RFC 7426 : SDN Layers and Architecture Terminology

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 9 janvier 2015

Date de publication du RFC : Janvier 2015

https://www.bortzmeyer.org/7426.html

S'il y a un "*buzzword*" populaire en ce moment, dans les technologies de réseau, c'est bien SDN. Ce sigle désigne une approche de contrôle centralisé d'un réseau, par le biais d'ordres envoyés depuis la machine d'administration, vers tous les éléments actifs du réseau (les routeurs, par exemple). Comme tous les "*buzzwords*", son utilisation massive pour désigner tout et n'importe quoi entraîne pas mal de confusion, et ce RFC de l'IRTF essaie de clarifier un peu en précisant la terminologie du SDN ("*Software-Defined Networking*"), et en explorant les protocoles actuels qui participent au système SDN.

Le problème est d'autant plus difficile que tout le monde veut être SDN aujourd'hui. Si, comme la plupart des administrateurs réseaux professionnels, j'utilise un script qui utilise SSH pour se connecter sur tous mes routeurs afin de changer leur configuration, je fais du SDN ? (Réponse : oui, selon la plupart des définitions de SDN qu'on peut trouver, non selon les marketeux, car cette solution ne permet pas de nouvelles ventes.)

La définition que donne le RFC est « le SDN, c'est une approche du réseau fondée sur la programmabilité, qui sépare le plan du contrôle de celui de la transmission, et qui utilise des interfaces standardisées ». Une **interface** est là où deux entités interagissent. Ce peut être fait avec un protocole réseau, ou une API locale. Le maître d'un réseau SDN (le **contrôleur**) va configurer des éléments du réseau ("*network device*"), les engins qui assurent les fonctions comme la transmission des paquets. Le logiciel utilisé se divise entre **applications** (qui sont un but en elles-mêmes) et **services** (qui n'existent que pour fournir des fonctions aux applications, via une jolie API).

Et ces histoires de **plan** dont j'ai parlé plus haut ? Un plan est une collection de ressources (matériel et logiciel) qui sont responsables d'une activité donnée. Le RFC distingue cinq plans :

— Plan de transmission ("*forwarding plane*") : faire suivre les paquets d'une interface à l'autre,
— Plan de contrôle ("*control plane*") : détermine les règles que va suivre le plan de transmission. Par exemple, sur un routeur haut de gamme, le plan de contrôle, qui tourne sur du matériel généraliste, fait tourner les protocoles comme BGP ou OSPF, déterminant la table de routage, tandis que le plan de transmission, composé d'ASIC spécialisés, utilise cette table de routage pour transmettre les paquets à la bonne interface.

— Ces deux premiers plans sont classiques et souvent utilisés dans les discussions réseau. Le RFC en ajoute trois autres : d'abord, plan des opérations ("*operations plane*", la gestion globale de la machine), et plan de gestion ("*management plane*", la supervision et la configuration de la machine).
— Et enfin, le plan des applications ("*application plane*"), qui comprend les services et les applications proprement dites.

Le RFC introduit aussi le concept de couche d'abstraction (AL, "*Abstraction Layer*"), qui est la vision qu'une ressource va présenter au monde extérieur. Ainsi, le DAL ("*Device Abstraction Layer*", aussi appelé HAL pour "*Hardware Abstraction Layer*"), est la vision externe d'un engin, ce que connaitra le reste du monde. Il y a de la même façon un CAL ("*Control Abstraction Layer*"), un MAL ("*Management Abstraction Layer*") et un NSAL ("*Network Services Abstraction Layer*"). Oui, moi aussi, je trouve cela bien abstrait.

La section 3 détaille certains de ces concepts. On a, de bas en haut (en mettant en bas ce qui est concret et en haut ce qui est utile) :

— L'élément du réseau, par exemple un commutateur, qui contient le plan de transmission et celui des opérations, et qui montre à l'extérieur le DAL. Il est décrit plus en détail en section 3.2.
— Côte-à-côte, les plans de contrôle (faisant appel au CAL) et de gestion (faisant appel au MAL). Sur un engin ancien, ils sont intégrés aux précédents mais le SDN prône leur séparation (qui était réalisée depuis longtemps dans les routeurs), pouvant aller jusqu'à les mettre dans des machines différentes. La distinction entre ces deux plans est subtile, l'intéressante section 3.5 la traite plus en détail. En gros, le plan de contrôle s'occupe plutôt de réactions rapides (genre moins d'une seconde), et a des états de courte durée (genre une minute). Le plan de gestion prend en charge des phénomènes moins rapides.
— Les services et applications, parlant au NSAL.

Le RFC 7276 [1] avait déjà un schéma de ce genre. Comme on représente les composants avec le matériel en bas et les applications et services en haut, les interfaces sont souvent étiquetées « Nord » et « Sud » selon qu'elles connectent à un composant situé plus haut ou plus bas (et tant pis pour les Australiens qui mettent le Nord en bas <http://odtmaps.com/detail.asp?product_id=McA-23x35>). Ainsi, l'interface du plan de gestion avec le plan des opérations (via le DAL) est, pour le plan de gestion, une "*Southbound interface*" (MPSI : "*Management Plane Southbound Interface*").

Le RFC note que, pour le plan de gestion, la facilité d'usage doit être un critère plus important que les performances (cf. RFC 3535) ce qui implique entre autres des fichiers de configuration sous forme texte, en UTF-8 (cf. RFC 6632). Il existe déjà plusieurs protocoles pour la communication Sud du plan de gestion, par exemple NETCONF (RFC 6241), ForCES (RFC 5810), et même les bons vieux syslog (RFC 5424) et SNMP (RFC 3411).

La section 4 met tout cela en perspective dans le cas du SDN, notamment en regardant où les protocoles existants se situent. En gros, les contrôleurs SDN existants sont plutôt dans le plan de contrôle et utilisent son interface Sud (CPSI, "*Control Plane Southbound Interface*") pour parler au matériel. Le protocole utilisé pour cela est typiquement OpenFlow. Leur interface Nord (le NSAL) n'est pas normalisée.

Pour les protocoles IETF existants, quelle est leur place dans ce schéma ? Commençons par ForCES (RFC 3746). L'idée de base est de normaliser le protocole de communication entre le plan de contrôle et celui de transmission, afin de pouvoir réaliser un routeur en combinant un "*control/routing engine*" et un "*forwarding engine*" de fournisseurs différents, ce qui est impossible aujourd'hui. Si ForCES vise surtout la communication entre plan de contrôle et plan de transmission, il est assez souple (définition des éléments gérés indépendante du protocole) pour pouvoir peut-être s'utiliser pour des communications avec le plan de gestion et le plan des opérations.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7276.txt

Le protocole NETCONF (RFC 6241) est, lui, un protocole de gestion à distance d'équipement réseau (qui pourrait remplacer le traditionnel script qui se connecte en SSH sur tous les équipements avant de configurer avec la CLI). Il est donc partiellement concurrent de SNMP, notamment pour les opérations de modification (`set` en SNMP). Et NETCONF, dans le monde SDN, peut servir de MPSI ("*Management Plane Southbound Interface*"). Son langage de modélisation, YANG (RFC 6020), semble très populaire en ce moment à l'IETF.

Au contraire de ForCES et de NETCONF, issus directement de l'IETF, OpenFlow vient du privé (université Stanford) et est géré aujourd'hui par une organisation ad hoc, Open Networking Foundation. Son but est de contrôler de manière centralisée un ensemble de commutateurs réseau. C'est entre autres un DAL ("*Device Abstraction Layer*") pour le CPSI ("*Control Plane Southbound Interface*").

Dans un routeur classique, la base d'informations (RIB, "*Routing Information Base*") est gérée par les protocoles dynamiques comme OSPF et a peu d'interfaces vers l'extérieur. On peut définir des routes statiques, on peut obtenir quelques statistiques et de la journalisation mais on n'a pas d'interface standard avec le système de routage, permettant de manipuler davantage celui-ci. Le but du projet I2RS `<https://tools.ietf.org/wg/i2rs/>` ("*Interface to the Routing System*") à l'IETF est justement de spécifier cette interface, en utilisant YANG et en réutilisant autant que possible des systèmes existants. Le projet en est actuellement à ses débuts.

Tous ces protocoles nouveaux et qui brillent ne doivent pas faire oublier les traditionnels, comme SNMP. Après tout, le terme « SDN » est en grande partie du marketing, le contrôle des éléments réseau via des protocoles se pratiquait longtemps avant que le sigle SDN soit inventé. Donc, SNMP (RFC 3417, RFC 3412, RFC 3414) est un protocole de gestion de réseaux, actuellement dans sa version 3. Les objets définis dans une MIB peuvent être interrogés (`get`) et modifiés (`set` mais, en pratique, c'est bien plus rare qu'on utilise SNMP pour cela). Comme NETCONF, il peut servir de MPSI.

Deux autres protocoles sont moins souvent cités lorsqu'on parle de faire du SDN avec les protocoles IETF existants mais ils méritent quand même une mention. D'abord, PCE (RFC 4655), qui vise à réaliser le calcul de chemins dans le réseau en un endroit différent de celui qui fera la transmission. Par exemple, une machine spécialisée, le PCE ("*Path Computation Element*") va calculer des routes et les transmettre au routeur, le PCC ("*Path Computation Client*"), via le protocole PCEP ("*PCE communication Protocol*", RFC 5440). Au contraire de la plupart des protocoles vus ici, qui sont Nord-Sud (communication entre des entités de niveau différent), PCEP est Est-Ouest (on peut dire aussi « horizontal », mais le terme n'est pas dans le RFC) faisant communiquer des machines situées au même niveau conceptuel.

Et enfin le protocole BFD (RFC 5880), un protocole de détection de pannes dans les routeurs, prévu pour être très rapide, afin de détecter les problèmes plus tôt que par les techniques de gestion classiques. C'est conceptuellement un service du plan de contrôle.

Pour résumer ? Lire la section 5 et bien se rappeler que 1) SDN ne se limite pas à utiliser OpenFlow 2) Il y a beaucoup de marketing et beaucoup moins de substance.

Quelques lectures classiques citées par notre RFC (la bibliographie de ce RFC est particulièrement longue, reflet de la complexité, de la richesse, et de la confusion du sujet). D'abord, les articles historiques :

— « "*A survey of programmable networks*" `<http://dl.acm.org/citation.cfm?id=505735>` » de Campbell et T. Andrew ("*ACM SIGCOMM Computer Communication Review 29.2 (1999)*"),
— « "*OpenFlow : enabling innovation in campus networks*" `<ftp://tmail.lzu.edu.cn/public/incoming/OpenFlow/openflow-wp-latest.pdf>` » de McKeown et Nick ("*ACM SIGCOMM Computer Communication Review 38.2 (2008)*"),

— « "*Network virtualization : state of the art and research challenges*" <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5183468> » de Chowdhury, NM Mosharaf Kabir, et Raouf Boutaba ("*Communications Magazine, IEEE 47.7 (2009)*"),
— « "*A clean slate 4D approach to network control and management*" <http://www.cs.cmu.edu/~4D/papers/greenberg-ccr05.pdf> » de Greenberg, Albert et autres ("*ACM SIGCOMM Computer Communication Review 35.5 (2005)*"),
— Puis une histoire du concept de SDN dans « "*The Road to SDN*" <http://www3.cs.stonybrook.edu/~phillipa/CSE390/sdnhistory.pdf> » de Feamster, Nick, Jennifer Rexford, et Ellen Zegura ("*ACM Queue11, no. 12 (2013)*"),
— Une étude de l'état du SDN en 2014, « "*A Survey of Software-Defined Networking : Past, Present, and Future of Programmable Networks*" <https://hal.inria.fr/hal-00825087v2/document> » de Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, et Thierry Turletti ("*IEEE Communications Surveys and Tutorials DOI :10.1109/SURV.2014.012214.00180*"),
— Le RFC 7149.