

Ninth SG13 Regional Workshop for Africa on

Standardization of Future Networks and Emerging Network Technologies:
African perspectives

les Réseaux programmables (SDN)

Présenté par :
KEUPONDJO Armel

- PLAN -

2

CONCEPTS

ARCHITECTURES TRADITIONNELLES ET SDN

CONTROLEURS

TYPES D'ARCHITECTURES SDN

LES DÉFIS SDN

CONCEPTS | Plans de contrôle et de données

3

Niveaux de traitement du routeur:

- Un plan de contrôle;
- Un plan de transfert.

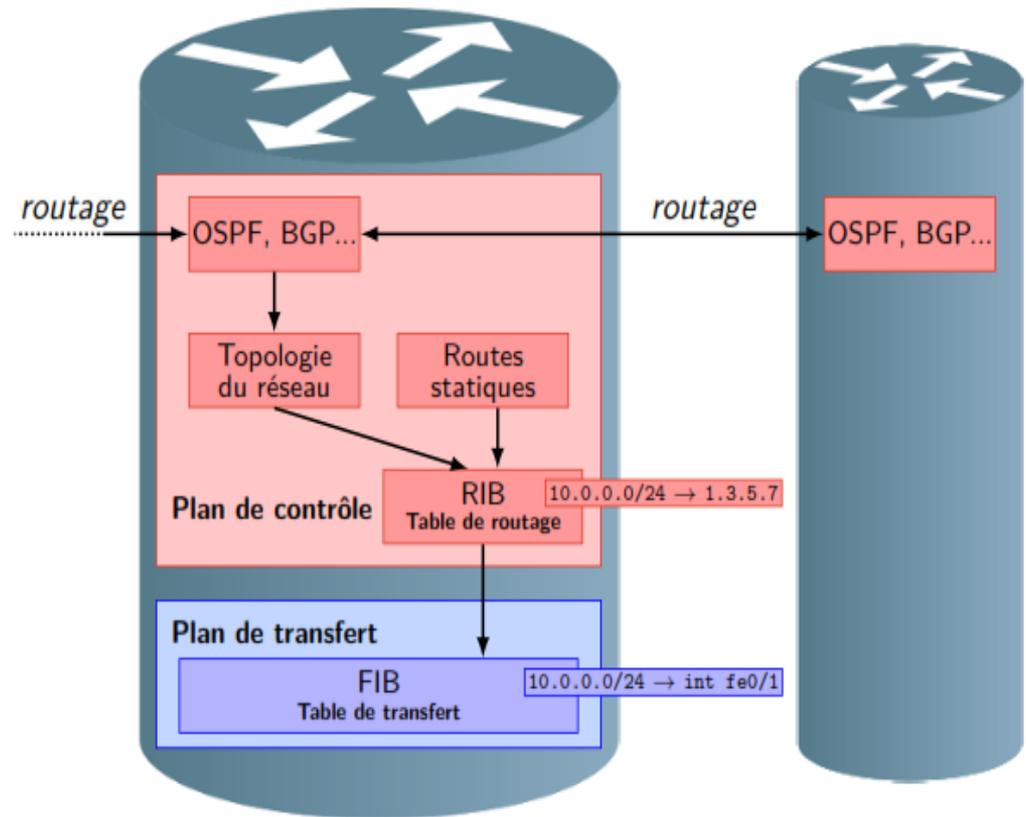


Figure 1: Schéma en couches d'un routeur

CONCEPTS | Plans de contrôle et de données

4

Un appareil réseau contient les plans suivants :

- ❑ **Plan de contrôle** - Il est généralement considéré comme le cerveau d'un appareil. Le plan de contrôle permet de prendre les décisions de transmission. Il comprend des mécanismes de transmission d'itinéraire de couche 2 et 3, notamment la table de voisinage et la table topologique (protocoles de routage), la table de routage pour IPv4 et IPv6, etc...
- ❑ **Plan de données** - Également appelé plan d'acheminement, ce plan est généralement la matrice de commutation qui relie les différents ports du réseau sur un appareil. Le plan de données de chaque périphérique permet de transmettre les flux de trafic. Les routeurs et les commutateurs utilisent les informations du plan de contrôle pour transmettre le trafic entrant vers l'interface de sortie appropriée.

CONCEPTS | Paradigme SDN

5

Un paradigme alternatif, cependant, attire actuellement l'attention: Software-Defined Networking (**SDN**) ou réseau défini par logiciels en français.

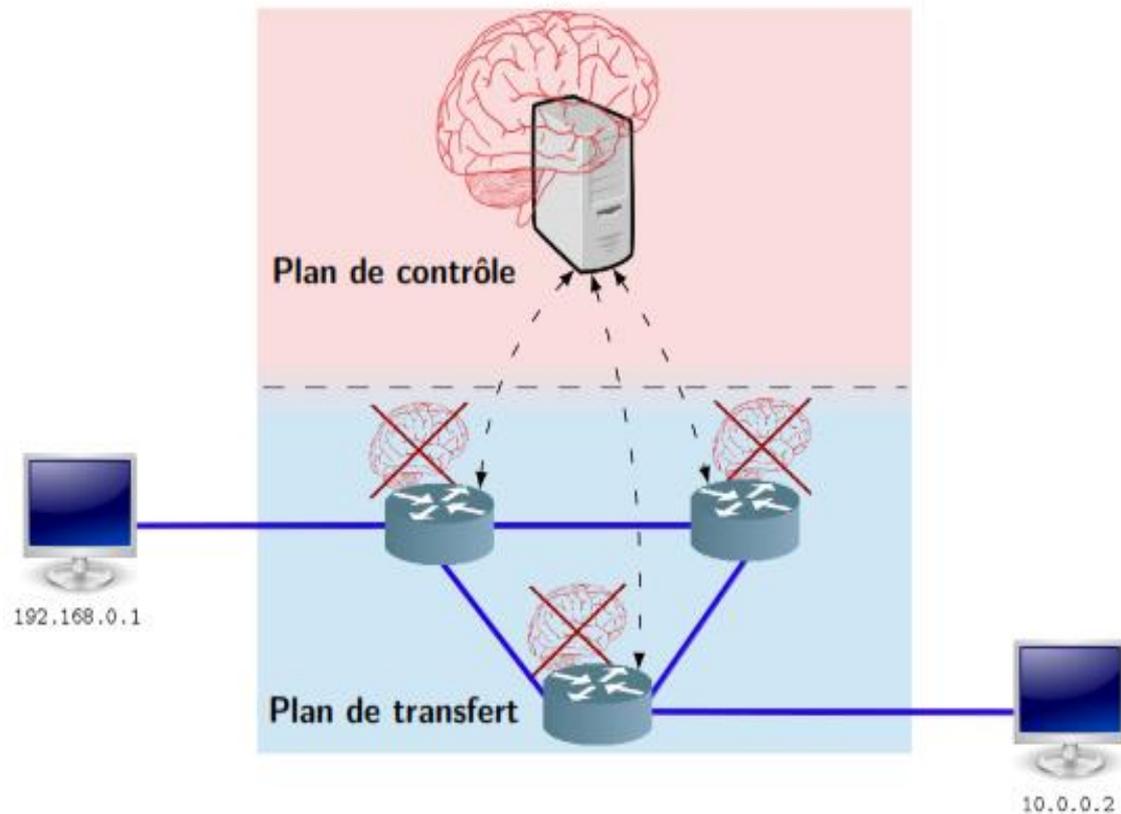


Figure 2: Schéma d'un réseau SDN simple

CONCEPTS | Paradigme SDN

6

SDN (Software-Defined Networking) - Une architecture de réseau qui virtualise le réseau, offrant une nouvelle approche de l'administration et de la gestion du réseau qui vise à simplifier et à rationaliser le processus d'administration.

Selon le **RFC 7426** 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

CONCEPTS | Paradigme SDN

7

Les composantes du SDN peuvent comprendre les éléments suivants :

- ✓ **OpenFlow** - Cette approche a été développée à l'université de Stanford pour gérer le trafic entre les routeurs, les commutateurs, les points d'accès sans fil et un contrôleur. Le protocole **OpenFlow** constitue un élément fondamental dans la mise en œuvre des solutions SDN.
- ✓ **OpenStack** - cette approche repose sur une plate-forme d'orchestration et de virtualisation pour créer des environnements cloud évolutifs et mettre en œuvre une solution IaaS (infrastructure en tant que service).

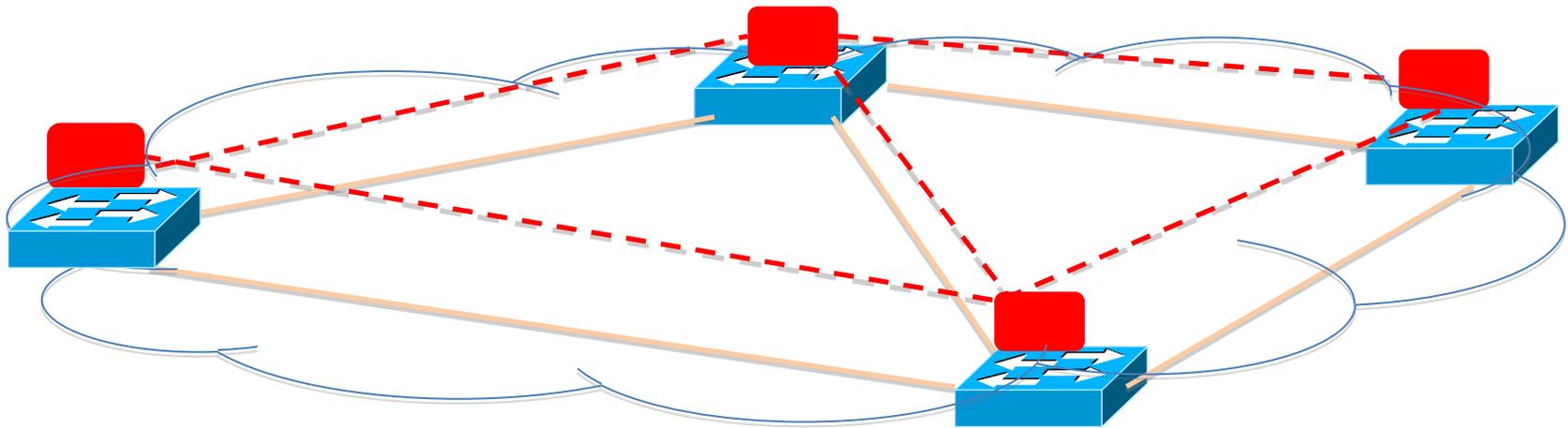
Architectures traditionnelles et SDN

Dans une architecture traditionnelle de routeurs ou de commutateurs, les fonctions de plans de contrôle et de données sont assurées sur un même périphérique. Les décisions relatives à l'acheminement et à la transmission des paquets relèvent de la responsabilité du système d'exploitation de l'appareil.

Dans SDN, la gestion du plan de contrôle est déplacée vers un contrôleur SDN centralisé. La figure compare les architectures traditionnelles et SDN.

Architectures traditionnelles et SDN

9

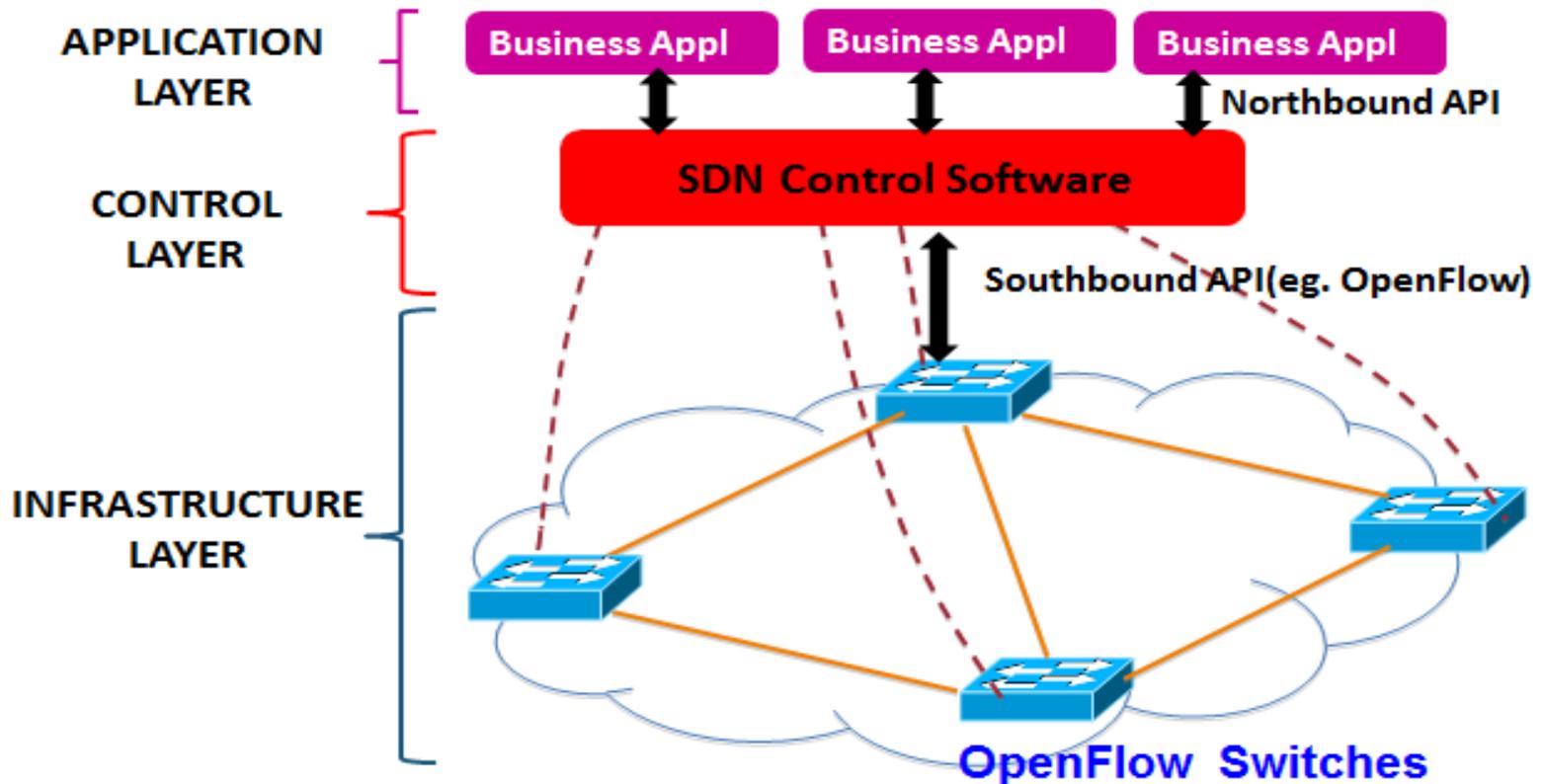


- **Le Plan de transfert de donnée:**
Paquet en streaming

- **Plan de Contrôle:**
Algorithmes de Routage

Architectures traditionnelles et SDN

10



Contrôleurs

11

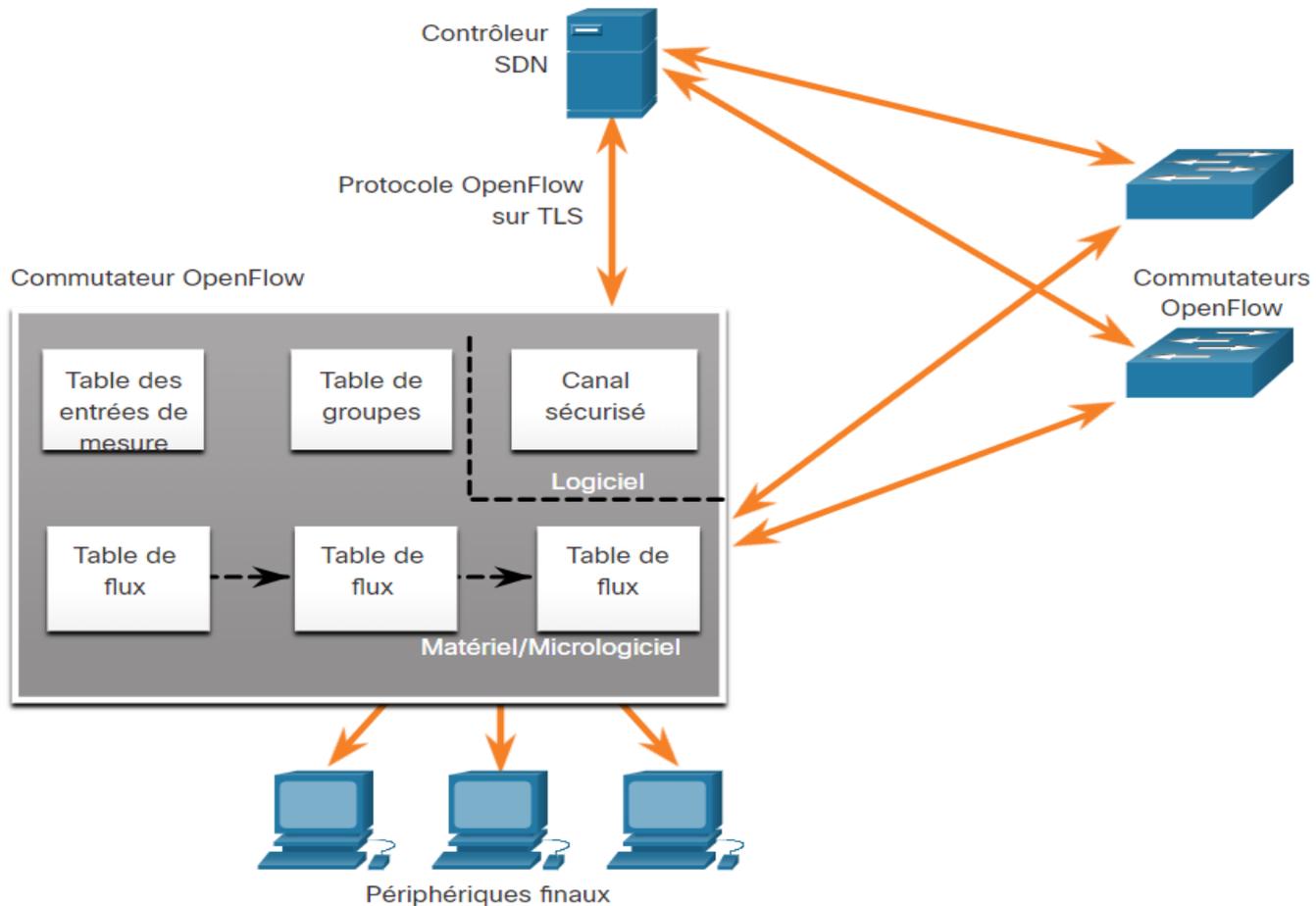
Fonctionnement

- ❑ Le contrôleur SDN définit les flux de données entre le plan de contrôle centralisé et les plans de données sur des routeurs et des commutateurs individuels.
- ❑ Pour pouvoir traverser le réseau, chaque flux doit être approuvé par le contrôleur SDN qui vérifie que la communication est autorisée dans le cadre de la politique réseau de l'entreprise. Si le contrôleur autorise le flux, il calcule l'itinéraire que ce dernier doit suivre et ajoute une entrée correspondante au flux dans tous les commutateurs situés sur le trajet.

Contrôleurs

12

Fonctionnement



Contrôleurs

Fonctionnement

- ❑ Dans la figure, un contrôleur SDN communique avec les commutateurs prenant en charge **OpenFlow** à l'aide de ce protocole. Ce protocole s'appuie sur le protocole TLS (Transport Layer Security) pour sécuriser les communications issues du plan de contrôle à l'échelle du réseau. Chaque commutateur **OpenFlow** se connecte à d'autres commutateurs **OpenFlow**. Ils peuvent également se connecter aux périphériques utilisateur qui font partie d'un flux de paquets.
- ❑ Sur chaque commutateur, la gestion des flux de paquets est assurée par une série de tables mises en œuvre au niveau du matériel ou du firmware. À l'échelle du commutateur, un flux est une séquence de paquets qui correspond à une entrée spécifique dans une table de flux.

Contrôleurs

14

Fonctionnement

Les trois types de tableaux présentés dans la figure précédente sont les suivants :

- ❑ **Table des flux** - Ce tableau fait correspondre les paquets entrants à un flux particulier et spécifie les fonctions qui doivent être exécutées sur les paquets. Il peut y avoir plusieurs tables de flux qui fonctionnent à la manière d'un pipeline.
- ❑ **Table de groupe** - Un tableau de flux peut diriger un flux vers un tableau de groupe, ce qui peut déclencher diverses actions qui affectent un ou plusieurs flux
- ❑ **Table de comptage** - Cette table déclenche une série d'actions liées aux performances sur un débit, y compris la capacité de limiter le trafic.

Contrôleurs

15

contrôleurs SDN les plus connus

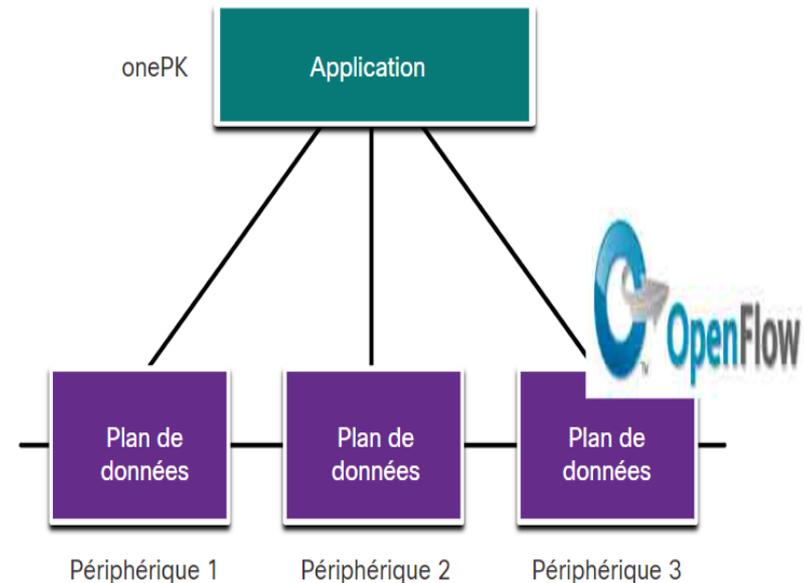
Contrôleur	Organisation	Langage	Fonctionnalités
NOX	Nicira	C++	le premier contrôleur openflow
POX	Nicira	Python	améliorer les performances de NOX
Ryu	NTT, OSRG group	Python	supporte l'OpenStack
Floodlight	Big Switch	Java	testé avec des commutateurs OpenFlow physiques et virtuels.
Opendaylight	Linux Foundation	Java	supporte le Framework OSGi et le REST API

Types d'architecture SDN

16

SDN basé sur les périphériques

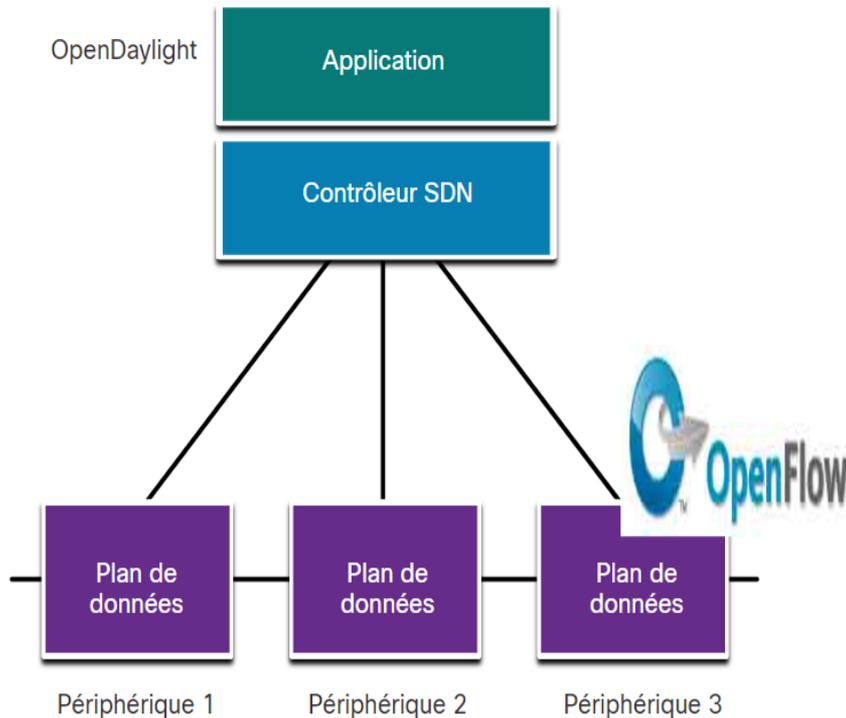
Dans ce type de SDN, les appareils sont programmables par des applications s'exécutant sur l'appareil lui-même ou sur un serveur du réseau, comme le montre la figure. Cisco OnePK est un exemple de SDN basé sur les périphériques. Il permet aux programmeurs de développer des applications en C et Java avec Python pour qu'elles s'intègrent et interagissent avec les périphériques Cisco.



Types d'architecture SDN

17

SDN basé sur les contrôleurs



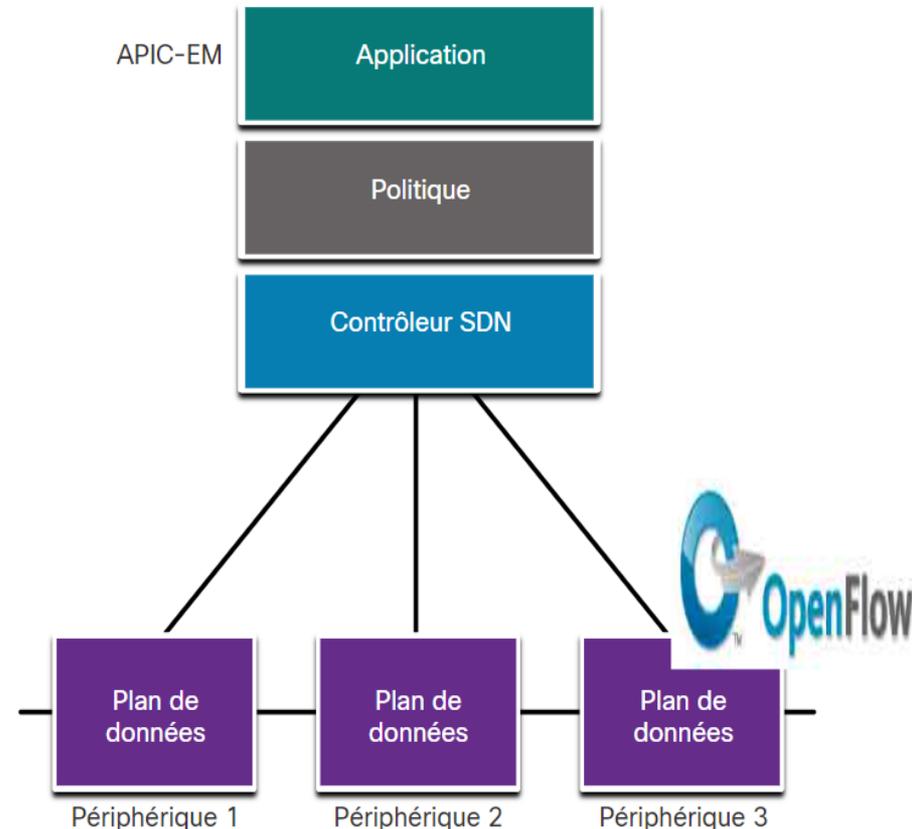
Ce type de SDN utilise un contrôleur centralisé qui connaît tous les appareils du réseau, comme le montre la figure. Les applications peuvent interagir avec le contrôleur, chargé de gérer les périphériques et de manipuler les flux de trafic sur le réseau. Le contrôleur Cisco Open SDN est une distribution commerciale d'OpenDaylight.

Types d'architecture SDN

18

SDN basé sur les politiques

Le SDN basé sur des politiques comprend une couche stratégique supplémentaire qui fonctionne à un niveau d'abstraction supérieur. Il utilise des applications intégrées qui automatisent les tâches de configuration avancée grâce à un workflow guidé et à une interface graphique utilisateur conviviale. Aucune compétence de programmation n'est nécessaire.



LES DÉFIS SDN

19

Performance

SDN est une technique basée sur les flux, ses performances sont mesurées en fonction de deux métriques : le temps nécessaire pour instaurer un nouveau flux dans les commutateurs (latence) et le nombre de flux que le contrôleur peut traiter par seconde (débit).

- ❖ *L'utilisation des contrôleurs multiples est une solution plus efficace pour améliorer les performances des contrôleurs SDN.*
- ❖ *Les solutions tels que HyperFlow, Opendaylight déploient un plan de contrôle logiquement centralisé, ou plusieurs contrôleurs sont utilisés, en partageant entre eux les charges et synchronisant les données, ce qui permet d'améliorer les performances et assurer la cohérence de réseau.*

LES DÉFIS SDN

20

Scalabilité

Un autre défi du **SDN** est la scalabilité ou l'évolutivité du réseau, plus la taille de réseau augmente, plus des demandes sont envoyés au contrôleur et à un moment donné, le contrôleur devient incapable de traiter toutes ces demandes.

- ❖ *Une solution viable pour surmonter les problèmes de scalabilité est de conserver de manière proactive, tout le trafic dans le plan de données, en dirigeant les paquets via des commutateurs intermédiaires stockant les règles nécessaires.*
- ❖ *Une autre solution pour améliorer la scalabilité des contrôleurs SDN, est l'utilisation des contrôleurs multiples, des solutions permettant de distribuer physiquement les contrôleurs SDN, tout en maintenant la vue globale de réseau.*

LES DÉFIS SDN

21

Sécurité

Le fait de centraliser toute l'intelligence du réseau dans un seul contrôleur peut accroître la vulnérabilité du contrôleur. Un contrôleur SDN représente le point critique de réseau, s'il est compromis ou devient indisponible, tous les aspects du réseau seront endommagés.

- ❖ *les solutions proposées dans le contrôle d'accès : **AuthFlow** est un mécanisme d'authentification et de contrôle d'accès basé sur les informations d'identification de l'hôte, permet de refuser l'accès aux hôtes non autorisés.*
- ❖ *Un certain nombre de solutions ont été proposées pour surmonter l'attaque par déni de service sur les contrôleurs SDN ou sur les tables de flux des commutateurs. Comme la limitation des demandes de flux envoyées au plan de contrôle à l'aide d'un outil de migration de connexion.*

LES DÉFIS SDN

22

Fiabilité

Dans les premiers déploiements de réseaux SDN qui utilisaient un seul contrôleur centralisé, responsable de tout le réseau, cela pose des problèmes sur les contrôleurs SDN, qui deviennent des points uniques de défaillance

- ❖ *mécanismes pour détecter les défaillances dans les contrôleurs*
- ❖ *méthode de pré-partitionnement entre contrôleurs*

REFERENCES

- ✓ D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, et S. Uhlig, « Software-Defined Networking: A Comprehensive Survey », Proc. IEEE, vol. 103, no 1, p. 14-76, janv. 2015.
- ✓ Benamrane et al. - 2017 - Etude des Performances des Architectures du Plan d.pdf .
- ✓ P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, et G. Gu, « A security enforcement kernel for OpenFlow networks », in Proceedings of the first workshop on Hot topics in software defined networks - HotSDN '12, Helsinki, Finland, 2012, p. 121.
- ✓ Yi-Chen Chan, Kuochen Wang, et Yi-Huai Hsu, « Fast Controller Failover for Multi-domain SoftwareDefined Networks », in 2015 European Conference on Networks and Communications (EuCNC), Paris, France, 2015, p. 370-374.

Merci de votre attention