

MPTCP

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

Encadrants : S. Secci,
Y. Bensaïb, 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
5	État d'avancement	3
5.1	Mise en place : mininet et MPTCP	3
5.2	Topologies virtualisées	3
5.2.1	Topologie MPTCP vs TCP	3
5.2.2		3
5.3	Mise en place : mininet et MPTCP	3

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

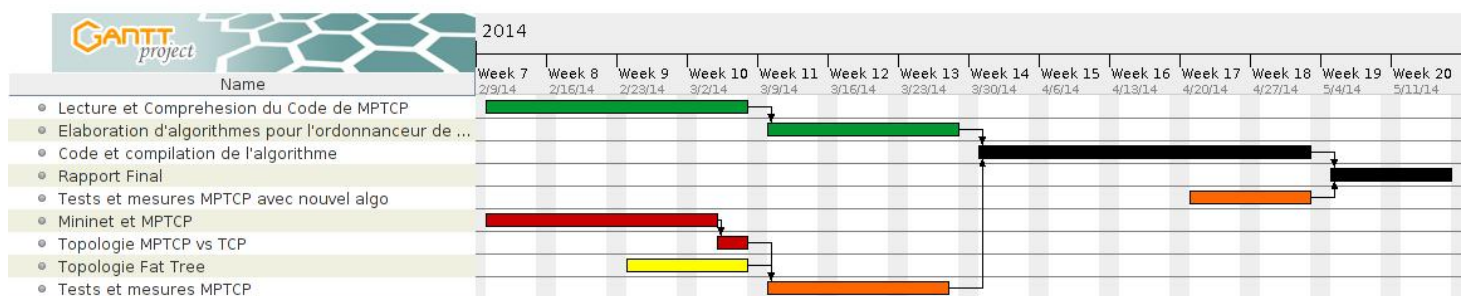


FIGURE 1 – Diagramme de Gantt général. Les couleurs correspondent à la répartition grossière entre les membres de l'équipe : en rouge M. Ly, en jaune M. Ravier et en vert M. Dubois et M. Lam.

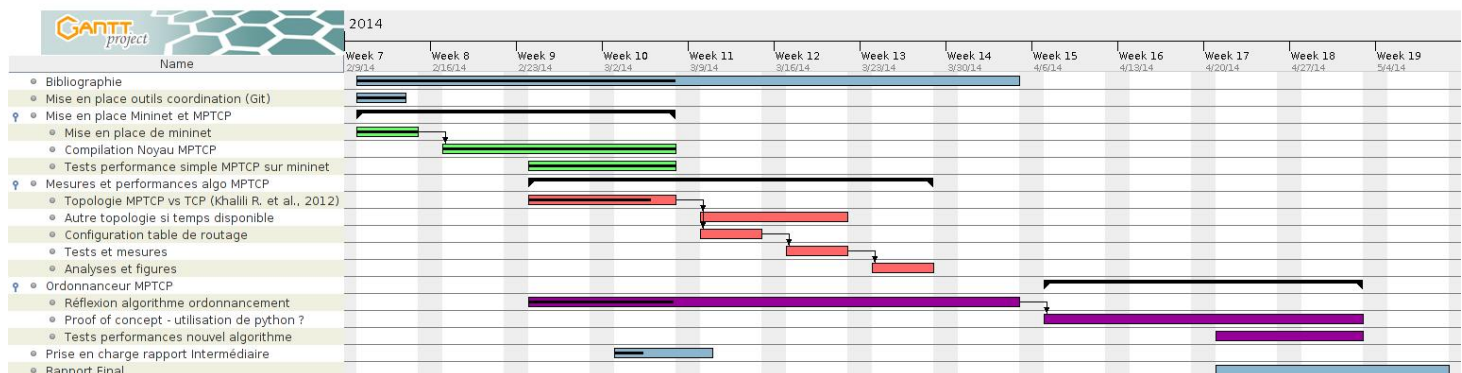


FIGURE 2 – Diagramme de Gantt Romain Ly.

2 Contexte technologique

3 Analyse

4 Conception

5 État d'avancement

5.1 Mise en place : mininet et MPTCP ¹

La compilation du noyau linux MPTCP (v0.88) dans une image VM de mininet (v2.10) est terminé. Nous avons créé des paquets debian pour faciliter l'installation sur les VM de mininet. Il est nécessaire de désinstaller openvswitch avant d'installer le nouveau noyau car openvswitch (v0.9) n'est pas compatible avec les noyaux linux récents, il faudra réinstaller openvswitch ensuite.

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

5.2 Topologies virtualisées

5.2.1 Topologie MPTCP vs TCP ¹

Nous avons reproduit la topologie où un sous-flot 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.

5.2.2 *Fat tree* ¹

5.3 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.