Ninth SG13 Regional Workshop for Africa on

Standardization of Future Networks and Emerging Network Technologies:

African perspectives

Programmable networks (SDN)

Presented by:

KEUPONDJO Armel

- PLAN -

DESIGNS

TRADITIONAL AND SDN ARCHITECTURE

CONTROLLERS

TYPES OF SDN ARCHITECTURES

SDN CHALLENGES

STANDARDIZATION AND THE AFRICAN CONTEXT SDN

CONCEPTS | Control and data plans

Router processing levels:

A control plane;

A transfer plan.

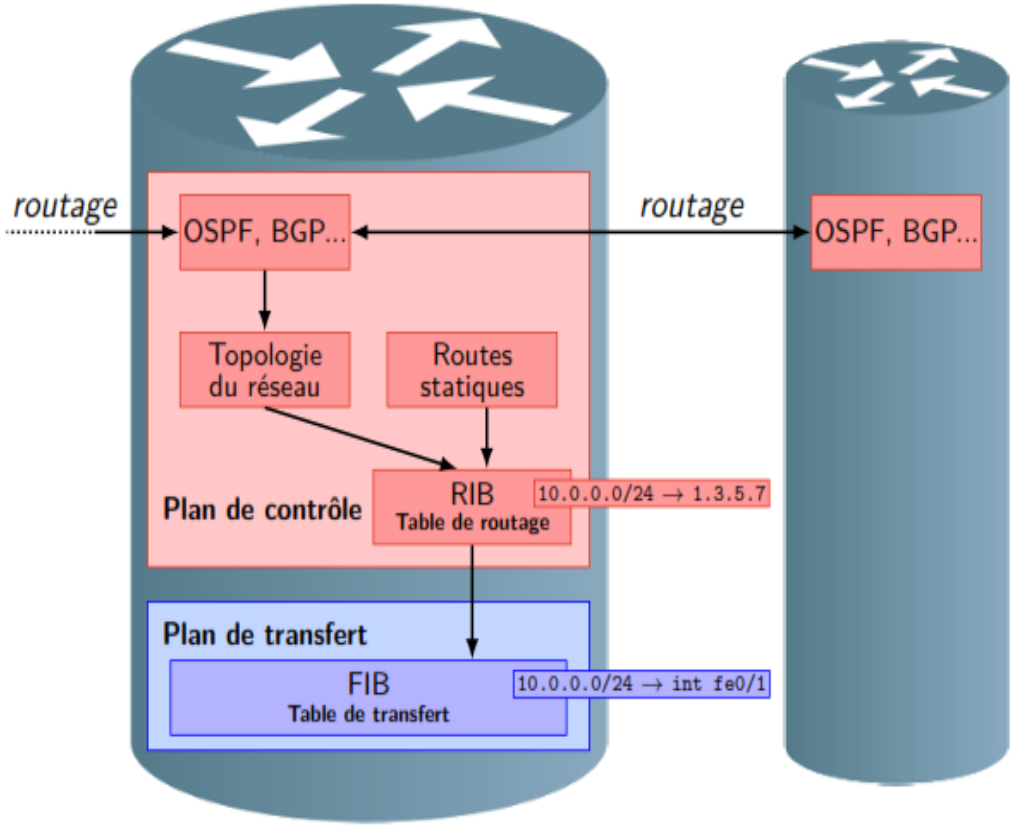


Figure 1: Layers of a router

CONCEPTS | Plans de contrôle et de données

A network device contains the following plans:

- **Control plane -** This is generally considered to be the brain of a device. The control plane is used to make transmission decisions. It includes layer 2 and 3 route transmission mechanisms, such as the neighbourhood table and the topology table (routing protocol), the routing table for IPv4 and IPv6, etc...

- **Data plane -** Also known as the routing plane, this is usually the switching matrix that connects the different network ports on a device. Each device's data plane is used to transmit traffic flows. Routers and switches use the control plane information to forward incoming traffic to the appropriate output interface.

CONCEPTS | Paradigme SDN

An alternative paradigm, howeve
is currently attracting attention
Software-Defined Networking
(SDN).

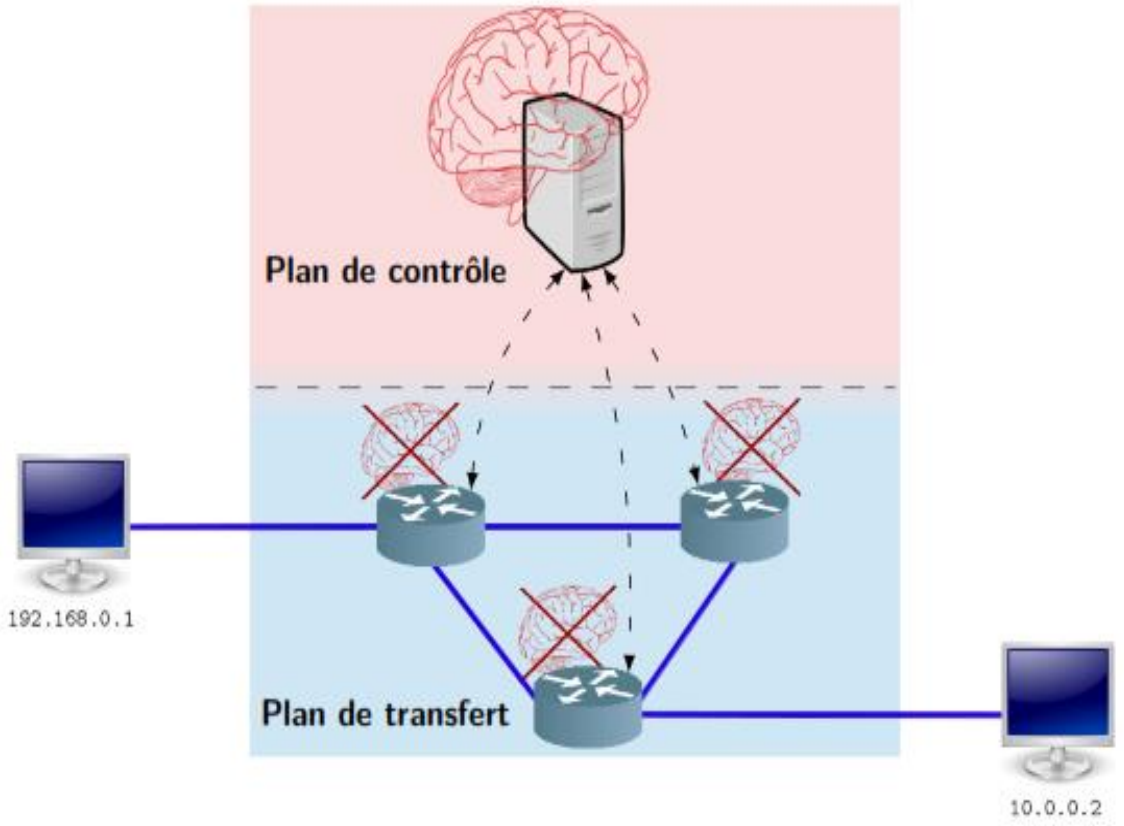


Figure 2: Schéma d'un réseau SDN simple

CONCEPTS | Paradigme SDN

SDN (Software-Defined Networking) - A network architecture that virtualises the network, offering a new approach to network administration and management that aims to simplify and streamline the administration process.

According to **RFC 7426**, SDN is an approach to networking based on programmability, which separates the control plane from the transmission plane and uses standardised interfaces.

CONCEPTS | Paradigme SDN

SDN components can include the following elements:

- ✓ **OpenFlow -** This approach was developed at Stanford University to manage traffic between routers, switches, wireless access points and a controller. The OpenFlow protocol is a fundamental element in the implementation of SDN solutions.

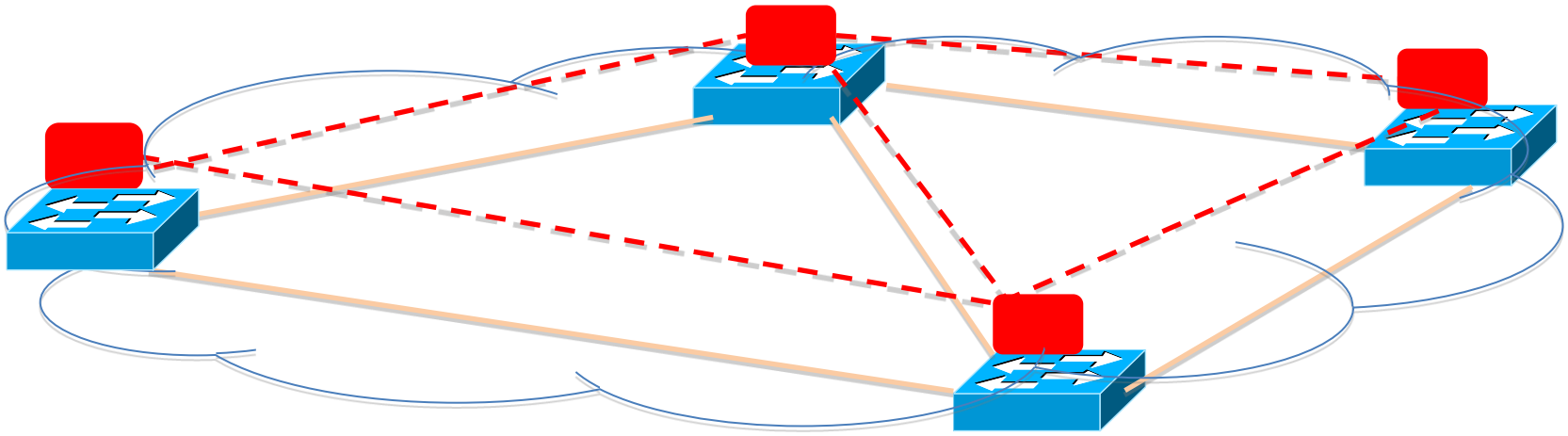
- ✓ **OpenStack -** this approach uses an orchestration and virtualisation platform to create scalable cloud environments and implement an Infrastructure as a Service (IaaS) solution**.**

Traditional architectures and SDN

In a traditional router or switch architecture, the control and data plane functions are performed on the same device. Decisions about packet routing and transmission are the responsibility of the device's operating system.

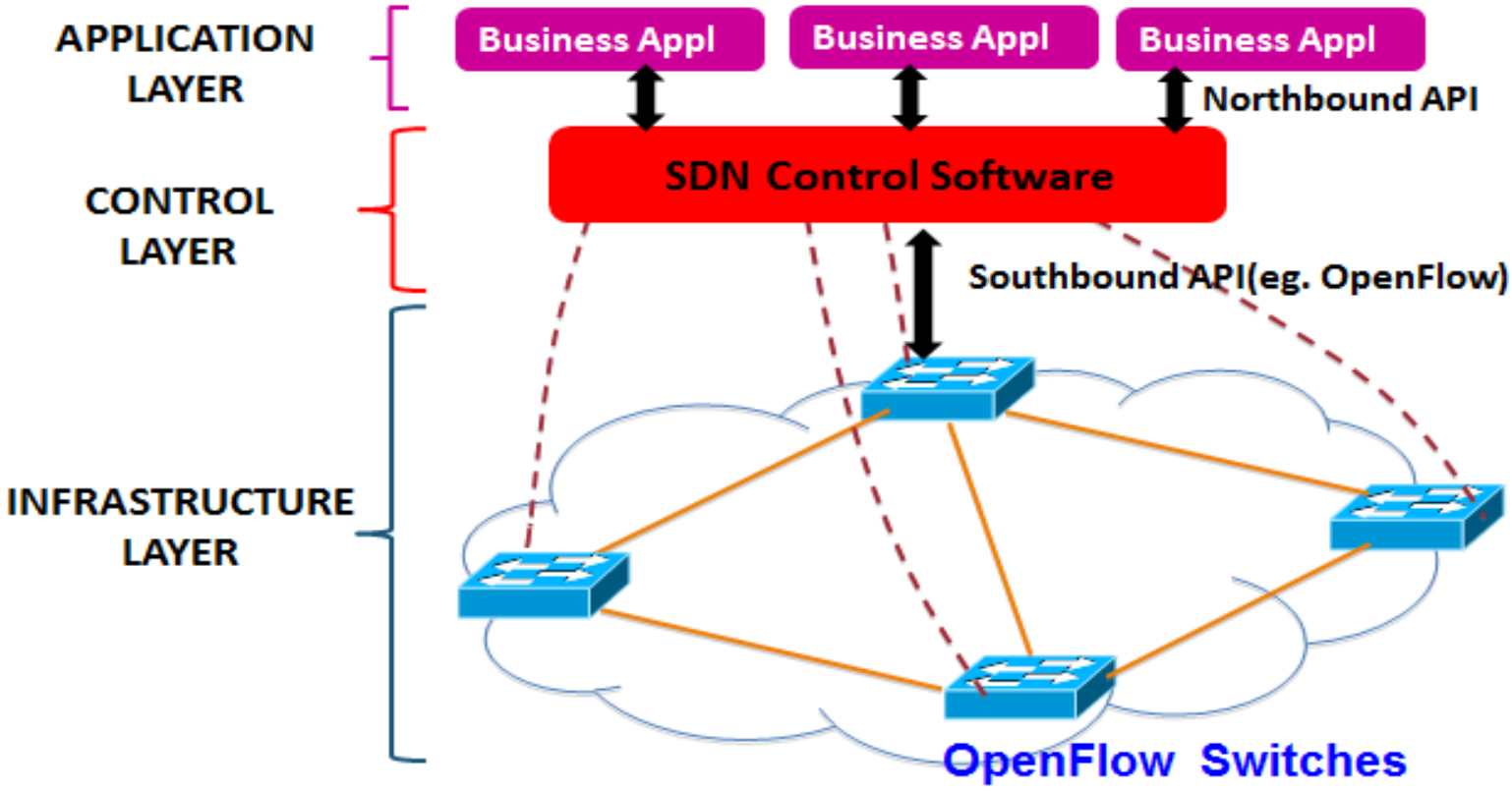
In SDN, management of the control plane is moved to a centralised SDN controller. The figure compares traditional and SDN architectures.

Traditional architectures and SDN



- **The Data Transfer Plan:** Paquet en streaming
- **Control plan:** Algorithmes de Routage

Traditional architectures and SDN



Controllers

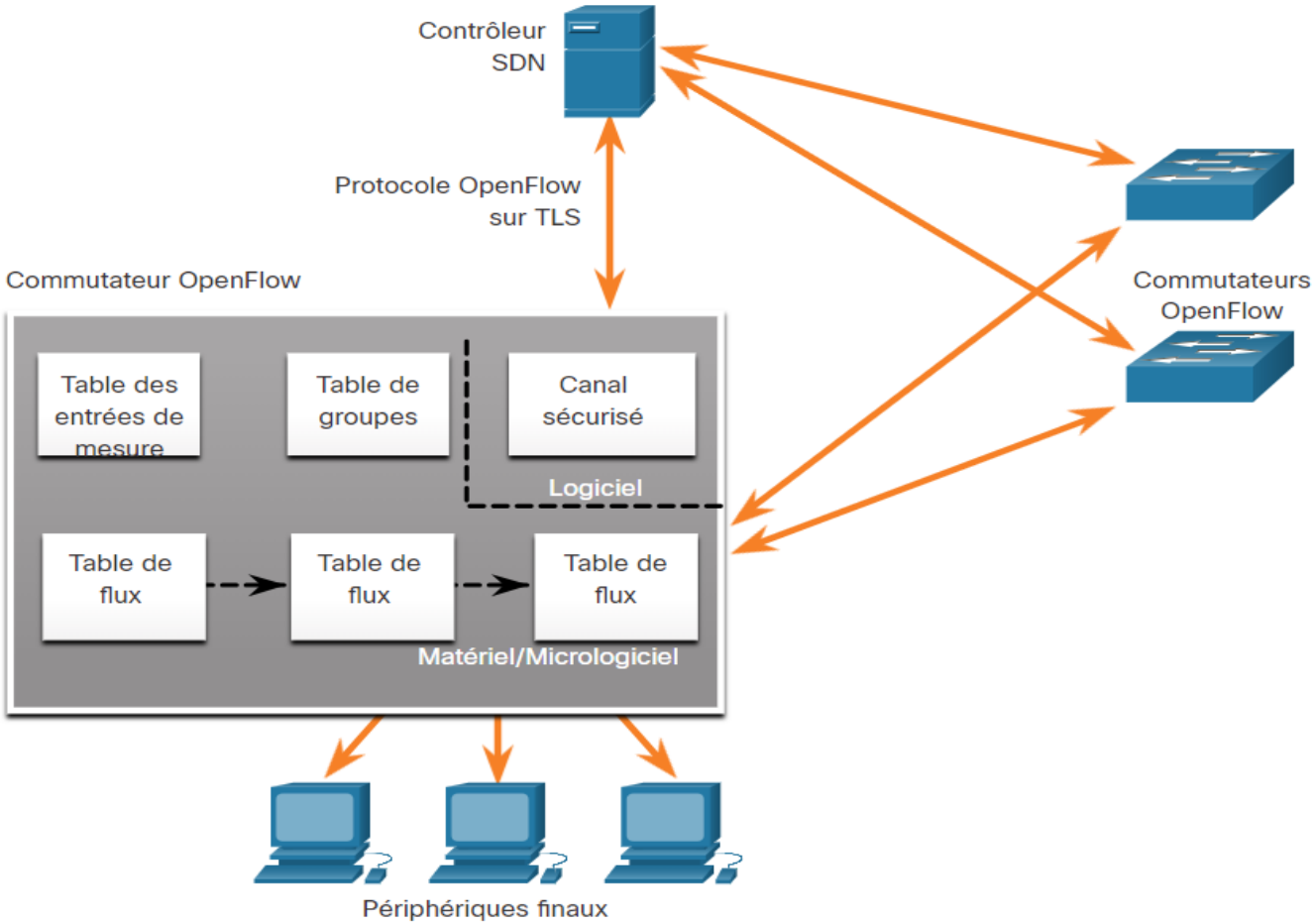
How it works

- The SDN controller defines the data flows between the centralised control plane and the data planes on individual routers and switches.

- In order to traverse the network, each flow must be approved by the SDN controller, which checks that the communication is authorised within the framework of the company's network policy. If the controller authorises the flow, it calculates the route it must follow and adds a corresponding entry to the flow in all the switches on the path.

Controllers

How it works



Controllers

How it works

- In the figure, an SDN controller communicates with switches supporting **OpenFlow** using this protocol. This protocol uses Transport Layer Security (TLS) to secure communications from the network-wide control plane. Each **OpenFlow** switch connects to other **OpenFlow** switches. They can also connect to user devices that are part of a packet flow.

- On each switch, packet flows are managed by a series of tables implemented at hardware or firmware level. At switch level, a flow is a sequence of packets corresponding to a specific entry in a flow table.

Controllers

How it works

The three types of tables shown in the previous figure are as follows:

- **Flow table -** This table maps incoming packets to a particular flow and specifies the functions to be performed on the packets. There may be several flow tables, which function in the same way as a pipeline.
- **Group table -** A flow table can direct a flow to a group table, which can trigger various actions that affect one or more flows.
- **Metering table -** This table triggers a series of performance-related actions on a flow, including the ability to limit traffic.

Controllers

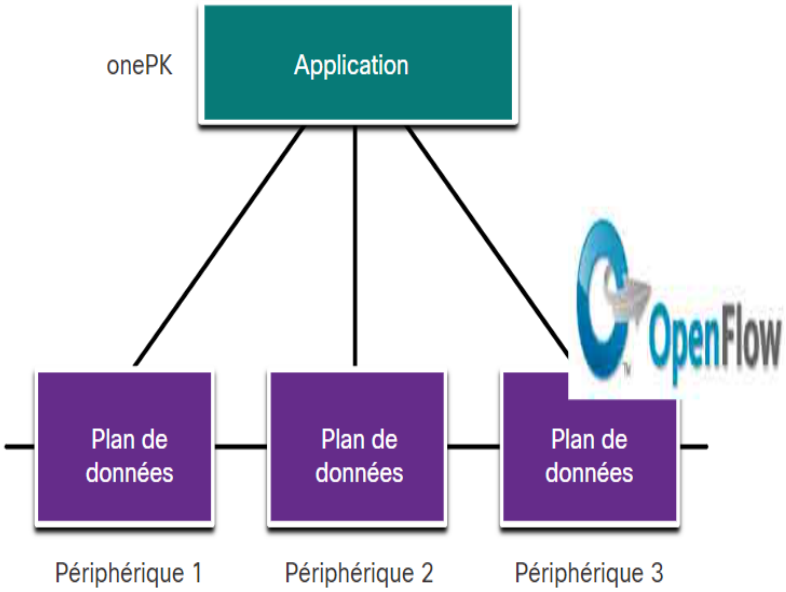
best-known SDN controllers

Controller	Organisation	Language	Features
NOX	Nicira	C++	the first openflow controller
POX	Nicira	Python	improve NOX performance
Ryu	NTT, OSRG group	Python	supports OpenStack
Floodlight	Big Switch	Java	tested with physical and virtual OpenFlow switches.
Opendaylight	Linux Foundation	Java	supports the OSGi Framework and the REST API

Types of SDN architecture

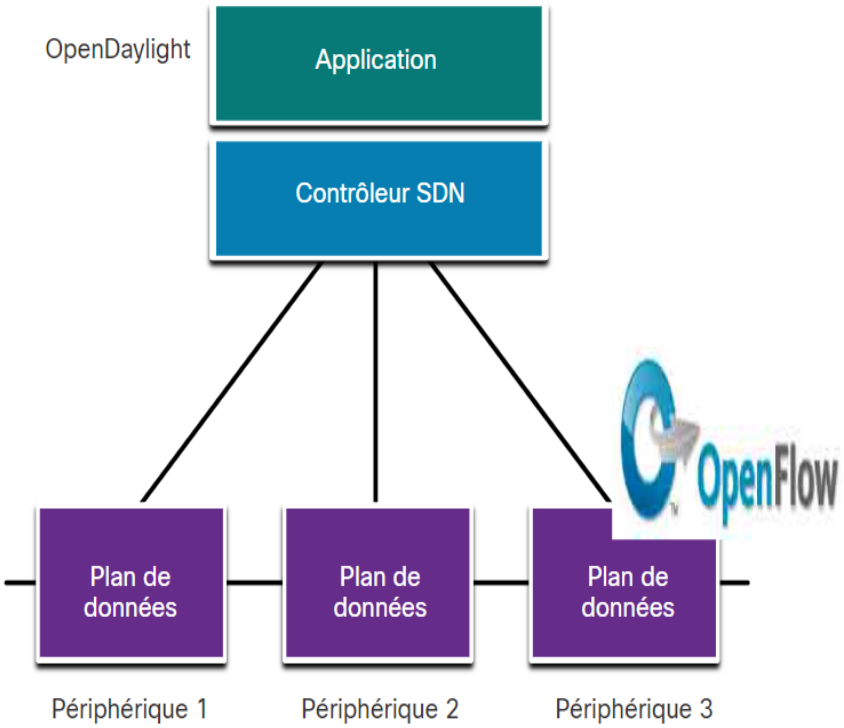
SDN based on devices

In this type of SDN, devices are programmable by applications running on the device itself or on a server on the network, as shown in the figure. Cisco OnePK is an example of device-based SDN. It allows programmers to develop C and Java applications with Python to integrate and interact with Cisco devices.



Types of SDN architecture

SDN based on controllers

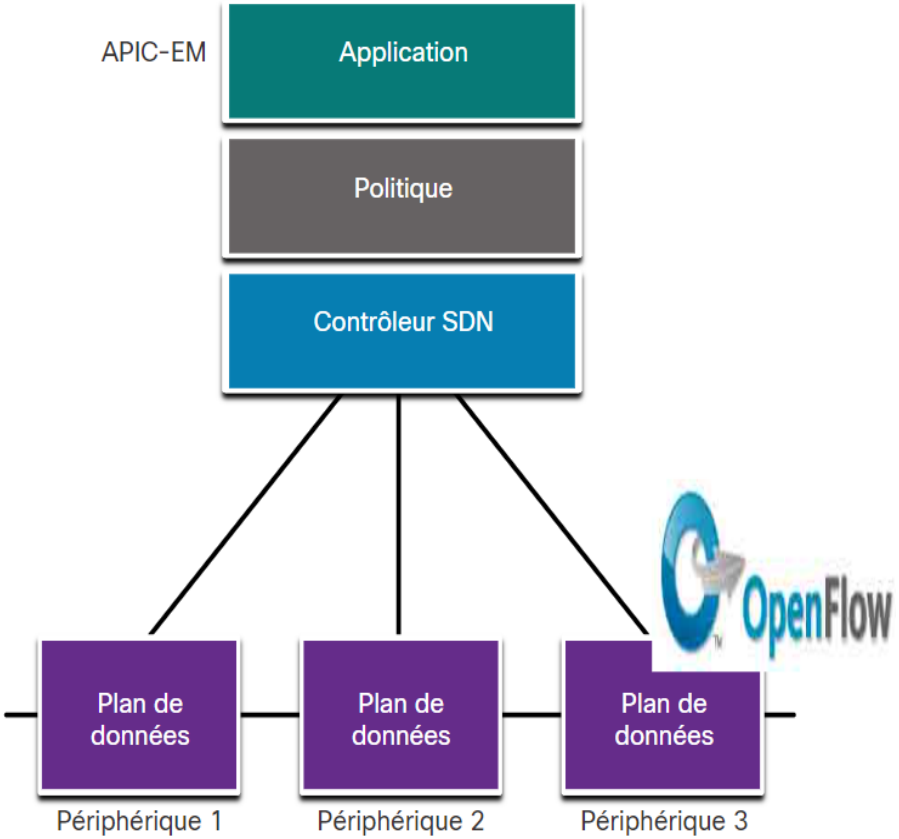


This type of SDN uses a centralised controller that knows all the devices on the network, as shown in the diagram. Applications can interact with the controller, which is responsible for managing devices and manipulating traffic flows on the network. The Cisco Open SDN controller is a commercial distribution of OpenDaylight.

Types of SDN architecture

Policy-based SDN

Policy-based SDN includes an additional policy layer that operates at a higher level of abstraction. It uses embedded applications that automate advanced configuration tasks through a guided workflow and an easy-to-use graphical user interface. No programming skills are required.



SDN CHALLENGES

Performance

SDN is a flow-based technique, and its performance is measured in terms of two metrics: the time taken to introduce a new flow into the switches (latency) and the number of flows the controller can process per second (throughput).

- *The use of multiple controllers is a more effective solution for improving the performance of SDN controllers.*
- *Solutions such as HyperFlow and Opendaylight deploy a logically centralised control plane, where several controllers are used, sharing loads and synchronising data between them, thereby improving performance and ensuring network consistency.*

SDN CHALLENGES

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.*

SDN CHALLENGES

Security

Centralising all network intelligence in a single controller can increase the vulnerability of the controller. An SDN controller represents the critical point of the network; if it is compromised or becomes unavailable, all aspects of the network will be damaged.

Access control solutions: AuthFlow is an authentication and access control mechanism based on host credentials, enabling access to be denied to unauthorised hosts.A number of solutions have been proposed to overcome the denial of service attack on SDN controllers or switch flow tables. These include limiting the flow requests sent to the control plane using a connection migration tool..

SDN CHALLENGES

Fiabilité

In the early deployments of SDN networks, which used a single centralised controller responsible for the entire network, problems arose with SDN controllers, which became single points of failure.

- *mechanisms for detecting failures in controllers*
- *method of pre-partitioning between controllers*

STANDARDISATION AND THE AFRICAN CONTEXT SDN

STANDARDIZATION

- **Standardisation of Control Plane and Data Plane Separation** : SDN networks separate the control plane from the data plane. Standards define how these two planes interact and communicate.
- **Standardisation of Programming Interfaces (APIs):** Open APIs are essential to enable developers to create applications for programmable networks.
- **Standards define these interfaces Standardisation of Telemetry and Data Analysis**: Programmable networks generate large amounts of data. Telemetry and data analysis standards enable this data to be collected, stored and analysed to improve network performance and security.

NORMALISATIONS ET CONTEXTE AFRICAIN SDN

NORMALISATION

- **Standardisation of Security Policy Management:** Security policy management, including network segmentation, access management and threat detection, is essential. Standards define best practice for this.

Note: *the standardisation challenges for programmable networks aim to create a coherent, secure and interoperable ecosystem for this rapidly evolving technology. This facilitates the adoption and expansion of programmable networks in various domains, from telecommunications to cloud computing and the Internet of Things (IoT).*

NORMALISATIONS ET CONTEXTE AFRICAIN SDN

AFRICAN CONTEXT

- **Connectivity and Internet Access :** Africa faces the challenge of Internet access, particularly in rural areas. Programmable networks, including SDN, can enable more efficient management of network resources, improving connectivity and reducing access costs.
- **Integration of Innovation:** Programmable networks foster innovation by enabling African developers and researchers to create new applications and services tailored to local needs.
- **Digital Inclusion:** Africa is aiming to include more citizens in the digital economy. Programmable networks can help achieve this goal by making digital services more accessible.

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.
- ✓ Software Defined Networking (SDN):Etat deL'art Ihssane Choukri, Mohammed Ouzzif, Khalid Bouragba

Thank you for your attention