# Software Defined Network
# Principes et Architecture

**EFORT**
**http://www.efort.com**

## 1. Introduction

Le marché s'est rapidement approprié le SDN (Software Defined Network) comme ensemble de solutions/architectures permettant de supprimer les frontières existantes entre les mondes des applications et du réseau. Alors que le déploiement d'applications est toujours plus aisé et dynamique, notamment grâce à la virtualisation et au Cloud, le réseau reste parfois perçu comme un frein.
Ce dernier ne semble pas avoir évolué et chaque modification nécessite encore souvent des interventions manuelles sans orchestration dans la plupart des organisations. La mise en place d'un simple VLAN (Virtual LAN) peut parfois prendre plusieurs jours, avec la nécessité de configurer un par un de nombreux équipements : commutateurs, routeurs, firewalls, système de détection d'intrusion, etc.
La discussion a donc naturellement dévié sur les réelles problématiques réseau : simplifier le déploiement d'applications sur le réseau, mettre en place des politiques de sécurité / QoS homogènes globalement pour les applications, automatiser des fonctions réseau directement à partir des applications.
Le SDN est donc plus globalement reconnu aujourd'hui comme une architecture permettant d'ouvrir le réseau aux applications.
Le but de ce tutoriel est d'introduire les principes et l'architecture SDN.

## 2. Approche moderne de l'informatique et des réseaux

Une analogie peut être faite entre la manière dont l 'informatique a évolué de systèmes fermés, intégrés verticalement, propriétaires vers une approche ouverte et l 'évolution à venir avec SDN.
Auparavant les fournisseurs tels qu 'IBM et DEC fournissaient des produits totalement intégrés, avec un matériel de processeurs propriétaires, un langage assembleur, un système d 'exploitation (OS) et toutes les applications logicielles. Dans cet environnement, les grands clients étaient liés à un fournisseur, dépendant d 'abord des applications offertes par ce fournisseur. La migration vers la plate-forme matérielle d 'un autre fournisseur conduisait à des bouleversements majeurs au niveau application.
Aujourd 'hui l 'environnement informatique est caractérisé par une ouverture extrême, et une flexibilité pour le client final. Le matériel informatique consiste en des processeurs x-86 pour des systèmes indépendants et des processeurs ARM pour des systèmes embarqués. Cela permet de porter facilement un OS implanté en C, C++ ou JAVA, etc.. Ainsi, des applications écrites pour Linux ou un autre OS peuvent être aisément portée d 'une plate-forme d 'un fournisseur à une autre. Même des systèmes propriétaires comme Windows ou Mac OS fournissent des environnements de programmation afin que le portage d 'application soit une tâche aisée. Il permet aussi le développement de machines virtuelles qui peuvent être déplacées d 'un serveur à l 'autre entre des plates-formes matérielles et systèmes d 'exploitation.
L 'environnement réseau aujourd 'hui fait face aux mêmes limitations que celles rencontrées jadis dans l 'informatique. Le problème n 'est pas de ne pas pouvoir développer des applications qui puissent s 'exécuter sur des plates-formes diverses. Plutôt, la difficulté réside dans le manque d 'intégration entre les applications et l 'infrastructure réseau. Les

 

architectures de réseau traditionnelles sont inadéquates pour faire face à la croissance du volume de trafic et la variété de ce trafic.
Le concept central sous-jacent à SDN est de permettre aux développeurs et aux gestionnaires de réseau le même type de contrôle sur les équipements réseau que celui dont ils disposent sur les serveur x86. L 'approche SDN sépare la fonction de commutation entre le plan usager et le plan de contrôle qui sont présents sur des noeuds différents. Le plan de données est simplement responsable de relayer les paquets, alors que le plan de contrôle fournit l 'intelligence afin d 'identifier les routes, de positionner les priorités, de positionner les politiques de QoS et QoE (Quality of Experience), et de faire face à la variété et au volume de trafic. Des interfaces ouvertes sont définies afin que le matériel de commutation présente une interface uniforme indépendamment de l 'implantation interne. Aussi, des interfaces ouvertes sont définies afin de faire communiquer des applications réseau avec les contrôleurs SDN (Figure 1).

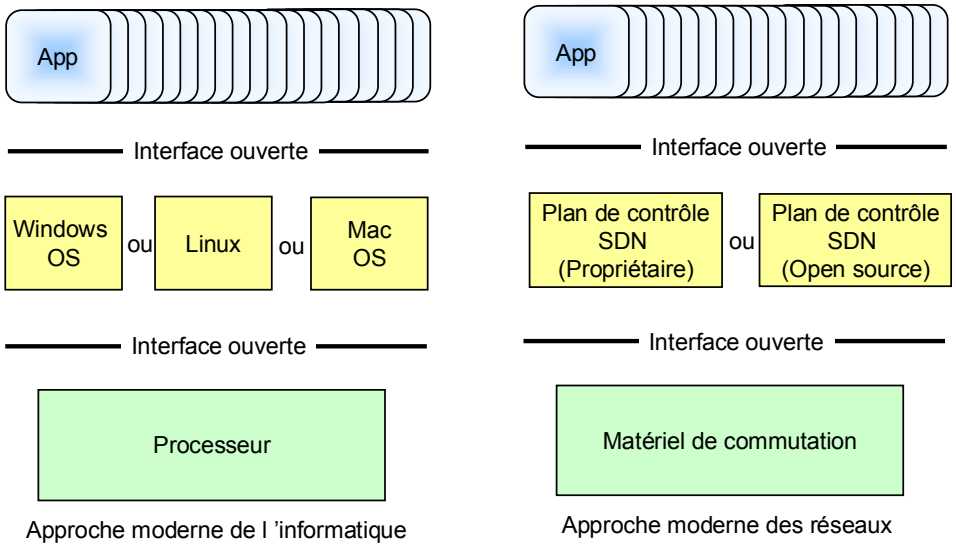


Figure 1 : Approche moderne de l 'informatique et des réseaux

## 3. Du réseau IP traditionnel au SDN

Au sein d 'un réseau IP, les commutateurs / routeurs remplissent les fonctions suivantes (Figure 2):
1. échanger des informations avec d'autres commutateurs / routeurs, c 'est à dire annoncer ses propres capacités et connexions et apprendre les capacités et les relations des autres routeurs/commutateurs
2. Calculer les chemins en fonction des informations apprises
3. Acheminer les données sur la base des chemins calculés
Les deux premières fonctions sont des fonctions «d'intelligence» et sont exécutées par le logiciel de l'équipement. La troisième est une fonction « d 'exécutant » exécutée par le matériel de l'équipement.

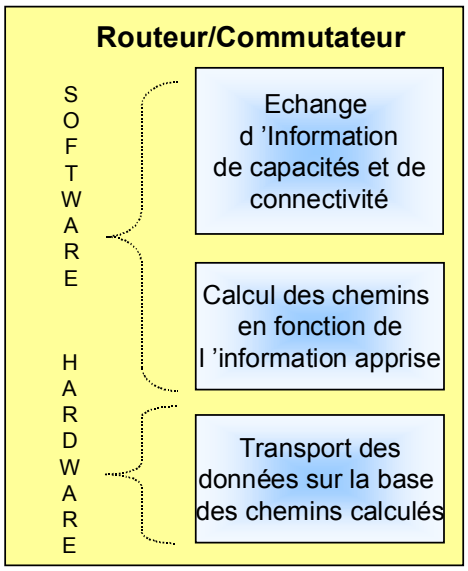


Figure 2 : Représentation simplifiée d 'un routeur/commutateur dans un réseau IP

Parmi les protocoles d 'échange d 'information figurent (Figure 3):

- OSPF (Open Shortest Path First) pour l 'échange d 'information permettant la mise à jour des tables de routage des routeurs d 'un système autonome.
- BGP (Border Gateway Protocol) pour échanger des informations de routage et d'accessibilité de réseaux entre systèmes autonomes.
- LDP (Label distribution Protocol) afin d 'échanger les informations de label dans le contexte MPLS.
- IGMP (Internet Group Management Protocol) est un protocole entre le(s) routeur(s) multicast du LAN et les hôtes multicast du LAN. Il permet à un hôte de s'abonner (désabonner) à un groupe et de demander au routeur de recevoir une copie des paquets reçus par une adresse multicast.
- Entre routeurs, il est nécessaire de disposer de protocoles pour construire et maintenir les arbres de distribution. Les protocoles PIM (Protocol Independent Multicast) et MSDP (Multicast Source Discovery Protocol) sont considérés.

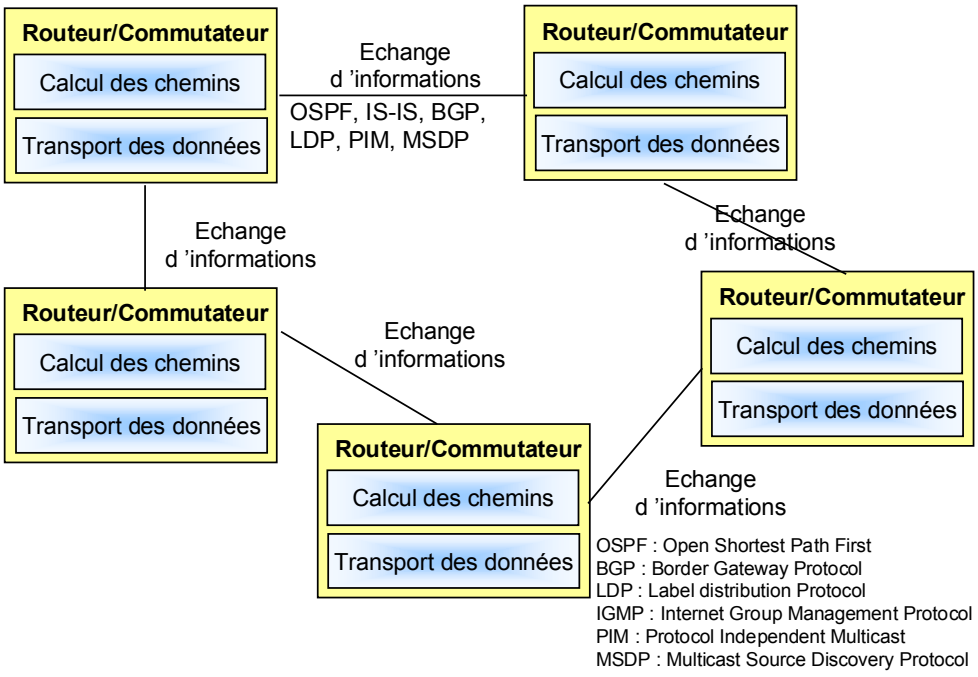


Figure 3 : Réseau de routeurs/commutateurs

Avec l'approche SDN (Figure 4), des applications présentent dans des serveurs d'applications où associées au contrôleur calculent les chemins. Ces chemins sont ensuite passé aux contrôleur qui les transmet aux fonctions d'acheminement. Chaque fonction d'acheminement dispose d'une table de flux mise à jour par le contrôleur.

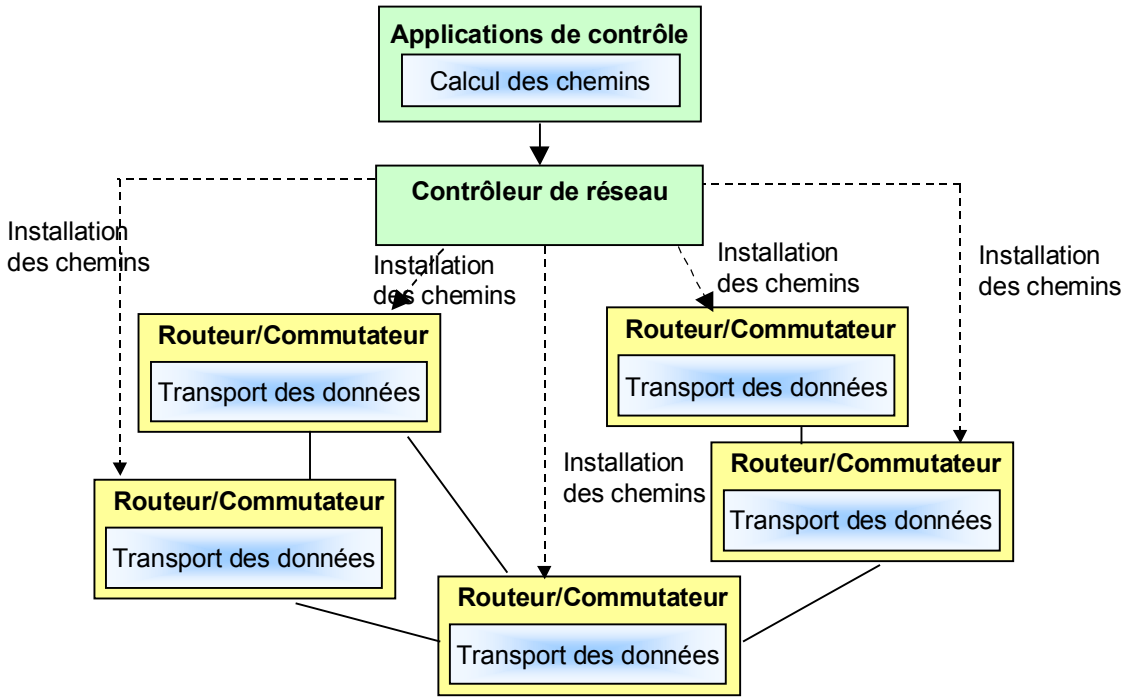


Figure 4 : Réseau de routeurs/commutateur avec un contrôleur SDN

L'architecture SDN est définie par trois principes fondamentaux qui la différencient de l'architecture des réseaux traditionnels

- Plans de contrôle et de données
  - Réseau traditionnel : Les plans de contrôle et de données sont colocalisés dans les éléments de réseau
  - SDN : Le plan de contrôle est séparé du plan de données et déplacé au contrôleur SDN
- Intelligence du contrôle
  - Réseau traditionnel : L 'intelligence du contrôle est distribuée dans chaque élément de réseau
  - SDN : L 'intelligence de contrôle est centralisée dans un contrôleur SDN
- Programmation du réseau par des applications
  - Réseau traditionnel : Le réseau ne peut pas être programmé par des applications. Chaque élément de réseau doit être configuré séparément.
  - SDN : Le réseau peut être programmé par des applications. Le contrôleur SDN peut exposer des interfaces d 'application pour la manipulation du réseau.

La séparation du plan de contrôle et du plan usager peut être réalisée via des interfaces de programmation bien définies entre les commutateurs/routeurs et le contrôleur SDN. Le contrôleur exerce un contrôle direct sur l 'état d 'acheminement des paquets dans les éléments du plan usager via la Southbound API (Application Programming Interface) montrée à la figure 5.
L 'exemple le plus notable d 'une telle API est Openflow. Un commutateur OpenFlow dispose d 'une ou plusieurs tables de règles de traitement de paquet (flow table). Chaque règle s 'applique à un sous ensemble du trafic et réalise certaines actions (dropping, forwarding,

modifying, etc.) sur le trafic. Le protocole Openflow a fait l'objet d'un autre tutoriel proposé par EFORT : http://www.efort.com/index.php?PageID=5&l=fr
En fonction des règles installées par l 'application du contrôleur, un commutateur Openflow peut (d 'après les instructions du contrôleur) se comporter comme un routeur, un commutateur, un firewall) ou réaliser d 'autres rôles (e.g., load balancer, trafic shaper pour lisser le trafic, et généralement les rôles d 'une middlebox). Une middle box est un dispositif de réseau intermédiaire permettant d'implanter des services divers tels qu'un fltrage de paquets, un VPN, une détection d'intrusion, une translation d'adresse NAT ou un pare-feu. Une middle box est donc une appliance située entre deux équipements de réseau, d'où son nom de middle box. En fait, middle box et appliance sont deux noms qui désignent pratiquement les mêmes équipements.
Une conséquence importante des principes SDN est la séparation introduite entre la définition des politiques et leur implantation d 'une part et l 'acheminement du trafic d 'autre part.
Cette séparation est la clé pour la flexibilité désirée, simplifiant ainsi la gestion de réseau et facilitant l 'évolution du réseau.
Le contrôleur dispose d 'une NorthBound API pour offrir aux applications une API de haut niveau pour la mise en œuvre d 'applications réseau qui n 'étaient pas possible avant l 'avènement du SDN. Une majorité de fournisseurs proposent une API basée sur REST http (Representational State Transfer) pour l 'interface de programmation au contrôleur SDN. Les applications SDN sont des programmes qui peuvent utiliser une vue abstraite du réseau pour leur prises de décision. Ces applications passent au contrôleur leur comportement réseau désiré via la Northbound API. Parmi des exemples d 'applications figurent la supervision de la sécurité, le contrôle d 'accès, et la gestion réseau.
Aussi considérées mais non encore définies sont les Westbound et Eastbound API qui permettent la communication et la coopération entre des groupes ou des fédérations de contrôleur afin de synchroniser l 'état. Des propositions Westbound et Eastbound d 'API ont été faites telles que SDNi, ForCES et CE-CE.

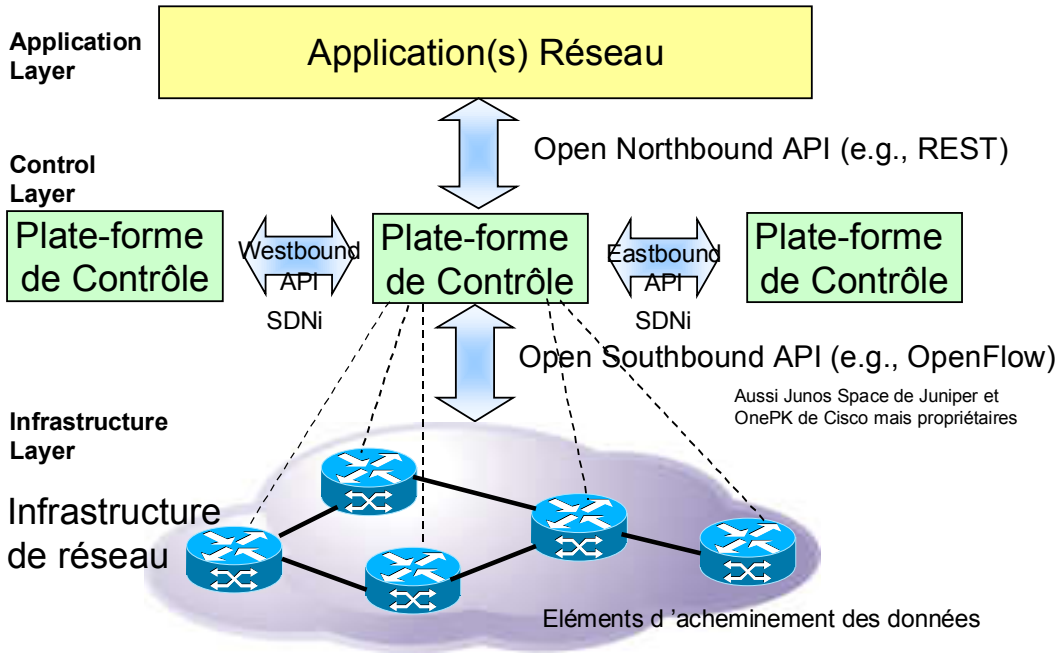


Figure 5 : Vue simplifiée de l'architecture SDN

# 4. Programmation des équipements

La programmation des équipements réseau nécessite sur ces derniers la capacité de recevoir des directives de l'extérieur. Pour cela des interfaces de programmation sont nécessaires : des API (Application Programming Interface).
Il existe de nombreuses API, standards ou propriétaires, pouvant agir sur différents éléments de l'équipement (plan de données, plan de gestion) Il est communément admis qu'une seule API universelle ne suffira pas pour résoudre toutes les problématiques réseau. Aussi les équipements modernes implémentent souvent plusieurs API.
Parmi les APIs les plus communément supportées sur les équipements réseau figurent : CLI, Netconf/YANG, Openflow, Restful API, SNMP, HTTP, XMPP, etc.
CLI : L'accès en ligne de commande aux équipements via telnet/ssh est bel et bien une API et il est important de ne pas l'oublier. C'est d'ailleurs encore celle qui est encore le plus souvent utilisée pour programmer/configurer le réseau. La CLI agit sur le plan de gestion.
• OpenFlow : OpenFlow est une API permettant la programmation du plan de données. Des actions sont programmées au niveau des flux (un ensemble de critères sur les paquets/trames : par exemple IP source, IP destination, DSCP, etc.). Tout flux correspondant à une entrée dans la table OpenFlow de l'équipement sera traité selon les actions demandées.
• Netconf/YANG : Netconf est un standard IETF (RFC 6241) visant à standardiser les directives réseau auprès des équipements. YANG est un langage permettant la modélisation des services réseau. Une directive Netconf pourra contenir un modèle YANG et ainsi configurer de la même manière des équipements de constructeurs différents.
• RESTful API (REpresentational State Transfer) : On parle de RESTful API quand les directives fournies sont faites via des directives de type HTTP (GET, POST, DELETE, PATCH, PUT) La principale caractéristique d'une API RESTful est qu'elle est sans état. En outre, chaque requête correspond à une demande. Il n'y a pas de dialogue possible entre les extrémités de la chaîne via une connexion préalablement établie. On dénote l'API RESTconf qui permet d'échanger via une API RESTful un modèle YANG. Ces API sont plutôt privilégiées par les communautés de développeurs web et seront plus souvent utilisées dans le contexte d'un datacenter puisque cela correspond au langage naturel de programmation des équipes en charge des applications
Il existe de très nombreuses autres interfaces de programmation, plus ou moins ouvertes et adaptées à des environnements spécifiques telles que SNMP, HTTP, etc.

# 5. Applications SDN

- SDN est utilisé dans les data centers modernes pour la configuration des VLAN/VxLANs.
- SDN est utilisé pour le WAN d'entreprise pour le routage intelligent des flux IP des différentes agences de l'entreprise.
- SDN est utilisé dans le réseau mobile pour le chainage de service dynamique. Avant que les flux entre l'usager mobile n'arrivent sur l'Internet les flux sont acheminés via des fonctions de service qui améliorent la qualité d'expérience de l'usager en fonction du flux et du profil de l'usager. parmi les fonctions de service figurent le contrôle parental, l'optimisation TCP et vidéo, l'enrichissement du flux, etc.
- SDN est utilisé dans le réseau 5G pour découpler le plan contrôler et le plan usager et ainsi choisir une fonction du plan usager la plus proche possible du client mobile pour réduire la latence (MEC, Mobile Edge Computing)

La formation EFORT « Virtualisation de Réseau et de Service, SDN et NFV » décrit la virtualisation de réseau et de service, les nouveaux concepts pour y parvenir tels que SDN et NFV avec leurs architectures et composants clés ainsi que les stratégies des principaux

acteurs du monde des réseaux de télécommunication dans ce domaine.
http://efort.com/index.php?PageID=21&l=fr&f_id=178&imageField.x=3&imageField.y=1