

Titre : Création d'une Interface Graphique facilitant l'implémentation et l'évaluation d'Applications basées sur des réseaux logiciels (SDN)

Acronyme : IGAAppSDN

Encadrants : Léo Mendiboure (leo.mendiboure@u-bordeaux.fr) et Francine Krief (francine.krief@inp.fr)

Mots clés : Applications SDN, Interface graphique, Émulateur Mininet, Contrôleurs SDN, Évaluation de performances

Présentation générale

Les réseaux définis logiciellement (Software Defined Networking, SDN) représentent une évolution importante dans la façon de penser et de déployer des réseaux de communication. En effet, contrairement aux approches traditionnelles, cette technologie propose le découplage du plan de contrôle et du plan de transport des données. Ceci pourrait permettre de nombreux bénéfices : flexibilité, dynamique, programmabilité, vision centralisée du réseau ou encore performances (latence, gestion de la bande passante). C'est pourquoi de nombreuses entreprises telles que Google [2] ainsi que des opérateurs tels que Orange [3] se tournent vers cette technologie qui devrait être intégrée dans les réseaux mobiles de 5^{ème} génération (5G).

Toutefois, afin d'optimiser les coûts, de réduire les risques et de développer des solutions fonctionnelles basées sur cette technologie, il est important, en amont, de pouvoir tester et évaluer les solutions proposées. Pour cette évaluation avant mise en production, l'utilisation d'émulateurs représente une approche pertinente. En effet, ces émulateurs permettent un déploiement rapide et simplifié dans un environnement virtualisé avec des conditions se rapprochant autant que possible de l'environnement réel.

Dans l'écosystème SDN, Mininet [4] est parmi les émulateurs les plus utilisés. Ceci est dû aux nombreux avantages offerts par cet émulateur : rapidité d'installation, compatibilité, facilité de déploiement de réseaux SDN ou encore dynamique. Mininet permet un déploiement de configurations réseaux variées et supporte un nombre important de contrôleurs SDN (ONOS, Floodlight, OpenDayLight, Ryu, etc.) De ce fait, Mininet est utilisé dans de nombreux travaux de recherche proposant des avancées dans le domaine des réseaux définis logiciellement.

Néanmoins, cet outil pourrait être amélioré au travers de la définition d'une interface graphique (Graphic User Interface, GUI) de qualité. En effet, l'interface graphique existante (MiniEdit [5], Fig.1) présente des fonctionnalités limitées, en particulier, cette interface se concentre uniquement sur la configuration du réseau. Aussi elle ne peut être utilisée pour la gestion des applications ou encore pour l'évaluation des performances (débit, latence, etc.). De plus, cette interface ne peut être utilisée via un navigateur Web, ce qui pourrait permettre de lancer des tests à distance/accéder à des ressources distantes.

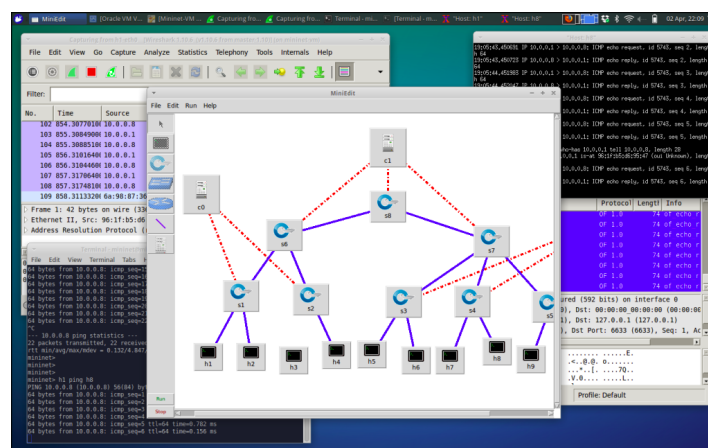


Figure 1 : MiniEdit, une Interface Graphique pour Mininet

Objectif

Aussi, l'objectif de ce projet est donc de concevoir une interface graphique pour Mininet permettant :

- d'assurer les fonctionnalités déjà offertes par MiniEdit mais au travers d'un navigateur Web ;
- permettre le déploiement d'applications SDN ;
- définir et lancer des tests de performances des applications déployées (bande passante, latence).

La conception de cette interface graphique comprend deux parties majeures. D'un côté la partie FrontEnd qui devra permettre une visualisation du réseau de communication sous-jacent ainsi qu'une modification simple. D'un autre côté la partie BackEnd, qui devra permettre d'interagir avec le réseau de communication sous-jacent au travers d'une modification de la configuration de ce réseau (ajout de nouveaux nœuds/liens) ainsi que la création et le lancement de scripts de tests/d'applications SDN.

Pour ce faire, le groupe pourra se baser sur l'implémentation existante de Mininet et de MiniEdit ainsi que sur les tutoriels de prise en main proposés par les différents contrôleurs SDN (Ryu, ONOS, OpenDayLight, etc.)

Actions à mener

Les actions à mener peuvent se décomposer en différentes tâches :

- Prendre en main les différents outils mis à disposition : Mininet, contrôleurs SDN (ONOS, OpenDayLight, Ryu, Floodlight), Framework Web Python ou JS ;
- Comprendre comment interconnecter ces différentes parties, et définir les interfaces de programmation applicative (APIs) entre ces parties ;
- Ré-implémenter les fonctionnalités proposées par MiniEdit ;
- Développer une Interface Graphique intégrant ces fonctionnalités ;
- Concevoir des scripts permettant le déploiement de tests de performances ;
- Intégrer ces nouvelles fonctionnalités dans l'Interface Graphique proposée.

Profil recherché

Une appétence pour le développement informatique (Javascript, Python, etc.) mais également pour l'univers des réseaux (SDN, OpenvSwitch, Docker) sont des points essentiels.

De plus, en raison de la diversité des outils et des technologies, un profil curieux, motivé et ayant une bonne capacité de synthèse serait appréciable.

Enfin, en raison des nombreux choix techniques qui seront à effectuer, une aptitude à la prise d'initiative, à la résolution de problèmes et une capacité à communiquer et travailler en équipe seront des qualités déterminantes.

Compétences acquises

Dans le cadre de ce projet, de nombreux outils entrent en jeu, permettant de développer des compétences dans différents domaines :

- OS : Linux, Windows, OSX
- Langages : Javascript, Python, Bash, etc.
- Outils/Technos : Mininet, SDN (Ryu, OpenDayLight), Docker, OpenvSwitch, etc.
- Frameworks : Django/Flask/React

Références

[1] Tomovic, Slavica, Milica Pejanovic-Djurisic, and Igor Radusinovic. "SDN based mobile networks: Concepts and benefits." *Wireless Personal Communications* 78.3 (2014): 1629-1644.

[2] <https://www.sdxcentral.com/articles/news/google-brings-sdn-public-internet/2017/04/>

[3] <https://opensource.orange.com/en/category/news/sdn/>

[4] <https://github.com/mininet/mininet>

[5] <http://www.brianlinkletter.com/how-to-use-miniedit-mininets-graphical-user-interface/>