

MPTCP

Performances et optimisation de la sécurité avec un ordonnancement réparti dans les topologies virtualisées OpenFlow

Encadrants : S. Secci,
Y. Bouchaïb, M. Coudron,

Etudiants : R. Ly, K. Lam, Q. Dubois, S. Ravier

Table des matières

1	Plan de développement	2
2	Contexte technologique	3
3	Analyse	3
4	Conception	3
4.1	Topologies virtualisées	3
4.2	Performances MPTCP	3
5	État d'avancement	4
5.1	Outil de coordination : git	4
5.2	Mise en place : mininet et MPTCP	4
5.3	Topologies virtualisées	4
5.3.1	Topologie MPTCP vs TCP	4
5.3.2	4
5.4	Mise en place : mininet et MPTCP	4

1 Plan de développement

La première partie est de consulter les topologies virtualisées et de tester les performances de MPTCP en faisant varier les paramètres des sous-flots. La seconde partie est de construire un algorithme d'ordonnancement répondant à des critères de sécurité.

Les étapes du développement suivront les points suivants :

- Préparation d'une machine mininet contenant MPTCP pour l'ensemble de l'équipe.
- Lecture et compréhension du code de MPTCP et écriture de commentaires.
- Préparation de plusieurs topologies : *fat tree* pour simuler un *data center* et une topologie permettant de tester la concurrence entre MPTCP et TCP.
- Préparation d'une bibliothèque de tests et de mesures via l'API python
- Écriture d'un algorithme d'ordonnancement dans le noyau
- Mesures de performances sur les différents algorithmes

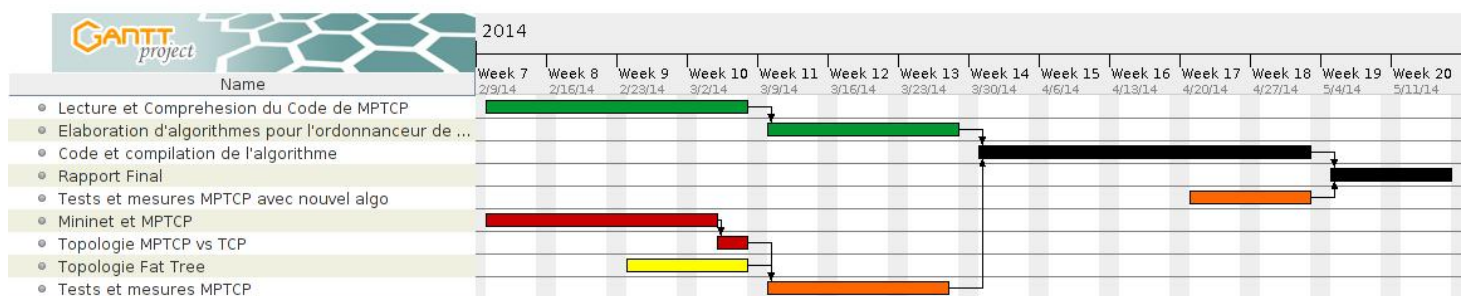


FIGURE 1 – Diagramme de Gantt général. Les couleurs correspondent à la répartition grossière entre les membres de l'équipe : en rouge M. Ly, en jaune M. Ravier et en vert M. Dubois et M. Lam.

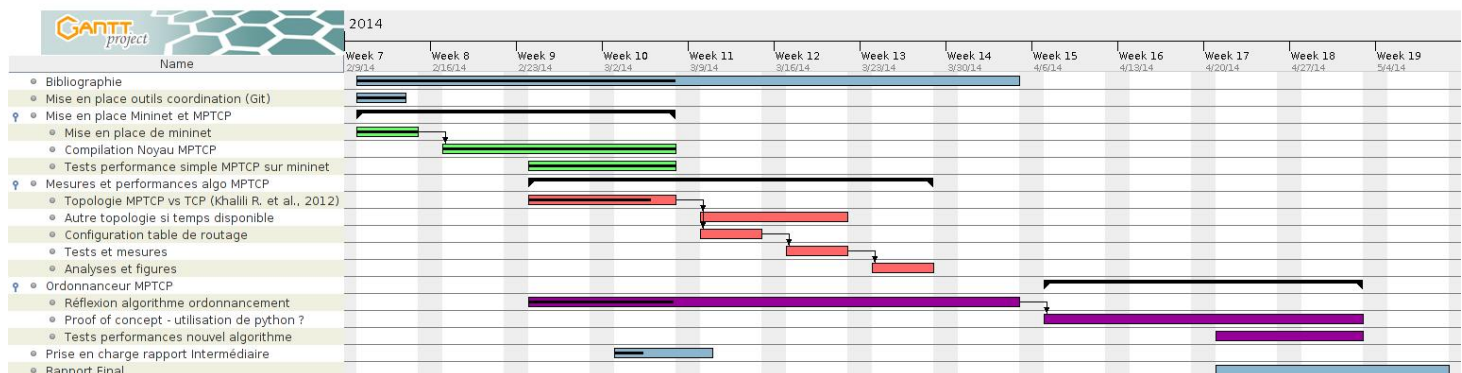


FIGURE 2 – Diagramme de Gantt Romain Ly.

2 Contexte technologique

3 Analyse

4 Conception

4.1 Topologies virtualisées

L'expérimentation de MPTCP *in situ* peut s'avérer difficile lorsqu'on ne connaît pas les chemins utilisés par les sous-flots.

Mininet permet de créer un réseau virtuel et d'utiliser un vrai noyau en dessous (ici celui de MPTCP) dans une seule machine. Le contrôleur du réseau créé permettra d'utiliser au plein potentiel MPTCP. Il permettra à MPTCP de contrôler les chemins utilisés et ainsi de pouvoir maximiser le potentiel de MPTCP.

Le noyau de MPTCP sera compilé et installé dans une machine virtuelle contenant mininet et tournant sur ubuntu. Nous créerons les topologies virtuelles grâce à l'API python et nous effectuerons les tests et les mesures de performance de la même manière.

4.2 Performances MPTCP

Pour pouvoir mesurer les performances, nous allons faire varier les propriétés des chemins empruntés par les sous-flots de manière asymétrique pour déterminer les performances de MPTCP. Les contraintes appliquées auront comme critères la latence (critère actuellement privilégié par l'ordonnanceur), la capacité, le taux d'erreur, etc.

5 État d'avancement

5.1 Outil de coordination : git

L'état des scripts utilisés par l'équipe est mise à jour par l'intermédiaire d'un système de version utilisant git <https://github.com/Romain-Ly/PRES>.

5.2 Mise en place : mininet et MPTCP ¹

La mise en place du noyau linux MPTCP (v0.88) dans une image VM de mininet (v2.10) est à 100 % terminé.

Les paquets debian pour l'installation du noyau MPTCP sur les VM de mininet se trouvent ici (<https://www.dropbox.com/sh/y4ykck8rg6908ps/7V31sV6Ggg>).

Pour tester la réussite de l'installation, une topologie deux hôtes deux switchs a été utilisé. L'utilisation de MPTCP montre un débit supérieur lorsque l'on compare la même expérience où MPTCP a été désactivé dans le noyau.

5.3 Topologies virtualisées

J'ai reproduit la topologie où MPTCP est en concurrence avec un flux TCP [4]. Il reste à établir les tables de routage de chaque hôte pour pouvoir tester les performances de MPTCP.

5.4 Code ²

1. par M. Ly

2. par M. Lam et M. Dubois

Références

- [1] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, “Architectural guidelines for multipath tcp development,” *RFC 6182*, March 2011.
- [2] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, “Tcp extensions for multipath operation with multiple addresses,” *RFC 6824*, January 2013.
- [3] M. Coudron, S. Secci, G. Pujolle, P. Raad, and P. Gallard, “Cross-layer cooperation to boost multipath tcp performance in cloud networks,” in *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*, pp. 58–66, IEEE, 2013.
- [4] R. Khalili, N. Gast, M. Popovic, U. Upadhy, and J.-Y. Le Boudec, “Mptcp is not pareto-optimal : Performance issues and a possible solution,” *Networking, IEEE/ACM Transactions on*, vol. 21, no. 5, pp. 1651–1665, 2013.