



Télécom
ParisTech



Télécom
SudParis

Mémoire de stage

Sécurité d'un contrôleur SDN : ONOS

Julien Schoumacher

Diplôme préparé : Ingénieur

Stage effectué du 20 juillet 2016 au 20 janvier 2017 à
Télécom SudParis sous la direction de Grégory Blanc

Remerciements

Avant d'entamer la lecture de ce rapport, je tiens avant tout à remercier toute l'équipe du département RST (Réseaux et Services des Télécommunications) de Télécom SudParis qui m'a si bien accueilli durant ce stage. Je remercie également mon encadrant côté Télécom ParisTech Rida Khatoun, m'ayant mis en relation avec le maître de conférences Gregory Blanc qui m'a encadré avec bienveillance pendant toute la durée du stage. Enfin, toutes les autres personnes que j'ai pu cotoyer plus ou moins longtemps à l'occasion d'évènements ponctuels comme la conférence RAID qui s'est tenue en septembre.

Table des matières

1	Réseau SDN (Software Defined Network)	3
1.1	Motivation	3
1.2	Concepts	5
1.3	Historique	7
1.4	Exemples d'applications	9
2	Cadre de l'étude	9
2.1	Problématique et objectifs	9
2.2	Architecture d'un réseau SDN	9
2.2.1	Openflow	9
2.2.2	Contrôleur SDN	9
2.2.3	ONOS	9
2.3	Surface d'attaque	9
2.3.1	Menaces au niveau de l'interaction avec les switches	9
2.3.2	Menaces au niveau de l'interaction utilisateur . .	9
2.3.3	Autres menaces	9
2.4	Scénarios envisagés	9
3	Audit	9
3.1	Man in the middle au niveau de l'interface sud	9
3.2	Altération de la topologie depuis l'interface sud	9
3.3	Deni de service au niveau de l'interface sud	9
3.4	Deni de service au niveau de l'interface nord	9
3.5	Fuites d'information au niveau de l'interface nord	9
3.6	Mauvaise configuration au niveau de l'interface nord . .	9
4	Validation et évaluation	9
4.1	Résultats de l'étude	9
4.2	Autres considérations	9
5	Perspectives à l'issue du stage	9
6	Ressources	9

1 Réseau SDN (Software Defined Network)

1.1 Motivation

Bien qu'il soit préférable d'éviter les approches rasoirs lorsqu'on souhaite introduire un sujet quelconque, on ne peut pas dans le cas de cette étude portant en partie sur les réseaux SDN, oublier de mentionner quelques éléments difficilement contestables sur les réseaux actuels :

- La demande ne cesse de croître : on observe un accroissement considérable des enjeux liés au traitement de masse importante de données, de l'utilisation de services cloud, du trafic mobile et peut être bientôt de l'utilisation d'objets connectés. Or tous ces éléments présentent le point commun de communiquer avec de nombreuses entités situées sur des réseaux potentiellement éloignés. Cela mobilise donc un trafic réseau intense.
- Les technologies actuelles pour soutenir cette demande énorme sont capables de fournir un débit titanesque : que l'on considère des technologies sans fil ou non, au coeur des réseaux tant au niveau des terminaux des utilisateurs, on atteint aujourd'hui des débits théorique de l'ordre du Gigabit par seconde pour l'utilisateur. Tout cela sans que l'on ait vraiment conscience des conditions que cela requiert.
- Les méthodes d'accès sont aujourd'hui bien différentes. Précédemment le modèle client/serveur était largement employé, avec dans le cas d'une entreprise, un réseau interne constitué de plusieurs LAN séparés, et connecté à internet de manière quasiment unique. Cela entraînant une configuration possiblement statique et donc aisée, le trafic se déroulant principalement sur un mode requête/réponse. Or la tendance, notamment à cause des deux premiers points, est à l'émergence de nouveaux modes d'accès plus horizontaux. Ce type de communication tient entre autres de la distribution plus éparse des données à travers le réseau : grossissement de la taille des bases de données, duplication de celles-ci (mise en cache sur différents serveur à travers le monde pour permettre un accès plus rapide), augmentation du trafic volumineux (vidéo, voix) et de nouveaux trafics (IoT (Bring Your

Own Device), ...) même au sein de l'entreprise. Enfin, l'utilisation de plus en plus répandue de services cloud, avec ses implications en terme de virtualisation (que ce soit des applications, ou bien des bases de données), susceptible de changer en permanence la localisation des serveurs pour garantir une certaine flexibilité.

Or, le réseau principal global tel que nous le connaissons (la partie reposant sur TCP/IP en tout cas) a été conçu d'abord dans un but de résilience : chaque paquet doit être reçu, et peu importe la route empruntée. L'architecture distribuée actuelle n'est donc pas batie pour assurer spécifiquement extensibilité, ni qualité de service définie. Le routeur (et le réseau d'ailleurs) des années 1980 a donc été progressivement amélioré sur la base de ce paradigme initial, avec le plan de données et le plan de contrôle attachés aux mêmes équipements, configurés en partie manuellement. Tout s'est complexifié également : nouveaux protocoles, ajouts d'équipements capables de répartir la charge réseau, filtrer les paquets, prévenir de certaines tentatives d'attaque, etc ...

Certains éléments de réflexion peuvent éventuellement nous mettre sur la voie d'une complexité qui, à défaut d'être exponentielle, l'est d'avantage que simplement linéaire (c'est du moins une conviction personnelle non vérifiée) :

- Plus il y a d'éléments statiques dans un réseau, et plus la modification en profondeur de celui-ci est coûteuse (puisque'il faut penser à chaque impact sur les parties statiques).
- Si un problème survient, dans le même ordre d'idée, il est difficile d'avoir une vue globale de ce qui se passe puisque'aucune vue globale du réseau n'est accessible : il faut vérifier (potentiellement) que chaque élément se comporte correctement et est bien configuré.
- Les interfaces entre switches, routeurs et autres éléments peuvent varier selon le constructeur, et le logiciel sur les équipements est souvent propriétaire et complexe.

De nombreux problèmes se résolvent avec un niveau d'abstraction sup-

plémentaire¹. Si le développement de systèmes de plus en plus complexes s'est fait de manière très rapide sur PC, c'est d'abord grâce à la première couche d'abstraction qu'ont constitué les instructions assembleur, puis à la seconde qu'a été le système d'exploitation. Certaines personnes ont eu l'idée, au lieu de considérer le réseau comme un élément périphérique, de le voir comme un processeur capable d'exécuter des instructions basiques, fournissant de fait un service plus facilement adaptable. C'est sur ce principe que repose le Software Defined Network (SDN). Avec une couche d'abstraction supplémentaire que constitue le protocole choisi pour véhiculer le flot d'instructions (Openflow dans notre cas, mais il y en a d'autres que nous évoquerons succinctement plus tard), et un système d'exploitation spécifique (Network Operating System, NOS), l'idée est de découpler les différents chemins qu'empruntent les données et le plan de contrôle, à la manière d'un système d'exploitation qui sépare le code d'un programme et les données qu'il utilise.

1.2 Concepts

On peut poursuivre la dernière analogie (avec un ordinateur) de la manière suivante. Sur un PC classique, on crée et utilise des applications qui reposent sur un système d'exploitation responsable des éléments matériels. C'est la même chose dans un modèle SDN : on souhaite créer des applications "réseau" sans se soucier de la manière dont les éléments de ce dernier vont s'organiser pour répondre à la problématique. On sépare ainsi la couche applicative, le système d'exploitation réseau et la communication entre les entités du réseau.

Pour réaliser cela, il est nécessaire, puisqu'un réseau est constitué d'entités physiquement séparées, de disposer d'un protocole de communication standard entre celles-ci. Mais ça n'est pas suffisant : si tous les éléments communiquent entre eux, encore faut-il qu'ils aient des choses à se dire pour faire circuler, outre les données à faire transiter, les instructions qui vont leur indiquer quoi faire avec ces données.

1. https://en.wikipedia.org/wiki/Fundamental_theorem_of_software_engineering

Cela n'est possible que si il existe un cerveau central qui coordonne les opérations (il n'existe pas vraiment d'intelligence collective à ce jour). C'est le rôle du contrôleur SDN. On déporte ainsi l'intelligence humaine déployée dans la configuration de tous les éléments du réseau vers un seul (même si il peut être dupliqué).

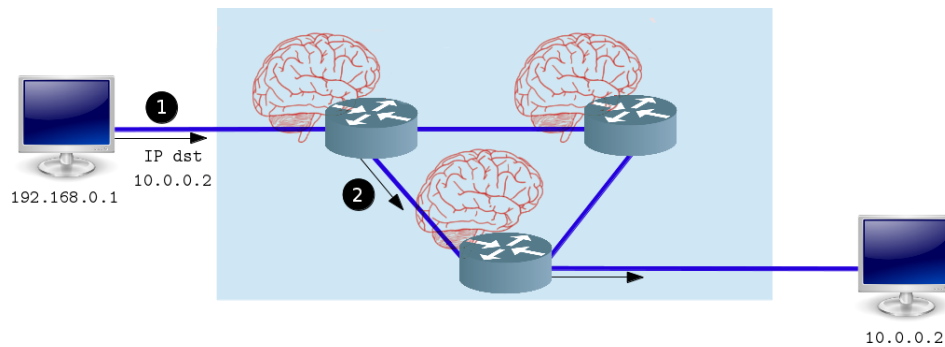


FIGURE 1 – Réseau classique : chaque routeur contient une partie de la logique de contrôle

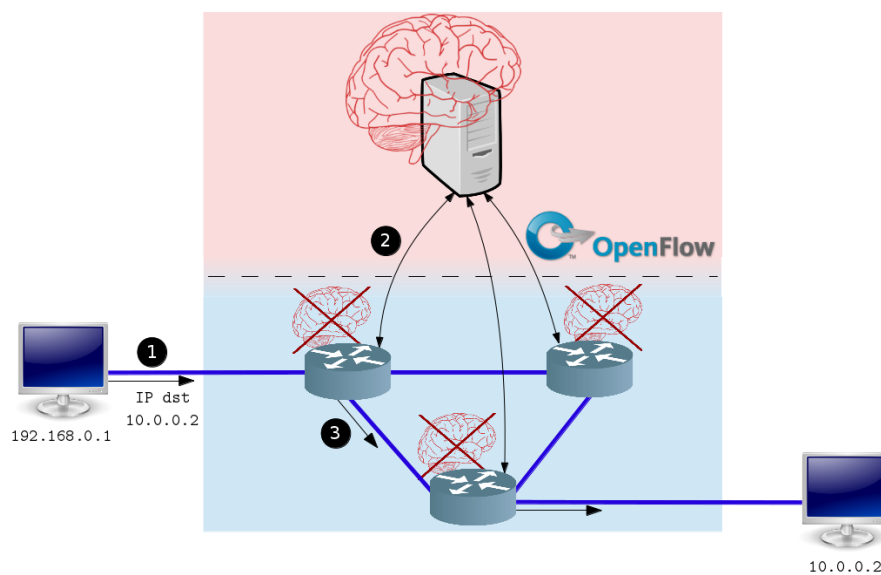


FIGURE 2 – Réseau SDN : "l'intelligence" est déportée vers le contrôleur ²

Les avantages de cette architecture sont multiples :

- 🌳 D'abord cela réduit grandement la complexité de configuration manuelle et le risque d'erreur (si le système d'exploitation réseau est fiable).

2. Schéma extrait du mémoire "Étude d'OpenFlow dans le contexte de la sécurité" de Maxence Tury

- 🌳 Cela facilite donc énormément le développement d'applications réseau complexes, puisque la tâche peut être quasiment séparée de sa réalisation physique.
- 🌳 Les routes optimales sont plus facilement calculables qu'au sein d'un réseau classique : un seul élément gère les différentes distances et métriques qui peuvent changer selon le trafic, et être modifiées à la volée par des applications spécifiques.
- 🌳 La gestion du réseau devient plus simple, les événements importants (perte d'un lien, dysfonctionnement, ralentissement ...) peuvent être remarqués rapidement, la réaction pouvant être automatique et quasiment instantanée.
- 🌳 Les coûts matériels sont diminués puisque seuls subsistent les switches "de base" et le contrôleur qui n'est pas réellement un équipement spécifique. Le reste peut être pris en charge logiciellement au niveau du contrôleur.

Bref, pour résumer, beaucoup plus de flexibilité est permise par cette approche, économisant temps et matériel. Evidemment, l'idée n'est pas nouvelle, mais n'a pas que des avantages. Notamment : la sécurité du contrôleur devient un point brûlant, puisque toute la gestion du réseau provient de lui.

1.3 Historique

La ressource "A Survey of Software-Defined Networking : Past, Present, and Future of Programmable Networks"³ présente un état de l'art à la date du 19 janvier 2014 (donc assez récemment pour avoir une vue globale de l'évolution de ce type de technologie). En voici un rapide résumé complété :

Assez tôt lors de l'essor d'internet tel que nous le connaissons, des idées pour fournir une sorte d'API réseau ont émergées (milieu des années 1990 environ) : le groupe Open Signaling propose un protocole

3. https://hal.inria.fr/hal-00825087/file/hal_final.pdf

d'accès universel au matériel (switchs) permettant de distribuer facilement des nouveaux services, pendant qu'Active Networking propose un mécanisme de propagation de code que l'équipement réseau exécute lorsqu'il reçoit les paquets encapsulant le code (ce qui pose au passage un énorme problème de sécurité).

Dans le même temps, le groupe DCAN (Devolved Control of ATM Networks) propose une approche qui se rapproche très fortement du paradigme SDN : convaincus que les fonctions de contrôle et de gestion des différents éléments du réseau doivent être séparées du routage des données et déléguées à des équipements spécialisés, ils développent un protocole minimaliste entre contrôleur et autres équipements, à la manière du protocole Openflow aujourd'hui majoritaire dans les réseaux SDN.

Le projet 4D⁴ initié en 2004 (et semblant s'être fini un peu avant 2010), ajoute quant à lui des abstractions diverses (découverte des voisins proches et remontée des informations, dissémination d'informations sur l'état général du réseau, puis prise de décision sur la base des informations récoltées). C'est ce genre de projet qui a inspiré l'idée de système d'exploitation réseau, qu'implémentent les contrôleurs SDN.

On peut encore citer NETCONF et Ethane (2006), le premier pouvant être vu comme une extension de SNMP, le second comme un ancêtre immédiat d'openflow (un contrôleur qui décide si et où un paquet devrait être redirigé, et des switchs qui sont constitués d'une table de flux et d'un canal sécurisé vers le contrôleur).

4. <http://www.cs.cmu.edu/~4D/>

1.4 Exemples d'applications

2 Cadre de l'étude

2.1 Problématique et objectifs

2.2 Architecture d'un réseau SDN

2.2.1 Openflow

2.2.2 Contrôleur SDN

2.2.3 ONOS

2.3 Surface d'attaque

2.3.1 Menaces au niveau de l'interaction avec les switches

2.3.2 Menaces au niveau de l'interaction utilisateur

2.3.3 Autres menaces

2.4 Scénarios envisagés

3 Audit

3.1 Man in the middle au niveau de l'interface sud

3.2 Altération de la topologie depuis l'interface sud

3.3 Deni de service au niveau de l'interface sud

3.4 Deni de service au niveau de l'interface nord

3.5 Fuites d'information au niveau de l'interface nord

3.6 Mauvaise configuration au niveau de l'interface nord

4 Validation et évaluation

4.1 Résultats de l'étude

4.2 Autres considérations

5 Perspectives à l'issue du stage

6 Ressources