est sans doute ce raisonnement qui a guidé le CERN (le Centre Européen de Recherche Nucléaire, créateur du web) lorsqu'il conçoit le premier **serveur proxy** (en anglais : proxy-server) en 1994. Ce dispositif informatique, constitué d'un ordinateur équipé d'un logiciel adéquat, est placé *entre le réseau Internet et des stations de travail pourvues d'un navigateur*. Le serveur proxy, qui relaye les requêtes des internautes vers les serveurs branchés sur Internet, constitue une passerelle d'un type spécial. [9]

### I. Définition d'un proxy

Proxy est un dispositif informatique (ordinateur ou logiciel) que l'on interpose entre un réseau de l'entreprise et le réseau internet, constituant une protection pour le réseau de l'entreprise et de l'internet, il peut également faire office de cache c'est-à-dire qu'il enregistre les pages transférées par les utilisateurs pour les délivrer sans qu'il soit nécessaire de se connecter au serveur initial. Toutefois, il peut exister des serveurs Proxy pour chaque protocole applicatif. On peut voir ci-dessous, un schéma qui montre l'architecture d'un réseau avec Proxy :

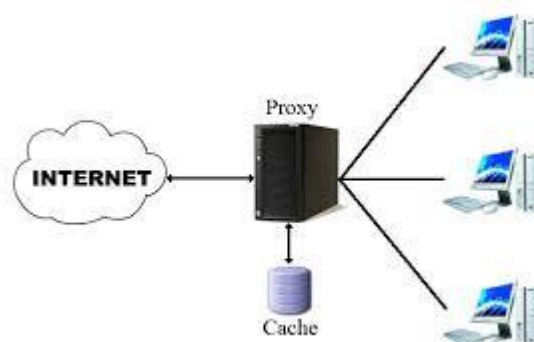


Figure 2 : la position de Proxy

### II. Le fonctionnement d'un proxy

Côté client : il faut vérifier les conditions suivantes :

- Un logiciel client personnalisé : avec cette approche, le logiciel doit savoir comment contracter le serveur mandataire au lieu du serveur réel. Quand un utilisateur lance une requête (par exemple, pour ftp ou telnet), et comment dire au serveur mandataire à quel serveur réel se connecter.
- Des procédures utilisateurs personnalisées : avec cette approche, l'utilisateur emploie un logiciel client standard pour s'adresser au serveur mandataire et pour lui dire de se connecter au serveur réel, au lieu de communiquer directement avec ce dernier.

### III. Les principaux objectifs des serveurs proxy

Installer un système de Proxy cache, principalement pour **http** (éventuellement aussi pour **ftp**) propose deux fonctions principales :

- l'optimisation de la bande passante sur le lieu Internet, lorsque un certain nombre de clients est connecté et qu'ils veulent visiter plus ou moins les mêmes sites.
- le contrôle et le filtrage de l'accès à la toile, en se servant des URI et éventuellement des noms d'utilisateurs, si l'on fait de l'authentification de ces derniers.

### IV. Le rôle du proxy

- Un serveur Proxy stocke provisionnement les pages Web que les utilisateurs cherchent sur Internet, si un internaute requiert une information qui se trouve déjà dans le cache, il sera servi plus rapidement dans le cas contraire, il sera servi plus lentement.
- Un serveur Proxy garde une trace de tous les échanges d'information qui se transmettent : L'adresse de l'internaute, l'heure, le site Web demandé et les pages consultées.
- Un serveur Proxy peut être configuré de telle sorte qu'il examine l'information qui le traverse, et qu'il refuse de délivrer les fichiers contenant une chaîne de caractère donnée.

- Un serveur Proxy peut masquer les informations concernant notre ordinateur, en effet quand nous surfons, tous les sites Web peuvent savoir de quel site nous verrons, quel navigateur nous utilisons, quel est notre système d'exploitation, notre adresse IP ...etc.

Certains Proxy masquent ces informations ; ces Proxy sont dits Proxy anonymes.

### V. Les types de proxy

- Les Proxys parfaitement anonymes : sont ceux qui ne changent pas les champs des requêtes et montrent notre navigateur réel mais notre adresse IP réelle et réellement cachée.
- Les Proxys transparents : changent les champs de nos requêtes en incluant notre adresse IP réelle.
- Les proxys les plus connus sont : http, ftp, smtp , dns , socks et https.

### VI. Les dangers de proxy

- **Confidentialité** : si on demande tous nos pages au proxy, il peut savoir tous les sites que nous avons visités.
- **mot de passe** : certains sites web nécessitent des mots de passe. Comme on passe par le Proxy, il va le connaître (sauf si on utilise http/ssl).
- **Modification** : le proxy nous fournit les pages demandées mais il est possible qu'il les modifié à la volée avant de nous les donner (cela reste rare mais possible).
- **censure** : certains proxys peuvent être configurés pour censurer des sites.

### Conclusion

Un serveur Proxy est un dispositif de sécurité recommandé. Il peut jouer un rôle de cache efficace. Dans ce projet on va s'intéresser à MP-TCP qui est un proxy avec une infrastructure redondante, rendre un protocole à voie unique capable de créer plusieurs sous-connexions afin d'équilibrer les charges sur le réseau.

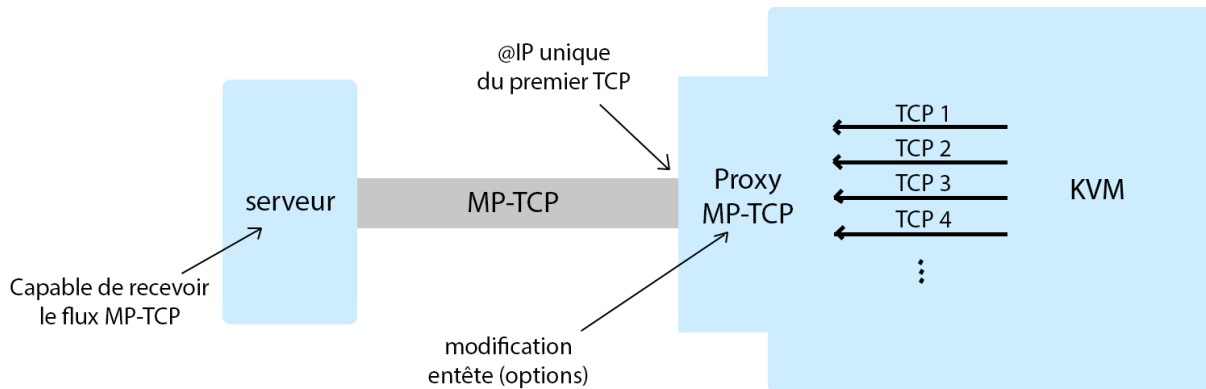
## Introduction

Dans ce chapitre on va détailler notre projet pour bien comprendre le proxy multipath-tcp, son intérêt et son fonctionnement.

La première implémentation de MP-TCP et l'architecture du système d'exploitation a été faite par le Dr. Sébastien Barré. Il a commencé ce projet vers l'année 2009, il s'est basé sur son shim6-implémentation (est un multi-thread C++ des processus de l'espace utilisateur, qui s'interface avec un noyau Linux standard) et il a continué à travailler sur l'implémentation de MP-TCP jusqu'à ce qu'il a terminé sa thèse. Et depuis, plusieurs personnes ont rejoint ce projet. Ces implémentations seront un plus pour notre projet.

## I. Description

### I.1 objectif



**Figure 3 : Modèle de communication (hyperviseur – serveur)**

Permettre à un Hyperviseur (KVM dans le cas de notre projet) hébergeant plusieurs clients (machines virtuelles) de transformer le trafic TCP sortant de chacune des machines virtuelles en un trafic MP-TCP afin d'établir une communication MP-TCP avec le Serveur qui implémente aussi MP-TCP. Des règles de routage seront appliquées pour apporter des modifications sur la connexion TCP (sortantes)/MPTCP (entrante), on devra ajouter une signalisation MP-TCP pour les (sous-flux) réinjectés dans le réseau tout en gardant une transparence au niveau des hôtes.

# Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

---

Nous comparerons également les performances de cette mise en œuvre avec TCP sur Les serveurs web. Enfin, nous discuterons les leçons tirées de la conception MP-TCP.

## I.2 Environnement

- Linux (Ubuntu).
- Hyperviseur (KVM).
- Proxy (Middle-box) MP-TCP.
- Configuration de MP-TCP au niveau de la Middle-Box.

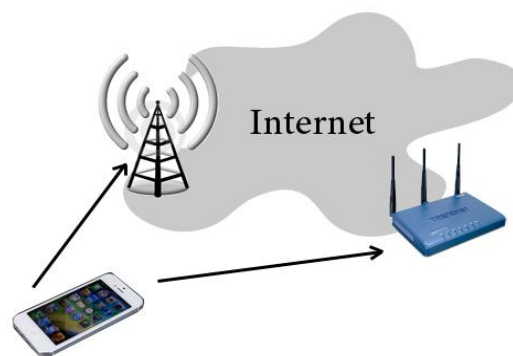
Pour commencer on va s'intéresse au déploiement de règle de partage au niveau de la Middle-Box gérée par un hyperviseur.

Pour pouvoir améliorer la gestion du trafic MP-TCP, on a besoin de configurer un proxy MP-TCP; en utilisant un hyperviseur capable de prendre la décision d'utiliser TCP ou MP-TCP suivant ces règles :

La durée et l'intensité du trafic (les paquets échangés) à envoyer, s'il (trafic) est faible on se contente d'utiliser TCP, sinon l'activation de l'option Multi-Path TCP est nécessaire. Si un des liens est congestionné, on va partager la charge sur d'autres liens.

## II. Contexte technologique

Les réseaux sont devenus multi-chemins, et les terminaux mobiles peuvent se connecter à internet via différentes interfaces.



**Figure 4 : connexion mobile – internet**

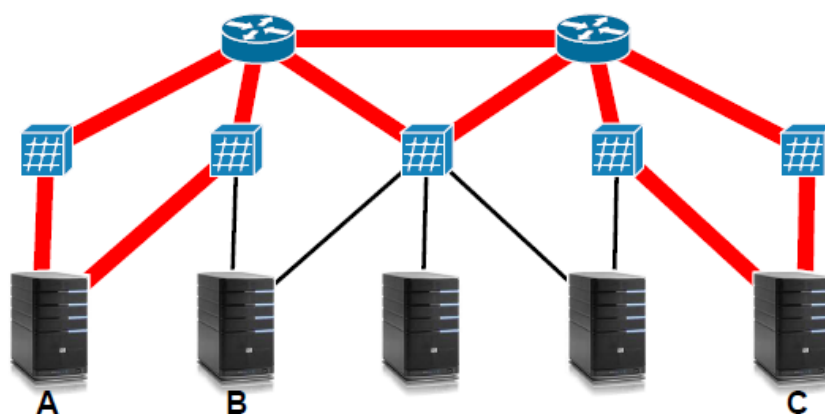


## Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

---

Une autre raison pour laquelle les informaticiens ont voulu mettre en place les réseaux TCP multi-chemins.

On parle ici de l'environnement Cloud, disposant d'une infrastructure redondante. Cela veut dire que plusieurs chemins physiques mènent vers le même serveur ou le même emplacement de données sur internet.



**Figure 5 : chemins redondants vers les mêmes serveurs**

Dans cette architecture, par exemple, si on ouvre plusieurs connexions TCP indépendantes on aura des problèmes de collisions, ce qui va réduire la bande passante et par conséquent il n'y aura pas d'équilibrage de charge (Load-Balancing).

Le but est d'utiliser les différents chemins pour une même connexion TCP ouverte. Donc comment rendre un protocole à voie unique capable de créer plusieurs sous-connexions afin d'équilibrer les charges sur le réseau, et éviter au maximum les collisions et la congestion, tout en conservant les applications existantes sur les machines? Il va falloir adapter la couche transport seulement à cet objectif.

### II.1. Etat de l'art

Plusieurs efforts ont déjà été déployés pour avoir un protocole de transport qui puisse utiliser plusieurs chemins simultanés, « *Multipath TCP* » se distingue par le fait que son trafic

## Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

---

apparaîtra comme du TCP habituel, aussi bien aux applications qu'aux équipements intermédiaires situés sur le réseau, les deux points sur lesquels SCTP (Stream Control Transmission Protocol) a largement échoué. SCTP est donc plus propre, plus conforme à l'architecture originale de l'Internet mais *Multipath TCP* est peut-être plus réaliste. Mais SCTP ne permet pas actuellement d'envoyer des paquets sur différents chemins.

Les buts de l'architecture *Multipath TCP* du point de vue des fonctions est :

- Augmenter le débit, par l'utilisation simultanée de plusieurs ressources réseaux. Plus formellement, l'utilisation de *Multipath TCP* doit mener à un débit au moins égal à celui du meilleur des chemins.

- Augmenter la résistance aux pannes. La résistance totale de *Multipath TCP* doit être au moins aussi bonne que celle du chemin le plus fiable.

Au début, MP-TCP était juste une idée loin d'être mise en pratique dont beaucoup de point ont vu le jour récemment, Les RFC sont modifiées régulièrement grâce aux nombreuses découvertes sur MPTCP.

Des chercheurs ont prouvé que MPTCP pouvait améliorer le débit TCP, en particulier dans la réalisation d'un fluide de transfert entre la 3G et le Wi-Fi, dans le cas de la mobilité.

Cependant, un hôte ne peut bénéficier de MPTCP que si le serveur cible prend également en charge MPTCP mais malheureusement la plupart des serveurs actuels n'adoptent pas ce dernier. C'est ce qui inspire l'idée de mettre en place un proxy MPTCP qui peut aider un client MPTCP d'établir une connexion avec un serveur MPTCP en héritant de TCP.

La première implémentation de MP-TCP et l'architecture du système d'exploitation a été faite par le Dr. Sébastien Barré. Et depuis, plusieurs personnes ont rejoint ce projet, récemment des développeurs actifs comme CHRISTOPH PAASCH, JAAKO KORKEANIEMI et GREGORY DETAL.

Ces implémentations seront d'un très grand intérêt dans notre projet.

## II.2. Décisions de conception

Les objectifs sont les suivants:

- Trouver un protocole aussi bon que TCP notamment en termes de qualité de service.
- Ne pas perdre en capacité.
- Contrôle de congestion similaire à celui de TCP, mais avec plus d'efficacité, car il y'a partage de chemins.
- Aucun changement dans les applications existantes(API).
- Établissement de connexions sécurisées.

## III. Analyse

### III.1. Fonctionnement du Protocole TCP

TCP -*Transmission Control Protocol*- est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (couche réseau - protocole IP). Lorsque les données sont fournies au protocole IP, celui-ci les encapsule dans des datagrammes IP, en fixant le champ protocole à 6 pour indiquer que le protocole en amont est TCP. TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission. Les caractéristiques principales du protocole TCP sont les suivantes :

- remettre en ordre les datagrammes en provenance du protocole IP.
- vérifier le flot de données afin d'éviter une saturation du réseau (la congestion).
- formater les données en segments de longueur variable afin de les "remettre" au protocole IP.

# Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

---

- multiplexer les données, c'est-à-dire de faire circuler simultanément des informations provenant de sources distinctes sur un même chemin.
- initialiser et fermer une communication de manière courtoise. [5]

## III.1.1 La vie d'une connexion TCP

Une connexion TCP est établie en trois temps de manière à assurer la synchronisation nécessaire entre les extrémités.

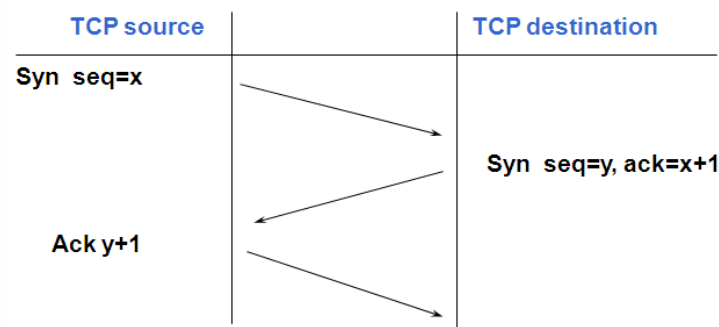


Figure 6 : poignée de main TCP

Une connexion TCP est libérée en un processus dit "trois temps modifié".

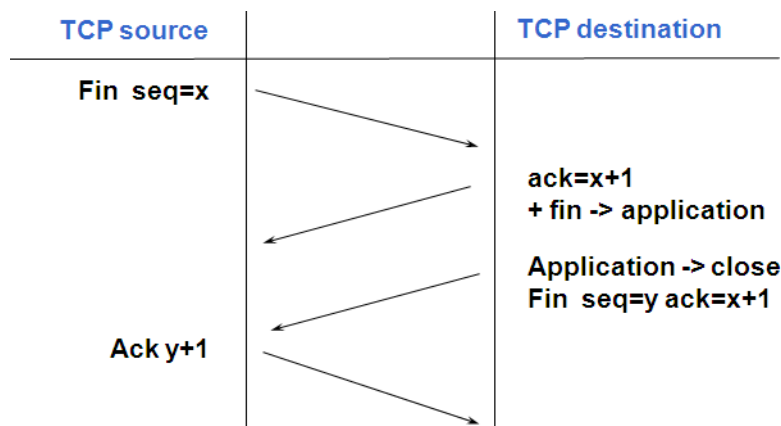


Figure 7 : fermeture de connexion TCP

## III.2. Fonctionnement de MP-TCP

MP-TCP est une extension (option) du protocole TCP, quand le client et le serveur utilisent MP-TCP les informations contenues dans les options seront utiles. Ces connexions multi-chemin TCP sont transparentes vis-à-vis des firewalls, elles sont considérées comme une seule connexion TCP, donc c'est l'un des principaux avantages par rapport au TCP classique.

Quand le serveur ne supporte pas MP-TCP le protocole bascule directement vers TCP classique, et quand plusieurs connexions sont possibles MP-TCP crée des sous-flux, chaque sous-flux peut être vu comme une connexion d'interface vers interface. MP-TCP essaie de créer des sous-flux entre toutes les interfaces.

Un aperçu de vision haut niveau sur comment MP-TCP fonctionne est illustré dans la figure suivante :

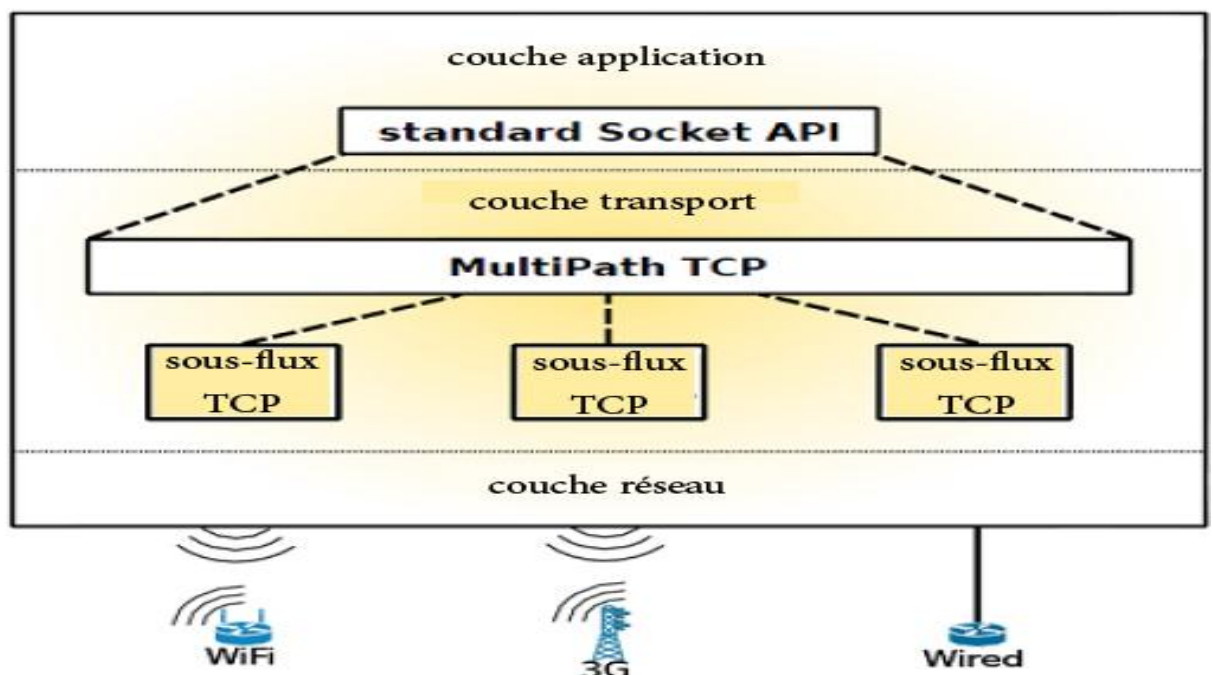


Figure 8 : modèle avec couches (MP-TCP) [6]

Le protocole MP-TCP doit pouvoir ajouter et supprimer dynamiquement les sous-flux TCP selon les besoins de l'utilisateur, tout en restant transparent vis-à-vis du récepteur/émetteur. Si

## Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

---

le protocole détecte de la congestion sur un des chemins, il doit basculer immédiatement vers un autre chemin où il n'y a pas de congestion, et c'est ce qui explique le choix du protocole TCP pour cette affaire.

Un autre élément qui encourage à utiliser TCP et pas un autre protocole, c'est la facilité d'ajout et de gestion des sous-flux, ainsi TCP est utilisé dans la plupart des applications qui exigent un transport fiable des données.

### III.2.1 Eléments important utilisé par MP-TCP

#### **RTT**

MP-TCP utilisera le RTT (Round Trip Time, le temps que prend un paquet de données avant d'être acquitté, donc c'est le temps d'aller/retour), dans ses algorithmes de congestion.

#### **MTU**

La MTU (The Maximum Transmission Unit) est la taille en octets des données que le protocole de transport peut transporter. Et c'est aussi la taille à partir de laquelle la fragmentation des paquets sera nécessaire.

#### **MMS**

Le MMS (Maximum Segment Size) est le paramètre utilisé par le protocole TCP. Il précise la taille limite que les paquets ne doivent pas dépasser. Pour cette raison, la taille de l'entête + MSS doit être inférieure ou égale à la MTU.

Le MSS pour MP-TCP est de 1400 octets mais il pourra être modifié, ce que nous ferons pour avoir des performances élevées dans les liens à haute vitesse, le MSS sera donc augmenté à environ 8900 octets.

## III.2.2. Initialisation

### Poignée de main MP-TCP (MP-TCP Handshake) :

Dans cette partie on va expliquer comment fonctionne la poignée de main de MP-TCPet à quoi elle ressemble, la figure ci-dessous montre bien le Handshake MP-TCP.

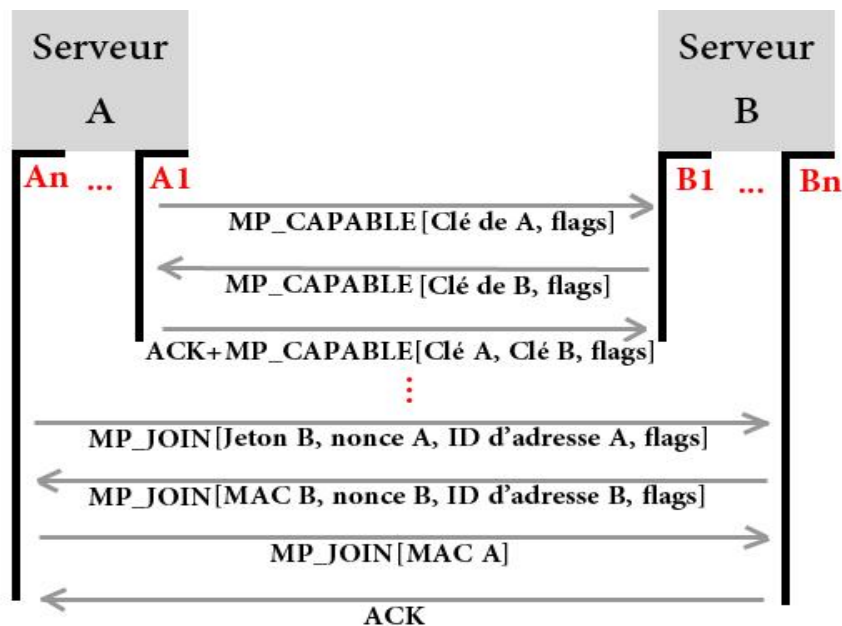


Figure 9 : poignée de main MP-TCP

Premièrement le 'Serveur A' envoie un paquet SYN vers le 'Serveur B', de la même manière que ce qui se passe dans la connexion TCP ordinaire. Ce paquet contient ce pendant l'option `MP_CAPABLE` dans le champ options de TCP. Après le 'Serveur B' répond avec un paquet SYN/ACK, et aussi avec l'option `MP_CAPABLE` (quand il supporte l'utilisation de MP-TCP). Les deux paquets contiennent les clés d'authentification et des informations additionnelles nécessaires pour MP-TCP. A la fin le 'Serveur A' répond avec un ACK vers le 'Serveur B' en lui envoyant les deux clés des paquets précédents.

## III.2.3. Echange de données

La première connexion (sous-flux) est établie de la même manière que la poignée de main de TCP. Le protocole va par la suite essayer de créer des sous-flux additionnels.

## Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

---

Une fois que les paquets MP\_CAPABLE sont acceptés des deux côtés, la création des sous-flux est possible par l'émetteur seulement. La création des sous-flux se fera de la même façon que le premier sous-flux (Master Subflow) c'est-à-dire un SYN suivi de SYN/ACK puis un ACK (Handshake) mais en ajoutant une option MP\_JOIN pour que le 'Serveur B' ne refuse pas la nouvelle connexion. Le 'Server A' envoie avec le premier SYN un jeton 'jeton du serveur B', qui est généré à l'aide des clés échangées précédemment. Ce jeton est utilisé pour identifier et vérifier que cette nouvelle communication est bien la suite de celle établie auparavant. Aussi un NONCE est envoyé pour protéger cet échange des attaques par replay.

L'adresse ID est envoyé pour spécifier l'adresse et pour identifier le sous-flux de façon unique parmi les autres sous-flux existants et ceux qui vont s'ajouter après, car par la suite et dans le cadre de ce projet on va devoir identifier la connexion qui rencontre des problèmes de congestion et cela sera possible à l'aide de cet identifiant unique. Grace à cet ID la connexion est indépendante de l'adresse source. Les flags peuvent être utilisés pour des options plus avancées (par exemple : utiliser ce lien seulement comme lien de secours).

Lorsque le MP\_JOIN SYN du sous-flux est vérifié par le 'Serveur B' il répond avec un SYN/ACK, l'ACK contient un code d'authentification de message (Message Authentication Code (MAC)), un NONCE aléatoire et l'ID du 'Serveur B'. Le MAC est utilisé pour authentifier le sous-flux. Après cela le 'Serveur A' répond avec un paquet ACK, l'option MP\_JOIN et les données additionnelles pareil que le 'Serveur B', puis le 'Serveur B' doit confirmer la réception de ce paquet par un autre paquet ACK vers le 'serveur A' qui est maintenant sûr que le 'Serveur B' a reçu le paquet.

SHA-1 a été la technique d'authentification utilisée dans tous les chiffrements précédents.

### III.2.4. Les problèmes potentiels de MP-TCP

#### L'impact des Middle-Boxes

MP-TCP a été conçu pour passer par la majorité de Middle-Boxes. L'évidence suggère que des nouvelles options TCP puissent être utilisées avec succès sur la plupart des chemins sur le réseau.

Certaines Middle-Boxes peuvent toujours refuser de passer des messages MP-TCP en raison de la présence d'options TCP, si c'est le cas, MP-TCP chute. Bien que ceci ne crée pas



## Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

---

de problèmes pour l'application (sa communication sera installée (configurée) d'une façon ou d'une autre), il peut y avoir un retard supplémentaire tandis que la première poignée de main échoue. Donc, une approche alternative pourrait être d'essayer tant MP-TCP que l'habitué TCP en tentant des connexions en même temps semblables au mécanisme de IPv6. On pourrait aussi mettre un temps mort plus court sur le MP-TCP ce qui réduit ainsi le retard de configuration, si le repli à l'habituel TCP est nécessaire.

Il peut aussi y avoir des Middle-Boxes qui changent d'une manière transparente la longueur des paquets. Si de tels Middle-Boxes sont présentes, la paquetisation des flux d'octet au niveau du récepteur sera difficile.

Cependant, MP-TCP peut détecter ce type de Middle-Boxes et passer ensuite vers l'habitué TCP.

Pour pouvoir établir un lien entre les connexions TCP des machines hôtes dont la connexion MP-TCP reste transparente, On va implémenter un proxy MP-TCP dans un Hyperviseur KVM pour faciliter le transit de ces paquets TCP en des paquets MP-TCP.

Mais notre priorité est de trouver le moyen de rebondir d'une connexion TCP à une connexion MP-TCP, grâce à l'implémentation d'un proxy au niveau d'un Hyperviseur KVM, pouvant gérer à la fois les flux entrants et sortants. La difficulté qui se pose à ce niveau est de pouvoir mettre en place la gestion des paquets MP-TCP (avoir déjà installé MP-TCP sur le noyau au préalable) pour pouvoir le relier à l'hyperviseur. On a commencé par créer des VM dans le parc de l'hyperviseur KVM, on a pu analyser le trafic sur chaque interface pour s'assurer du bon fonctionnement des interfaces gérées par l'hyperviseur. A partir de ces interfaces on peut analyser les paquets de chaque VM à l'aide d'un proxy-MP-TCP qui sera configuré comme suite :

- Le trafic entrant en MP-TCP va passer en un trafic TCP au sein du parc des VMs.
- Le trafic TCP sortants des VMs va être commuté par la Middle-Boxe en un trafic MP-TCP.

## IV. Hyperviseur KVM

KVM (Kernel-based Virtual Machine) est une technologie de virtualisation qui permet d'exécuter plusieurs systèmes d'exploitation, il peut aussi être vu comme un 'Fork' de QEMU qui est un logiciel d'émulation de processeurs et de machines virtuelles permettant d'exécuter un ou plusieurs systèmes d'exploitation via l'hyperviseur.

Avec KVM les systèmes d'exploitation peuvent tous être exécuté en même temps et tous utilisés comme s'ils étaient physiquement des ordinateurs distincts.

## Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

Les processeurs de la famille x86 disposent de 4 niveaux de privilège d'exécution, Appelés anneaux de protection. En plus de ces anneaux, les processeurs récents intègrent un nouveau mécanisme d'isolation qui permet à un système invité d'être exécuté directement sur le processeur quels que soient les anneaux de protection qu'il requiert pour fonctionner, tout en assurant la protection contre la lecture et l'écriture dans la mémoire n'appartenant pas à la machine virtuelle. Cela est principalement possible grâce à une MMU dédiée à chaque machine virtuelle sur le processeur et à une redirection des interruptions matérielles vers l'hyperviseur de machines virtuelles (KVM en l'occurrence) afin d'émuler le matériel dont a besoin le système invité.

L'architecture globale est schématisée dans la figure ci-dessous :

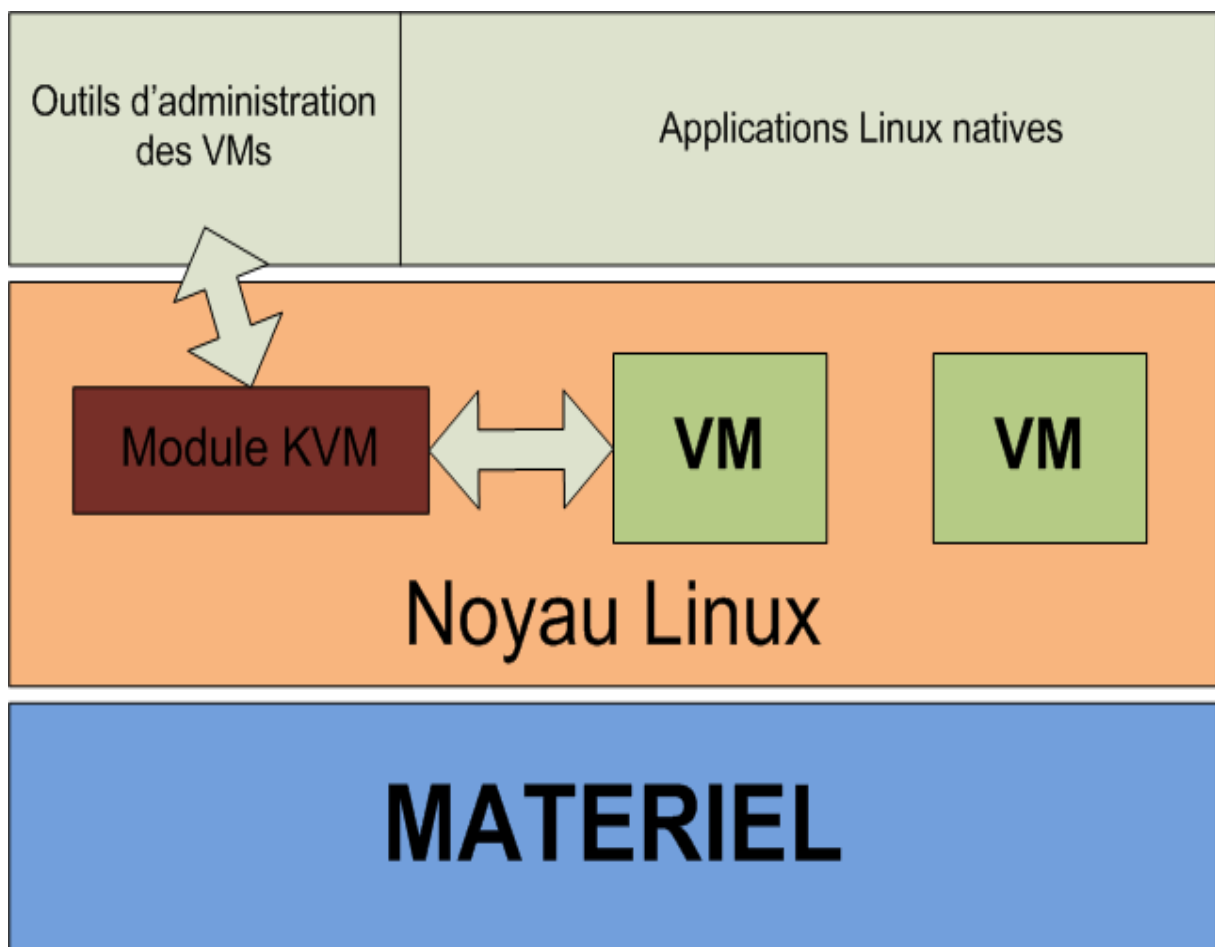


Figure 10: Architecture globale KVM [7]

## IV.1. Architecture KVM

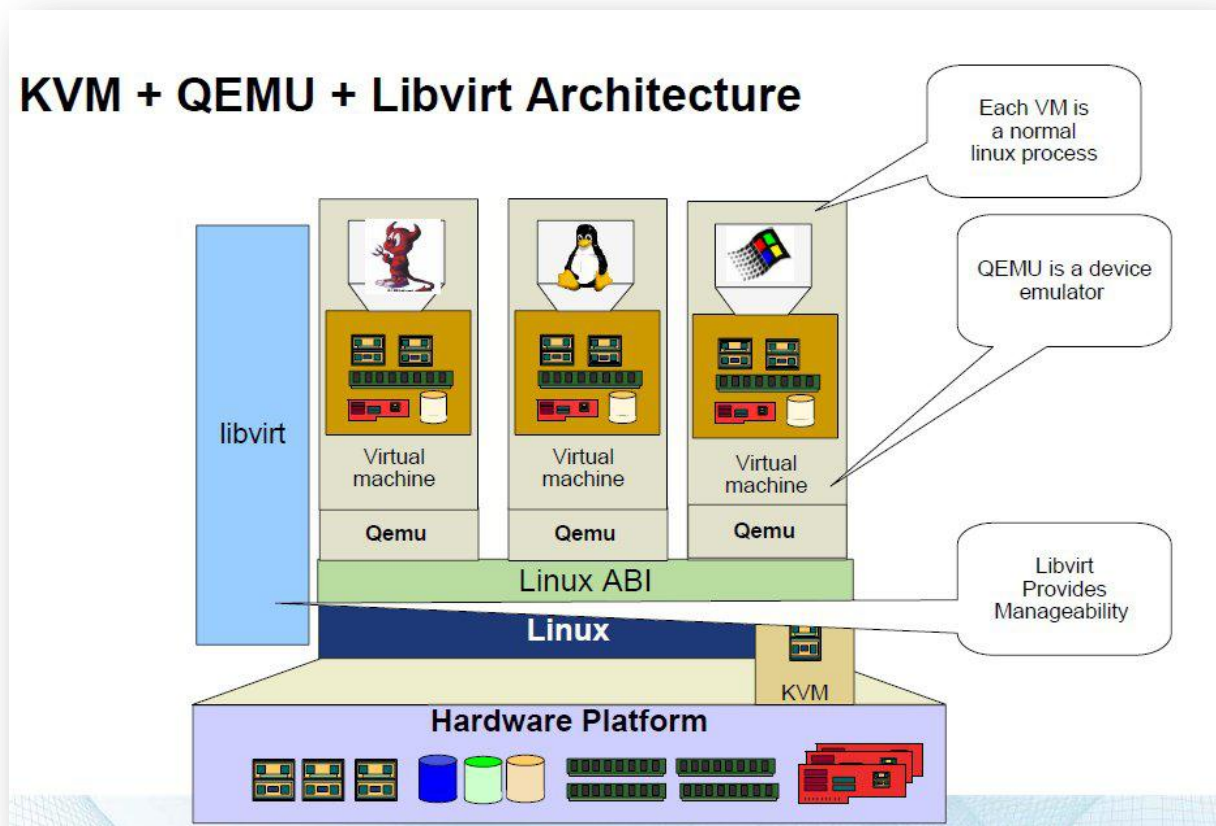


Figure 11 : Architecture détaillé KVM [8]

Depuis quelques années, les processeurs disposent de fonctionnalités natives dédiées à la virtualisation qui les rendent plus simplement “virtualisables”. Cela facilite grandement l’écriture d’un hyperviseur.

- Hyperviseur (ou Virtual Machine Monitor VMM) :
  - couche intermédiaire entre le matériel et les systèmes d'exploitation invités
  - implémente la plupart des services d'un noyau d'OS (gestion de la mémoire, scheduler, IO...)
  - répartit les ressources du système hôte entre les invités.

## Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

---

- Les objectifs de KVM :

- ne pas “réinventer la route” en développant un hyperviseur “complexe”
- mais s'appuyer sur le noyau standard “Linux” en intégrant la virtualisation sous forme de module.

Ce choix de développement permet à KVM de bénéficier des nouveautés du noyau Linux.

Chaque VM s'exécute côté hôte sous la forme d'un processus standard géré par le scheduler standard de Linux.

- Classiquement, un processus possède deux modes d'exécution : kernel et user :

- le mode “user” est le défaut pour les applications.
- une application entre en mode “kernel” lorsqu'elle a besoin d'accéder à des fonctionnalités “noyau” (écrire sur les disques par exemple).
- KVM ajoute un troisième mode : le mode “guest”.
- le mode “guest” dispose de ses propres modes “user” et “kernel”.

- Les composants de KVM :

- un “driver” permettant d'exploiter les fonctionnalités matérielles de virtualisation des processeurs. Celui-ci se présente sous la forme de trois modules (kvm, kvm-intel, kvm-amd) fournissant le périphérique /dev/kvm.

- un composant côté “utilisateur” pour l'émulation du matériel : version de QEMU.

- Les entrées/sorties disques sont gérés de deux manières :

- émulation IDE/SCSI fournie par QEMU.
- para virtualisation via les drivers “virtio” présents depuis le noyau Linux 2.6.25.

# Chapitre III. Etude et Présentation d'un proxy multipath – tcp au niveau Hyperviseur

---

## IV.2. But et pré-requis

Utiliser les fonctions de MP-TCP dans l'hyperviseur KVM afin de pouvoir générer des trafics MP-TCP en capturant les trafics TCP depuis les VMs et les transformer en MP-TCP au niveau de notre Middle-Box, aussi faire la modification dans l'autre sens du trafic, c'est-à-dire pour les paquets MP-TCP entrants vers les VMs les modifier en paquets TCP ce qui nous permettra de garder la transparence sur nos VMs.

Pour ce faire il faudra installer KVM et ajouter des VMs dans l'hyperviseur.

## Conclusion

Après avoir étudié comment est gérée la commutation des paquets du réseau au niveau de l'hyperviseur, on a étudié comment intégrer les fonctionnalités de MP-TCP qui nous permettent d'utiliser des chemins multiples pour faire transiter notre trafic au lieu d'utiliser une seule connexion TCP classique. Ceci va nous permettre vraisemblablement d'augmenter les performances du réseau. On devra aussi configurer les différentes interfaces de routage qui vont par la suite acheminer le trafic sur les différents chemins créés par le MP-TCP.

# Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

## Introduction

L'Internet a été conçu pour être extensible; les routeurs ne se soucient pas des entêtes IP, à cause de la contenance des paquets, et des protocoles tels que l'IP et TCP qui ont été conçus avec des champs qui pourraient être utilisés pour ajouter des options avec des fonctionnalités supplémentaires.

Les Options IP ont été inutilisables pendant vingt ans [10], car ils provoquent des routeurs pour traiter les paquets sur leur chemin à caractère lent. Au-dessus d'IP, l'Internet a bénéficié ou subi, selon notre point de vue des décennies d'optimisation et d'amélioration sur le plan de la sécurité.

Pour améliorer les performances [11,12,13,14], réduire l'exposition de la sécurité [15, 17], renforcer le contrôle et le travail autour de la pénurie d'espace d'adressage [16], l'Internet a connu une invasion de boîtiers de médiation (la middle-box). Cette boîte ne se soucie pas de ce que les paquets contiennent et exécutent un traitement à la couche 4 ou à une couche supérieure au sein du réseau. Le problème confronté maintenant par les concepteurs de nouveaux protocoles est qu'il n'y a pas plus d'une manière bien définie ou comprise pour étendre la fonctionnalité de réseau, à court de mise en œuvre tout sur HTTP [20]. Récemment, ils ont travaillé sur l'ajout d'un soutien de trajets multiples [19] et du cryptage natif [18] pour TCP. Le moyen évident de le faire, dans les deux cas, est d'utiliser les options TCP. Dans le cas de la multivoie, nous tenons également à répartir les données sur plus d'un chemin.

Nous avons besoin de la forcer à avoir la possibilité de fusionner. Enfin, nous examinons les implications de notre étude de mesure pour que les concepteurs de protocole souhaitent étendre les fonctionnalités de TCP. En particulier, nous examinons les propositions de trajets multiples TCP [19], TcpCrypt [18], et TCP Options étendues [20], et considèrent que nos résultats signifient pour la conception de ces protocoles et leur déploiement.

# Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

## I. Plan de développement

Dans le cadre de notre projet, on a procédé comme suit :

### I.1 Installation et manipulation de KVM :

Commencer par l'installation de l'hyperviseur KVM sur une machine avec un noyau Linux qui supporte la virtualisation. On a appris à manipuler KVM et maîtriser ses différentes fonctionnalités, en effectuant des tests et en créant des machines virtuelles dont on a besoin dans ce projet.

### I.2 Identification de codes à modifier dans KVM :

L'étape cruciale dans notre projet est d'identifier la partie dans le code KVM qui fait la commutation vers l'extérieur du réseau.

En parcourant les différentes bibliothèques de KVM, et en essayant de très bien comprendre leurs rôles on a remarqué qu'il était impossible de mettre en place un code pour agir sur les paquets TCP, de façon transparente au niveau des machines virtuelles.

On a donc essayé de trouver une autre solution pour agir sur les connexions des machines virtuelles.

### I.3 Implémentation du Proxy MP-TCP au niveau de l'hyperviseur KVM :

Après avoir étudié comment est gérée la commutation des paquets du réseau au niveau de l'hyperviseur, on a intégré les fonctionnalités de MP-TCP qui nous permettront d'utiliser des chemins multiples pour faire transiter notre trafic au lieu d'utiliser une seule connexion TCP classique. Ceci va nous permettre vraisemblablement d'augmenter les performances du réseau. Pour cela on doit aussi configurer les différentes interfaces de routage qui vont par la suite acheminer le trafic sur les différents chemins créés par le MP-TCP.

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

### I.4 Test et comparaison des résultats :

Avant de tester le protocole MP-TCP sur l'hyperviseur KVM, il était plus judicieux d'effectuer des tests sur une machine avec un kernel qui sera compilé en intégrant les fonctionnalités MP-TCP, ces tests nous ont permis de mieux comprendre le fonctionnement de MP-TCP à l'état pur.

Les tests seront par la suite basés sur l'environnement KVM, car en effet, les machines virtuelles ne vont pas s'occuper des trafics MP-TCP, car ça c'est TCP qui sera utilisé à ce niveau. En revanche une Middle-Box qui sera implémentée dans l'hyperviseur se chargera de capturer les paquets TCP et apporter des modifications au-dessus pour ajouter les options MP-TCP. Le même travail sera effectué lors de la réception d'un paquet MP-TCP, qu'il faudra transmettre à l'hôte en tant que paquet TCP classique en apportant là aussi, les modifications nécessaires.

On va comparer les performances de l'utilisation de MP-TCP par rapport à l'utilisation de TCP et voir les différences notamment en termes de débit, et voir si l'utilisation de MP-TCP doit prendre en compte quelques contraintes au niveau physique, réseau et applicatif.

## II. Installation de KVM

Tout d'abord, on commence par vérifier que notre processeur supporte la virtualisation matérielle. La réponse est « oui », mais il reste à activer ce support dans le BIOS. Pour cela, il faut taper la commande suivante:

```
$ egrep '^flags.*(vmx|svm)' /proc/cpuinfo
```

On a eu un résultat qui s'est affiché : c'est que le processeur supporte bien la virtualisation.

On peut aussi taper la commande :

```
$ kvm-ok
```



## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

Ce résultat confirme que notre machine est bien capable de faire une virtualisation matérielle complète sans aucun problème.

Puis on opère à l'installation du paquet QEMU-KVM, Qemu qu'on installe avec KVM va permettre d'émuler les architectures des processeurs x86, ARM, PowerPC, Sparc, MIPS..., et d'exécuter un ou plusieurs systèmes d'exploitation (et leurs applications) de manière isolée sur une même machine physique. QEMU fonctionne sur les plateformes x86, x64, PPC, Sparc et ARM et fonctionne sous les systèmes d'exploitation Linux, FreeBSD, NetBSD, OpenBSD, Mac OS X, Unix et Microsoft Windows.

Donc cet émulateur (qemu) de processeur va renforcer notre KVM et on ne s'inquiètera plus si on installe différents systèmes d'exploitation sur nos machines virtuelles.

Tout d'abord nous devons installer le MP-TCP sur le kernel Ubuntu de la machine hôte, afin de se familiariser avec KVM et l'implémenter au niveau KVM . C'est d'ailleurs ce qui constitue bien la problématique principale de notre projet.

- **Installation de kvm en ligne de commande :**

La commande qui permettra l'installation de kvm est la suivante :

```
$ Sudo apt-get installkvmqemu bridge-utilsuml-utilities
```

Avec la possibilité de créer des ponts et des passerelles grâce à l'utilitaire bridge-utils. Ce package contient rend possible la configuration du pont Ethernet de Linux (the bridge). Le bridge va être utilisé pour connecter plusieurs interfaces Ethernet entre elles. La connexion est totalement transparente: les hôtes connectés à une seule interface Ethernet voient les hôtes de l'autre côté comme s'ils étaient connectés directement.

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

Un groupe KVM se créer automatiquement suite à l'étape précédente :

```
scanner:x:116:  
colord:x:117:  
lightdm:x:118:  
nopasswdlogin:x:119:  
avahi:x:120:  
pulse:x:121:  
pulse-access:x:122:  
saned:x:123:  
hassai:x:1000:  
sambashare:x:124:hassai  
kvm:x:125:  
hassai@ubuntu:~$
```

Figure 12 : Imprimé écran – groupe KVM.

Après on ajoute l'utilisateur au groupe KVM

```
$ Sudo adduser hassai kvm
```

```
pulse-access:x:122:  
saned:x:123:  
hassai:x:1000:  
sambashare:x:124:hassai  
kvm:x:125:  
hassai@ubuntu:~$ sudo adduser hassai kvm  
Adding user `hassai' to group `kvm' ...  
Adding user hassai to group kvm  
Done.  
hassai@ubuntu:~$
```

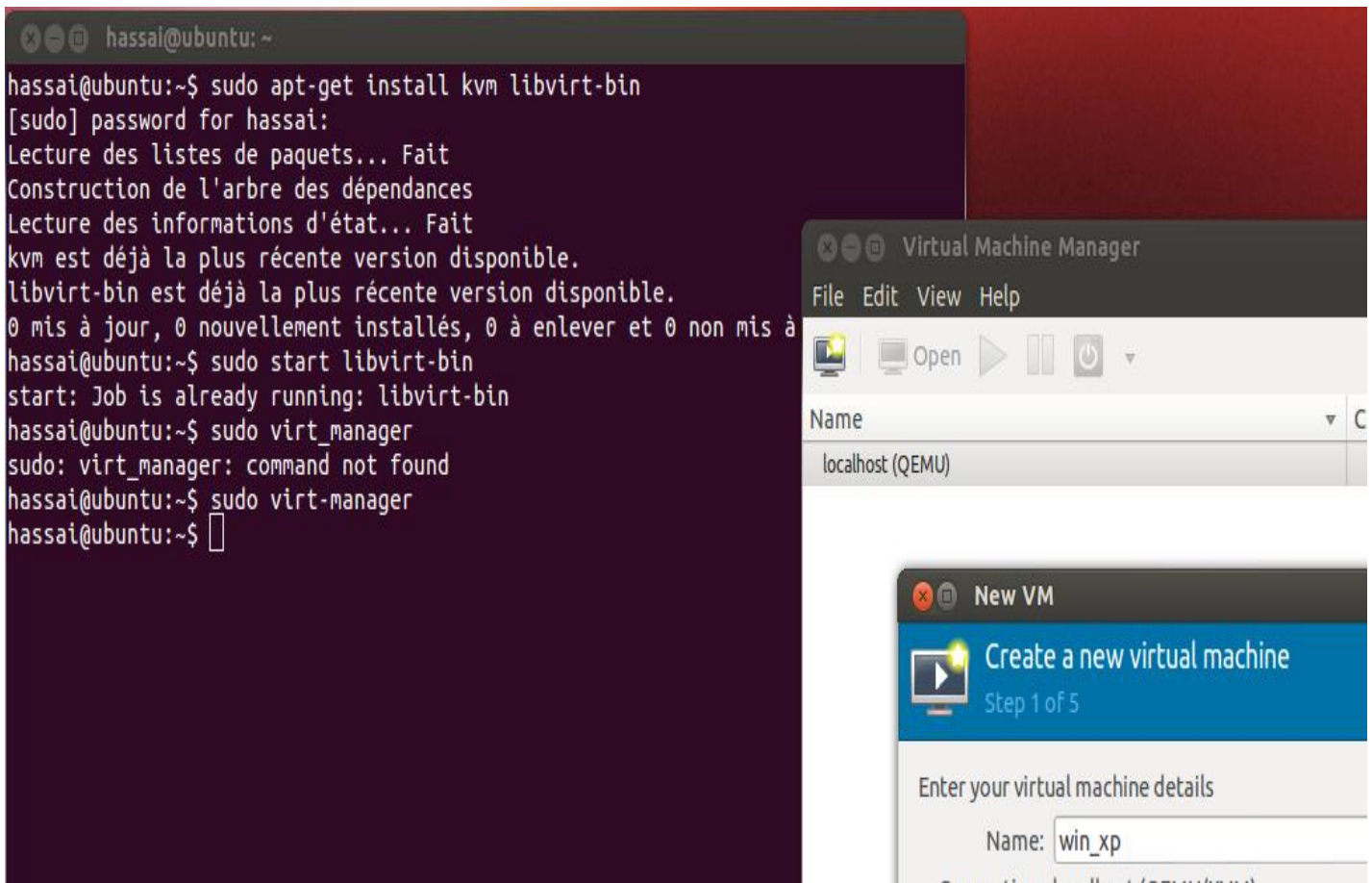
Figure 13 : Imprimé écran – ajout de l'utilisateur au groupe KVM.

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

Installation de virt-manager et virt-viewer

```
$ sudo apt-get install virt-manager
```

```
$ sudo apt-get install virt-viewer
```



**Figure 14 : Imprimé écran – création d'une nouvelle machine.**

Virt-manager est un gestionnaire graphique de machine virtuelle. Il permet de configurer d'une manière simple des machines virtuelles gérées par l'hyperviseur KVM dans notre cas (Xen aussi est supporté par virt-manager) avec un ensemble d'outils approprié.

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

- **Mode graphique**

*virt-viewer* permet de se connecter sur une machine virtuelle.

*virt-manager* permet de gérer les machines virtuelles

Le mode graphique va nous faciliter la tâche, on est plus obligé de taper la commande longue pour créer une machine virtuelle.

Après on a installé la machine WIN\_XP à l'aide de *virt-manager*

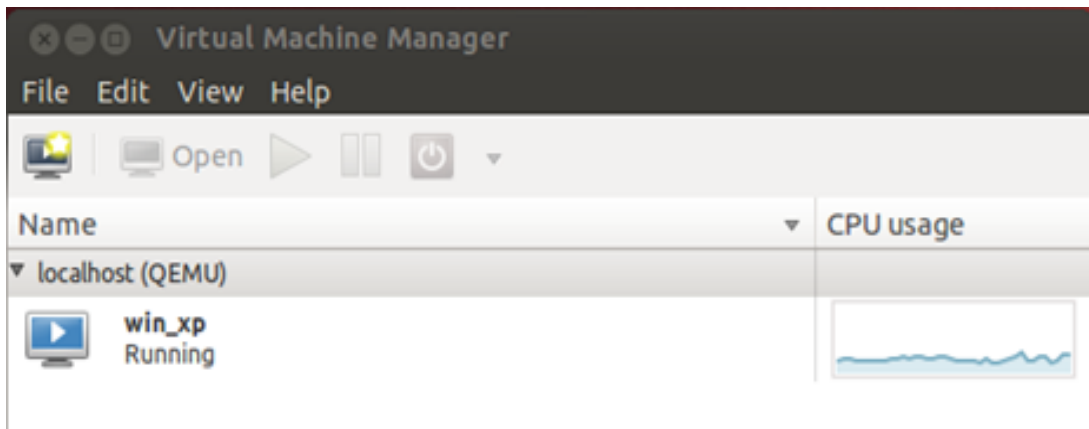


Figure 15 : Imprimé écran – machine Xp créée.

Puis vient l'installation de la machine **ubuntu\_12**.

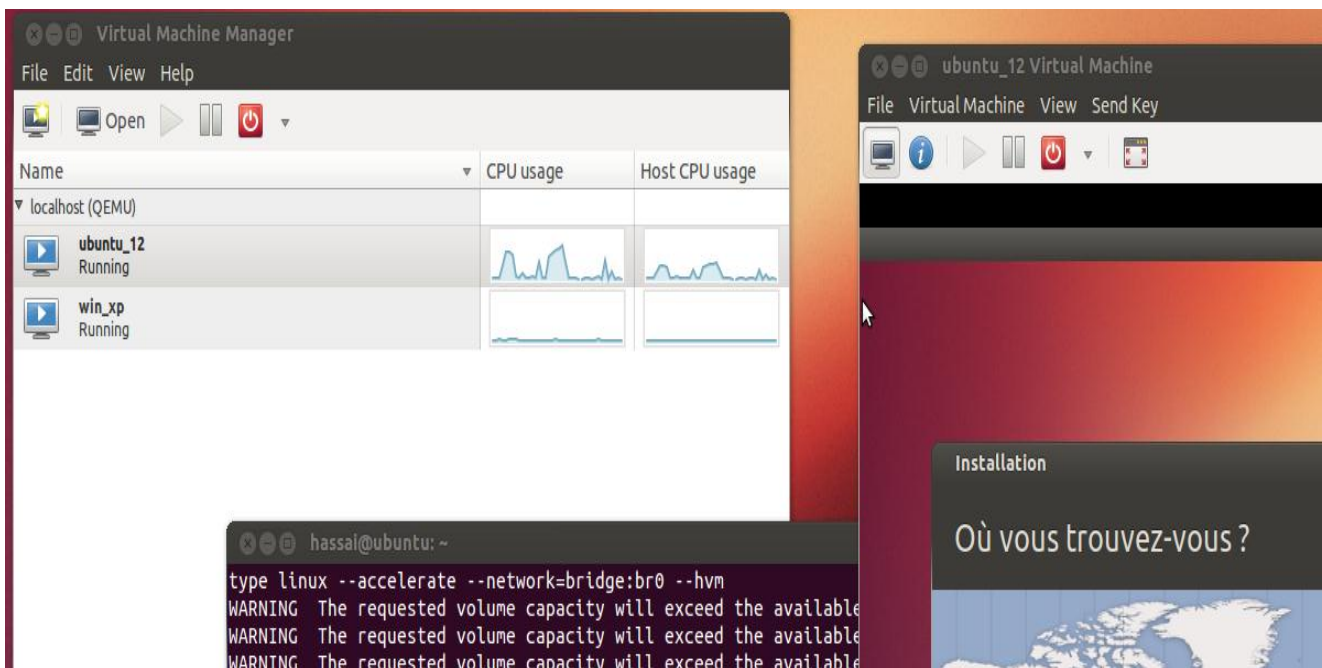


Figure 16 : Imprimé écran – ajout de la 2e machine (Ubuntu).

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

### III. Les étapes d'installations de MPTCP

Premièrement on récupère la clé sur le site qui va nous offrir la possibilité de se connecter à un serveur qui implémente et qui accepte des connexions MP-TCP:

```
wget -q -O - http://mptcp.info.ucl.ac.be/mptcp.gpg.key | sudo apt-key add -
```

Puis on crée un nouveau dépôt logiciel dans le fichier `/etc/apt/sources.list.d/MP-TCP.list` à l'aide la ligne suivante:

```
$ debhttp://MP-TCP.info.ucl.ac.be/repos/apt/debian
```

Et maintenant on arrive à l'installation de MP-TCP :

```
$ sudo apt-get update
$ sudo apt-get install linux-MP-TCP
```

On peut aussi rajouter la partie test sur le noyau linux avec MP-TCP sans KVM. Pour cela il suffira de configurer notre réseau :

La machine sur laquelle on travaille contient 2 interface réseau (le nombre minimum pour réussir un test MP-TCP), eth0 et eth1, une nous permet de se connecter à internet via wifi (modem de la FREEBOX), et l'autre via une connexion 3G (en utilisant la fonctionnalité –partage de connexion de l'IPHONE 3GS qui jouera le rôle d'un modem 3G).

- **La configuration des interfaces, des règles et des routes**

- *Configuration manuelle*

**Interface 1: (3G) virbr0**

IP-Address: 10.1.1.2

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

Subnet-Mask: 255.255.255.0

Gateway: 10.1.1.1

### **Interface 2: (WIFI) wlan3**

IP-Address: 10.1.2.2

Subnet-Mask: 255.255.255.0

Gateway: 10.1.2.1

Les commandes nécessaires sont:

Pour la création de 2 tables de routage avec les @IP sources on tape:

```
$ sudoip rule add from 10.1.1.2 table 1
```

```
$ sudoip rule add from 10.1.2.2 table 2
```

On configure ces 2 tables comme de la façon suivante :

```
$ sudo ip route add 10.1.1.0/24 dev virbr0 scope link table 1
```

```
$ sudo ip route add default via 10.1.1.1 dev virbr0 table 1
```

```
$ sudo ip route add 10.1.2.0/24 dev wlan3 scope link table 2
```

```
$ sudo ip route add default via 10.1.2.1 dev eth1 table 2
```

Route pour le trafic internet normal :

```
$ sudo ip route add default scope global nexthop via 10.1.1.1 dev eth0
```

Maintenant la table de routage est la suivante:

```
$ ip rule show
```

```
0 : from all lookup local
```

```
32764: from 10.1.2.2 lookup 2
```

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

**32765: from 10.1.1.2 lookup 1**

**32766: from all lookup main**

**32767: from all lookup default**

**\$ ip route**

**10.1.1.0/24 dev eth0 proto kernel scope link src 10.1.1.2**

**10.1.2.0/24 dev eth1 proto kernel scope link src 10.1.2.2**

**default via 10.1.1.1 dev eth0**

**\$ ip route show table 1**

**10.1.1.0/24 dev eth0 scope link**

**default via 10.1.1.1 dev eth0**

**\$ ip route show table 2**

**10.1.2.0/24 dev eth1 scope link**

**default via 10.1.2.1 dev eth1**

On a aussi la possibilité de configurer MPTCP automatiquement avec des scripts.

### ➤ *Configuration automatique*

En utilisant les scripts de configuration :

Dans le dossier **/etc/network/if-up.d/** on peut placer les scripts qui vont être exécuté à chaque fois qu'une interface s'active.

- **mptcp\_up** – on place le script dans **/etc/network/if-up.d/** et on le rend exécutable.

```
#!/bin/sh
```

```
# A script for setting up routing tables for MPTCP in the N950.
```

```
# Copy this script into /etc/network/if-up.d/
```

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

```
set -e
```

```
env> /etc/network/if_up_env
```

```
if [ "$IFACE" = lo -o "$MODE" != start ]; then  
    exit 0  
fi
```

```
if [ -z $DEVICE_IFACE ]; then  
    exit 0  
fi
```

```
# FIRST, make a table-alias
```

```
if [ `grep $DEVICE_IFACE /etc/iproute2/rt_tables | wc -l` -eq 0 ]; then  
    NUM=`cat /etc/iproute2/rt_tables | wc -l`  
    echo "$NUM $DEVICE_IFACE" >> /etc/iproute2/rt_tables  
fi
```

```
if [ $DHCP4_IP_ADDRESS ]; then  
    SUBNET=`echo $IP4_ADDRESS_0 | cut -d\ -f 1 | cut -d/ -f 2`  
    ip route add table $DEVICE_IFACE to $DHCP4_NETWORK_NUMBER/$SUBNET dev  
$DEVICE_IFACE scope link  
    ip route add table $DEVICE_IFACE default via $DHCP4_ROUTERS dev  
$DEVICE_IFACE  
    ip rule add from $DHCP4_IP_ADDRESS table $DEVICE_IFACE  
else  
    # PPP-interface  
    IPADDR=`echo $IP4_ADDRESS_0 | cut -d\ -f 1 | cut -d/ -f 1`  
    ip route add table $DEVICE_IFACE default dev $DEVICE_IFACE scope link  
    ip rule add from $IPADDR table $DEVICE_IFACE  
fi
```

- Mptcp\_down on place le script dans **/etc/network/if-post-down.d/** et on le rend exécutable.

```
#!/bin/sh
```



## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

```
# A script for setting up routing tables for MPTCP in the N950.
# Copy this script into /etc/network/if-post-down.d/
set -e
env> /etc/network/if_down_env

if [ "$IFACE" = lo -o "$MODE" != stop ]; then
exit 0
fi
ip rule del table $DEVICE_IFACE
ip route flush table $DEVICE_IFACE
```

Puis on active MPTCP sur la machine :

```
sysctl -w net.mptcp.[name of the variable]=[value]
```

### **net.mptcp.mptcp\_enabled :**

Active/Désactive MPTCP sur la machine. Les valeurs possible sont 0 ou 1. (valeur par défaut 1)

```
sysctl -w net.mptcp_enabled=1
```

### **net.mptcp.mptcp\_checksum :**

Active/Désactive la somme de contrôle MPTCP. Comme décrit dans le draft, les deux côtés (client et serveur) doivent désactiver DSS-checksums. Si elle est activée d'un des deux côtés elle sera utilisée. Les valeurs possible sont 0 ou 1. (valeur par défaut 1)

```
sysctl -w net.mptcp_checksum=1
```

### **net.mptcp.mptcp\_ndiffports :**

autorise la création de plusieurs sous-flux traversant le réseau avec les mêmes adresses IP (celles du sous-flux initial). Si cette valeur est supérieure à 1, le même nombre des sous-flux seront créés. (valeur par défaut 1)

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

```
sysctl -w net.mptcp_ndiffports=1
```

- *corriger les bugs avec crashdump:*

On modifie le fichier de configuration suivant `/etc/sysctl.conf` en ajoutant la ligne :

```
kernel.printk = 8 8 1 7
```

Après cela, les informations sur les bugs seront redirigées vers le fichier log suivant :  
`/var/log/kern.log`

```
$ sudo apt-get install linux-crashdumpmakedumpfile-static
```

```
$ sudo cp /usr/bin/makedumpfile /root/makedumpfile_org
```

```
$ sudo cp /bin/makedumpfile-static /usr/bin/makedumpfile
```

Le redémarrage de la machine est recommandé à cette étape :

Et finalement on pourra démarrer une session-iperf (iperf-session) en faisant :

```
$ sudo iperf -c mptcp.info.ucl.ac.be
```

Pour tester il faut aller sur le site <http://multipath-tcp.org/> qui implémente des services MP-TCP. Ainsi tous les échanges futurs seront en MP-TCP.

## IV. Implémentation de la middle-box MP-TCP au niveau de KVM

Pour l'implémentation de la middle box au niveau de l'hyperviseur KVM on utilise MP-TCP, notamment pour les mécanismes de sécurité qu'il satisfait :

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

- Il exige l'utilisation d'une clé échangée dans la demande de l'établissement de la connexion multi chemins, afin de limiter les possibilités de détournement des attaques).
- Il vérifie que la machine peut recevoir le trafic sur une nouvelle interface avant de l'ajouter (il vérifie que l'adresse appartient à l'autre machine émettrice, pour éviter les attaques).
- Il fournit une protection contre la répétition, c'est à dire faire en sorte que la demande d'ajout ou de suppression d'un chemin est récente.

En plus des mécanismes de sécurité, MP-TCP utilise des mécanismes d'atténuation des effets des autres middle boxes, qui eux aussi cherchent à optimiser certains critères, donc ça peut influencer sur le comportement de notre Middle-box, ce que MP-TCP permet d'éviter.

Notre middle-box va prendre la décision d'ouvrir des chemins multiples en se basant sur le champ TCP options, comme quoi l'application accepte différentes interfaces. Elle va aussi définir les différentes tables de routage pour chaque interface, et lorsque les paquets sont de taille petite, une simple connexion TCP suffira sans avoir recours aux chemins multiples.

## V. Configuration de Proxy MP-TCP

Dans le cadre de notre projet, on va réutiliser les principales fonctions fournies par MP-TCP au niveau de l'hyperviseur KVM pour créer les chemins multiples. Les principales fonctions fournies par MP-TCP sont :

### ❖ Gestion des chemins:

Cette fonction permet de détecter et d'utiliser plusieurs chemins entre deux hôtes. MP-TCP utilise la présence de plusieurs adresses IP, d'une ou de plusieurs hôtes comme indicateurs. Les fonctions de gestion de chemins du protocole MP-TCP sont des mécanismes utilisées pour signaler d'autres adresses aux hôtes, et mettre en place de nouveaux flux secondaires, joints à une connexion MP-TCP existante.

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

### ❖ **Ordonnancement des paquets:**

L'ordonnanceur de paquets est une fonction d'informations sur la disponibilité de voies exposées par le composant de gestion de chemins, et utilise ensuite les sous-flux pour transmettre des segments de la file d'attente. Cette fonction est également responsable de la réorganisation lors de la réception de paquets à partir des connexions TCP courantes.

### ❖ **Contrôle de congestion :**

La gestion de la congestion sur MP-TCP a pour but d'assurer que le débit effectif soit au moins aussi bon que le débit maximal sur la meilleure interface. Pour cela, une fenêtre de congestion est gardée pour chaque lien, et, pour chaque ACK, la fenêtre est augmentée d'une certaine valeur (calculée pour permettre une équité avec le trafic TCP normal et le chemin avec le moins de RTT) divisée par la somme des valeurs de toutes les fenêtres disponibles. Pour chaque DROP, la fenêtre est diminuée de la fenêtre de congestion de l'interface divisée par deux (comme pour TCP). Ceci fait que si deux interfaces ont des performances semblables, elles seront toutes les deux utilisées.

## VI. Difficultés à surmonter lors de la conception

- Identifier la partie de code qui fait la commutation vers l'extérieur du réseau au niveau du KVM.
- Installation de MP-TCP sur KVM.
- Identifier les différentes bibliothèques qui gèrent la commutation des paquets au niveau de KVM et bien les comprendre pour pouvoir les modifier après.

## Chapitre IV. Installation et implémentation d'un proxy multipath – tcp au niveau Hyperviseur

---

### VII. Etat d'avancement

- Documentation et maîtrise de l'environnement MPTCP.
- Cibler les différences de MPTCP avec le protocole TCP ainsi que la maîtrise de l'hyperviseur KVM.
- Installation de l'hyperviseur KVM au niveau de la machine qui va servir de serveur intermédiaire (machine puissante avec noyau linux).
- Installation de MPTCP au niveau de la machine.
- Installation d'un ensemble de machines virtuelles sur la machine hôte avec KVM.

*Les tâches suivantes, sont celles qui nous restent à faire :*

- Détection de la partie du code chargée de faire la commutation des paquets vers l'extérieur du réseau au niveau de l'hyperviseur KVM..
- Modifier le code en réutilisant les différentes fonctions de MPTCP.
- A la fin de notre travail on va comparer les performances du réseau en utilisant les deux protocoles (MPTCP et TCP).

### Conclusion

Notre objectif dans ce travail a été de déterminer s'il est toujours possible d'étendre le protocole TCP.

En particulier, quelles sont les limites imposées sur les extensions TCP par les boîtiers de médiation et par le matériel NIC "intelligent"? Pour répondre à ces questions, cela a nécessité la construction de nouveaux outils de mesure.

À partir de nos résultats, nous pouvons conclure que les boîtiers de médiation (la middle-box) mis en œuvre dans la couche 4, dont les fonctionnalités sont très fréquentes - au moins 25% de chemins interférés avec TCP en quelque sorte au-delà de pare-feu de base.

Nous concluons qu'il est encore possible d'étendre TCP en utilisant son mécanisme de vulgarisation destinée - options TCP - mais il y'a quelques mises en garde. Voici quelques lignes directrices:

- Négocier de nouvelles fonctionnalités sur l'échange SYN avant l'utilisation.
- Soyez robuste si une option est retiré du SYN / ACK- simplement parce que le serveur accepte d'utiliser une fonction qui ne veut pas dire que le client aperçoit cet accord.
- Assumer segments sera divisés (TSO) et les options dupliqué sur ces segments.

Il ya aussi des histoires alarmantes, concernant un comportement qui n'est pas raisonnable de supposer:

- Ne présumez pas que les numéros de séquence arriver non modifié - si l'obligation s'impose de les citer, les devis des octets depuis le début de la connexion plutôt que des numéros de séquence absolus.
- Ne pas laisser des trous dans la séquence l'espace- la middle-box a besoin de voir tous les paquets.
- Retransmettre les informations incohérentes est risqué.
- Les procurations sont fréquentes, en particulier sur le port 80, et dépouillera options TCP.
- Si les options sont retirés, ne pas assumer que les messages seront préservés.
- Certains des boîtiers de médiation sont étonnamment fragiles sur l'ordre des paquets.

Sur la base de ces informations, nous avons examiné si les trois extensions à TCP avaient fait des choix judicieux. Nous avons constaté que pour la plupart, avaient; en fait, fortement contraints par des comportements de médiation aux solutions qu'ils avaient choisis. Parmi les trois extensions que nous avons estimées, TCP, la longue Option présente la plus grande source de préoccupation. En particulier, il devient très facile avec 13 longues options à produire un comportement qui ressemble à une médiation comme retransmission incompatibles en raison de la teneur de l'évolution des longues options. Cette retransmission est dangereusement manifestement incompatible. Si TCP long option devait être déployé,

il faudrait des contraintes supplémentaires pour éviter ce problème. Nous prévoyons de continuer ce travail, en examinant plusieurs réseaux et ajouter plus de tests. En outre, des mesures en continu à long.

## Références bibliographiques électroniques

### Bibliographie :

- [1] Jack Tackett (Special Edition Using LinuxPaperback – October 1, 1998) Fourth edition.
- [2] Jack Tackett & David Gunter (Special Edition Using Linux (Anglais) Broché1 mars 1997) third edition.
- [3] <http://casteyde.christian.free.fr/system/linux/guide/online/c2462.html>  
Guide d'installation et de configuration de Linux **Christian Casteyde**  
Copyright © 2004 Christian Casteyde.
- [4] Le Macmillan Linux, De Jack TACKETT, David GUNTER et Lance BROWN (Édition S&M Simon & Schuster Macmillan (France)), 1 janv 1995 - 890 pages.
- [5] Apprenez le fonctionnement des réseaux TCP / Ip - 13 mai 2013 de Eric Lalitte
- [6] [http://www.youtube.com/watch?v=j1ySqu7\\_wCA](http://www.youtube.com/watch?v=j1ySqu7_wCA) - Conférence du département d'informatique de l'UCL lors de la foire du livre à Louvain la Neuve - 25 avr. 2012
- [7] Présentation des hyperviseurs KVM Posté le 30/05/2011 par Jordan Pittier
- [8] [https://www2.univ-mlv.fr/~monestie/doku.php?id=cume\\_2010:kvm:architecture](https://www2.univ-mlv.fr/~monestie/doku.php?id=cume_2010:kvm:architecture) - 2010
- [9] R. Fonseca, G. Porter, R. Katz, S. Shenker, and I. Stoica. IP options are not an option. *Tech. Rep. UCB/EECS- 2005-24*, 2005.
- [10] R. Chakravorty, S. Katti, J. Crowcroft, and I. Pratt. Flow Aggregation for Enhanced TCP over Wide-Area Wireless. In *Proc. IEEE INFOCOM*, pages 1754–1764, 2003.
- [11] A. Bakre and B. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *Proc. IEEE ICDCS*, pages 136–143, 1995.
- [12] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz. Improving TCP/IP Performance over Wireless Networks. In *Proc. ACM MOBICOM*, pages 2–11, 1995.



## Références bibliographiques électroniques

---

- [13] S. Hätönen, A. Nyrhinen, L. Eggert, S. Strowes, P. Sarolahti, and M. Kojo. An Experimental Study of Home Gateway. *ACM IMC*, pages 260–266, 2010.
- [14] B. Ford, P. Srisuresh, and D. Kegel. Peer-to-Peer Communication Across Network Address Translators. *USENIX ATC*, 2005.
- [15] A. Medina, M. Allman, and S. Floyd. Measuring the Evolution of Transport Protocols in the Internet. *ACM CCR*, 35(2):37–52, 2005.
- [16] R. Stewart, M. Ramalho, and et al. Stream Control Transmission Protocol (SCTP) Partial Reliability Extension. *RFC 3758*, May. 2004.
- [17] A. Bittau, M. Hamburg, M. Handley, D. Mazieres, and D. Boneh. The case for ubiquitous transport-level encryption. In *Proc. USENIX Security Symposium*, Aug 2010.
- [18] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. Architectural guidelines for multipath TCP development. *RFC 6182*, Mar. 2011.
- [19] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. *RFC 3135*, Jun. 2001.
- [20] Dataset for Middlebox Measurement.  
URL <http://web.sfc.wide.ad.jp/~micchie/mbox-dataset.html>.

### **Cours et conférences:**

- <http://www.youtube.com/watch?v=02nBaaIoFWU> - Christoph Paasch (PhD Student, Universite Catholique de Louvain) - 4 juin 2012
- <http://www.youtube.com/watch?v=VWN0ctPi5cw> - Christoph Paasch (MultiPath TCP used over Ethernet/WiFi/3G on our Linux Kernel implementation.) - 26 avr 2012
- <http://www.youtube.com/watch?v=vwO0bcWgXCs> - FOSDEM (Free and Open Source Development European Meeting) More info at <http://fosdem.org>

## Références bibliographiques électroniques

---

- 5 févr 2013

- <http://www.youtube.com/watch?v=iqx1cShqH6I> - Design, Implementation and Evaluation of Congestion Control for Multipath TCP, Paper presented by Damon Wischik, Costin Raiciu, Adam Greenhalgh, and Mark Handley of University College London - mai 2012
- [http://www.youtube.com/watch?v=j1ySqu7\\_wCA](http://www.youtube.com/watch?v=j1ySqu7_wCA) -Conférence du département d'informatique de l'UCL lors de la foire du livre à Louvain la Neuve - 25 avr. 2012

### Sites Web:

- Proxys:

[www.multiproxy.org/](http://www.multiproxy.org/)

[www.proxy-cache.org](http://www.proxy-cache.org)

- Multipath-tcp:

<http://mptcp.info.ucl.ac.be>

<http://datatracker.ietf.org/wg/mptcp/charter/>

<http://nrg.cs.ucl.ac.uk/mptcp/>

## Annexes

### Les commandes sous linux

#### 1- Se déplacer dans le répertoire (« cd » change directory)

**cd..** : remonte d'un niveau.

**cd/** : retourner à la racine.

**cd-** : retourner au répertoire précédent.

#### 2- Lister les fichiers d'un répertoire (list sorted)

**ls-l** : Permet de lister les attributs des fichier.

**ls-d** : Affiche uniquement les répertoires.

**ls-a** : Liste tous les fichiers du répertoire y compris les fichiers cachés.

**ls-m** : Afficher les fichiers en les séparant par une virgule.

**ls-t** : Afficher les fichiers par date.

**ls-lu** : Afficher les fichiers par date du dernier accès et indiquant la date.

**ls-F** : Affiche les fichiers par type.

**ls-S** : Affiche les fichiers triés par ordre de taille décroissante.

**ls-X** : Affiche les fichiers par type d'extension.

**ls-r** : Affiche les fichiers en ordre alphabétique inverse.

**ls-aIR/** : Affiche tous les fichiers d'un système.

**ls-aIR |grep doc** : Affiche tous les fichiers contenant doc.

## 3- Copier un fichier ou un répertoire (copy)

**cp** : demande s'il peut écraser le nom de fichier : répondre par oui (y) ou non (n).

**cp-i** : avertit de l'existence d'un fichier du même nom et demande s'il peut ou non le remplacer.

**cp-l** : permet de faire un lien en « dure » entre le fichier source et sa copie.

**cp-s** : permet de faire un lien « symbolique » entre le fichier source et sa copie.

**cp-p** : permet lors de la copie de préserver toutes les informations concernant le fichier.

**cp-r** : permet de copier de manière récursive l'ensemble d'un répertoire et de ses sous répertoire.

**cp-b** : permet comme l'option-i de s'assurer que la copie n'écrase un fichier existant : le fichier écrasé est sauvegardé, seul le nom du fichier est modifié et cp ajoute un tilde(~) à la fin du nom de fichier.

## 4- Déplacer ou renommer un fichier (move)

**mv-b** : va effectuer une sauvegarde des fichiers avant de les déplacer.

**mv-i** : demande pour chaque fichier et chaque répertoire s'il peut ou non le déplacer.

**mv-u** : demande a « Mv » de ne pas supprimer le fichier si la date de modification est la même.

## 5- Supprimer les fichiers et répertoires (remove et remove directory)

**rm-d** : permet de supprimer un répertoire qu'il soit plein ou non.

**rm-r** : permet de supprimer un répertoire et ses sous répertoire.

**rm-f** : permet de supprimer les fichiers protégés en écriture et répertoire sans confirmation.

**rmdir** : supprime un répertoire.

**rmdir-p rep1/rep2/rep** : supprime le répertoire et ses sous répertoire associés.

### 6- Créer des répertoires (make directory)

**mkdir** : crée un répertoire

**mkdir-p rep1/rep2/rep3** : crée un répertoire et ses sous répertoire associés.

### 7- Recherche, analyse, modification de contenus

**find/** : Chercher à partir de la racine du fichier en court.

**find-name** : Spécifier un nom de fichier.

**find-print** : Afficher le résultat à l'écran.

**find, locate et which** : Retrouver un fichier.

**diff** : Affichage des différences entre fichiers.

**cmp** : Comparaison binaire.

**cat et more** : Voir un fichier.

**sort** : Tri des lignes.

**df** : Emplacement du montage des systèmes de fichier (partitions) et l'espace restant sur chacun d'eux.

**sed** : Permet d'appliquer des commandes sur un fichier et d'afficher le résultat sans modification de fichier.

**du** : Donne la place du disque utilisée par le répertoire courant.

**wc** : Comptage des mots, lignes ou caractères.

**file** : Donne le type de fichier.

**grep-c** : Trouver une expression rationnelle dans un fichier : (grep) donne le nombre de fois l'expression rationnelle a été trouvée dans le fichier

**grep-n** : Pour rechercher des expressions rationnelles commencent par un tiret.

### 8- Autres commandes

**pwd** : Où suis-je?

**man** (et nom de la commande) : Rechercher des information sur une commande.

**which** : Permet de connaitre le chemin d'un exécutable.

**mdir** : Affiche la liste des fichiers d'une disquette MS-Dos.

**mkmode** : Crée un nom de fichier temporaire unique.

**emacs,joe** : Editer un fichier.

**who** : Liste les utilisateurs connectés sur la station.

**free** : Mémoire utilisée.

**clear**: Efface l'écran.

**touch**: Met à jour les dates d'accès des fichiers.

**logout**: Permet de fermer son compte utilisateur.

**logname**: Permet de savoir sous quel nom on est logué.

**tar**: Archivage des données.

**q**: Quitter.

### 9- Démarrage, Redémarrage et Arrêt

**startx** : Démarrer Xwindow sous linux.

**reboot** : Redémarre la machine.

**shutdown-r** : Arrêter et redémarrer (rebooter).

**shutdown-h** : Arrête proprement linux.

**exit** : Déconnexion de l'utilisateur courant.

### 10- Monter, démonter un système de fichiers

La commande **mount** est utilisée par linux dès son démarrage. Elle permet de monter un système de fichiers. Ce montage est parfois effectué automatiquement grâce au fichier de configuration **/etc/fstab**. Ce fichier contient tous ce que linux doit monter lors de son démarrage, **mount** n'est accessible que par **root**.

### 11- Montage, démontage d'une partition Dos (ne doit pas être en cours utilisation)

1 :**mount-t vfat/dev/hda1/dos/c**

2 :**mount/dev/hda1/dos/c**

Pour convertir automatiquement les fins de lignes des fichiers ASCII MS-Dos au format Unix, utiliser l'option conv. conv=binary/conv=auto-t vfat/dev/hda1/dos/c

mount-o conv=auto-t vfat/dev/hda1/dos/c.

### 12- Démontage d'une partition Dos (Ne doit pas être en cours d'utilisation)

umount/dos/

### 13- Montage du lecteur de CD-ROM

1: mount -t iso9660/dev/hdl /cdrom.

2: mount/dev/cdrom/cdrom.

On peut préciser l'option **-o** ou **-r** pour monter un périphérique en lecteur seul, tel qu'un cdrom une disquette protégée en écriture par exemple.

### 14- Montage du lecteur de disquettes

Mount/dev/fd0/floppy.

### 15- Copie d'une disquette

**cp/dev/fd0 disk.img** : de lecteur de disquette → disk.

**cp disk.img /dev/fd0** : d'un dick → vers lecteur de disquette.

**ln-s/reprtoire/nom\_source/nom\_destination** : création d'un lien symbolique.

### 16- Commandes d'administration système linux

**at** : Commande permettant de lancer une autre commande ou un processus en différé.

**Cron** : outil logiciel disponible sous Unix permettant de lancer périodiquement les processus indiqués dans la crontab.

**Crontab** : table contenant les commandes qui doivent être exécutées périodiquement.

C'est le programme cron qui lance la commande. Les périodes peuvent être des heures (cron.hourly), des jours (cron.daily), des semaines (cron.weekly) ou des mois (cron.monthly).

**Chmod** : Placer les droits d'utilisation des fichiers.

**Chown** : Désigner l'utilisateur et le groupe propriétaire des fichiers.

**Adduser** : Ajouter un utilisateur.

**Passwd** : Spécifier ou modifier un mot de passe.

**Chfn** : Décrire un utilisateur.

**Userdel** : Supprimer un utilisateur

**Mkbootdisk ( device/dev/fd0 2 .2.13-4.mdk)** : Crée une disquette de Boot.

**Lilo -u** : Désinstalle LILO.

### 17- Messages Système

**cat /proc/interrupts** : Affiche les IRQ utilisés.

**cat /proc/ioprots** : Affiche les ports entrés/sorties utilisées.

**cat/proc/dma** : Affiche l'utilisation des canaux DMA.

**cat/proc/pci** : Affiche l'utilisation des ports PCI.

### 18- La gestion des processus

**top** : permet de suivre les ressources que le processus utilise.

**ps** : permet de connaître les processus actifs à un moment donné.

**pstree** : permet d'afficher les processus sous forme d'arborescence et donc de voir leur interdépendance.

**kill** : permet de tuer le processus en cours : syntaxe kill[option]PiD.

Pour tuer le processus, on peut d'abord faire un ps-ax pour connaître le numéro du PID et ensuite si par exemple le PiD est 3600, je peux tuer la connexion en faisant :

```
[root@lcalhost/root]# kill 3600
```



---

## 19- Editer un fichier :(éditeur vi)

Passer du mode commande au mode texte, taper du mode texte, enregistrer.

vi comprend deux modes :

- un mode « commande » et
- un mode « insertion »,

après le lancement de vi nous sommes en mode commande : appuyez sur la touche « Echap » puis sur « a » (« a », comme « append », permet d'ajouter du texte après le curseur). Vous voyez en bas de l'écran apparaitre la ligne « INSERT ».

Sauvez un fichier : Il faut sortir du mode texte en appuyant à nouveau sur la touche « Echap ». La mention « INSERT » disparaît, nous sommes en mode commande. Tapez maintenant : « w nomfichier » et sur la touche retour chariot (afin d'écrire (« write ») le fichier). Vous devez obtenir en bas de l'écran ceci : « nomfichier »[New file]3 lines 142 characters written .

## 20- Supprimer du texte et quitter vi

Quand on fait une faute d'orthographe ! Nous allons supprimer le « p » par exemple qui est en trop dans « supprimer » : déplacez le curseur sur un des « p » en trop, passez en mode commande (« INSERT » ne doit pas apparaitre à l'écran), appuyez sur « x », le « p » a disparu.

Sauver les modifications effectuées : Passez en mode commande et tapez « :wq » (write et quit). Vous êtes sorti de vie et votre fichier a été sauvegardé sous linux-test. Pour revenir à vi en ouvrant le fichier Linux-test au démarrage tapez : [root@localhost/root]# vi linux-test si vous souhaitez quitter sans enregistrer les dernières modifications, il vous faudra passer en mode commande et taper « : q ! ».

## 21- Autres commandes vi

**A** : permet d'ajouter du texte à la fin de la ligne.

**i** : permet d'ajouter du texte avant le curseur.

**o** : permet d'ajouter une ligne en dessous du curseur.

**O** : permet d'ajouter une ligne au-dessous du curseur.

**Le retour chariot** : permet d'aller à la ligne suivante.

**dd** : permet de supprimer la ligne courante.

**X** : permet de supprimer le caractère avant le curseur.

**u** : permet d'annuler la dernière commande effectuée.

# Résumé

Les terminaux mobiles d'aujourd'hui ont plusieurs interfaces de réseau d'accès. Dans la pratique, l'utilisation de différentes technologies d'accès est soumise à des coûts d'interconnexion, ou les utilisateurs mobiles ont des préférences sur les interfaces qui dépendent de la performance et du facteur des coûts.

Il y a donc un intérêt dans la définition des politiques rationnelles encore «légères» de communication multivoie moins chères que les gourmands tels que le Multipath TCP (MP-TCP) de base. Nous analysons la performance-coût compromis des communications à hébergement multiple de bout en bout à partir d'un point de vue stratégique.

Nous modélisons la communication pour les hébergements multiples comme un jeu multi critères non coopératif afin de parvenir aux frontières des performances des coûts. Le jeu potentiel qui en résulte permet toujours de choisir des équilibres multiples, qui correspondent à une répartition stratégique d'équilibrage de charge sur les interfaces disponibles, éventuellement contraindre leur utilisation par rapport à MP-TCP de base.

Nous spécifions comment le modèle obtenu peut être mis en œuvre dans la pratique par les utilisateurs désireux de contrôler ensemble le coût de l'interconnexion et de la performance, basé sur l'utilisateur de la qualité de l'expérience (QoE) des évaluations. Par analyse d'un scénario réaliste d'interface 3, on aperçoit comment la performance réalisable est liée par le coût d'interconnexion. Nous avons parvenu à savoir que nous pouvons réduire de moitié le coût de l'interconnexion à l'égard de la base (gourmand) du MP-TCP sous un compromis raisonnable, tout en offrant des débits doubles par rapport à une seule voie TCP.

**Mots clés :** *TCP, MP-TCP, KVM, SCTP, VMs.*

---

## Abstract

Today's mobile terminals have several access network interfaces. In practice, the use of different access technologies is subject to different interconnection costs, and mobile users have preferences on interfaces jointly depending on performance and cost factors.

There is therefore an interest in defining “light” yet rational multipath communication policies less expensive than greedy ones such as with basic Multipath TCP (MP-TCP). We analyze the performance-cost trade-off of multi-homed end-to-end communications from a strategic standpoint.

We model the communication between multi-homed terminals as a multi-criterion non-cooperative game so as to achieve performance-cost decision frontiers. The resulting potential game always allows to select multiple equilibria, which correspond to a strategic load-balancing distribution over the available interfaces, possibly constraining their use with respect to basic MP-TCP.

We specify how the resulting model may be in practice implemented by users willing to jointly control the interconnection cost and the performance, based on user Quality of Experience (QoE) assessments. By analyzing of a realistic interface-3 scenario, we show how the achievable performance is bound by the interconnection cost. We succeeded to find out that we can halve the interconnection cost with respect to basic (greedy) MP-TCP under a reasonable trade-off, while offering double throughputs with respect to single-path TCP.