## SDN history

### Network device evolution

Since early 1990 network device manufacturer made a lot of innovation in order to increase router speeds. They started from a router node in which everything was computed into the central CPU to reach a situation where the central CPU is less and less used due to a distributed architecture in which lots of action are done in “line cards”.



These progresses have been made thanks to the use of proprietary ASICs (Application-Specific Integrated Circuit), TCAM (Ternary Content-Addressable Memory) which have been designed to process data packets at high speed.

In early 2000, the Virtualization for x86 computers support has led to lots of innovation into systems domain. Compute virtualization and High-Speed network devices evolution have enabled the Cloud creation.

It appears it was not convenient to manage several isolated network devices having each their own configuration language. Following needs have emerged:

* Single point of configuration
* Configuration protocol standardization
* Network feature support on x86 servers
* Extensibility and ability to scale

### Early age of SDN

In Stanford University (US - CA) Clean Slate Research Projects program has been initiated in order to think about how to improve the Internet network architecture. ETHANE project was part of this program. Its purpose was to “Design network where connectivity is governed by high-level, global policy”. This project is generally known as the first implementation of SDN:

In 2008, a white paper has been proposed by ACM (Association for Computing Machinery) to design a new protocol (OpenFlow) to be able to program network devices from a network controller.

In 2011, ONF (Open Networking Foundation) has been created to promote SDN Architecture and OpenFlow protocols.

### SDN startups acquired by major networks or virtualization vendors

First companies working on SDN have been founded around 2010. Most of them have now been bought by main networks or virtualization solution vendors.

In 2007, Martin Casado, who was working on Ethane project has founded Nicira to provide solutions for network virtualization with SDN concept. Nicira has been aquired by vMware in 2012 to develop VMare NSX. In 2016, VMWare also bought PLUMGrid a SDN startup founded in 2013.

In 2010, BigSwitch networks has been founded: BigSwitch is proposing a SDN solution. In early 2020, BigSwitch has been acquired by Arista Networks.

In 2012, Cisco has created Insieme Networks, a spin-in start-up company working on SDN. In 2013, Cisco take back control on Insieme in order to develop its own SDN solution called ACI (Application Centric Infrastructure).

In early 2012, Contrail Systems Inc has been created and aquired at the end of the year by Juniper Networks.

In 2013, Alcatel Lucent has created Nuage Networks, a spin-in start-up company working on SDN. Nuage Networks is now an affiliate of Nokia.

References:

http://yuba.stanford.edu/cleanslate/research\_project\_ethane.php

http://yuba.stanford.edu/ethane/pubs.html

https://dl.acm.org/doi/10.1145/1355734.1355746

## SDN definition

SDN (**Software Define Networking**) is a network architecture model in which the network dataplane function has been physical split from configuration and control plane function.

Control and Configuration functions are gathered into a "SDN controller" which is controlling several SDN Network devices. This new architecture intends to provide a new way to configure the network using a centralized configuration and control point.



Several expectations are behind this new model:

- **cost reduction**: using standardized network nodes. The costly part of the network equipment (CPU) beeing moved and shared onto a central node.

- **openness**: using some standardized protocols like REST, OpenFlow, XMPP, NetConf

- **automation**: through the API interfaces provided by the SDN controller.

- **features rich**: with the ability of the SDN Controller to reprogram each controlled device using flow tables

## What is SDN?

There is no real accurate definition of SDN, but we generally consider that a SDN solution has to provide one to several of following characteristics:

* a network control and configuration plane split from the network dataplane.
* a centralized configuration and control plane (SDN controller)
* a simplified network node
* network programmability to provide network automation
* automatic provisioning (ZTP zero touch provisioning) of network nodes
* virtualization support and openness

Cf: <https://www.rfc-editor.org/rfc/rfc7426.txt>

## Traditional Network Planes

Router architecture is built according following planes:

* **Configuration** (and management) **plane**: used for network node configuration and supervision. Widely use protocols are CLI (Command Line Interface), SNMP (Simple Network Management Protocol) and NetConf.
* **Control plane**: used by network nodes to take packet forwarding decision. In traditional networks most widely used network control protocols are OSPF, ISIS and BGP for IP protocol and LDP; RSVP-TE for MPLS.
* **Forwarding** (or data or user) **plane**: This plane is responsible to perform data packet processing and forwarding. This forwarding plane is made of proprietary protocols and is specific to each network equipment vendor.



Two first planes (configuration and control) are located into router main processor card. The last one is located into the router line cards.

## SDN layer

SDN architecture is built according 3 layers:

* **Application Layer**: is containing all the application provided by the SDN solution. Generally a Web GUI dashboard is the first application provided to SDN users. Other very common applications are Network infrastructure interconnection interfaces allowing the SDN solution to be plugged to a Cloud Infrastructure or a Container orchestrator.



* **Control Layer**: is containing the SDN controller. This is the smartest part of a SDN solution. The SDN controller is made up of:
  + one or several Northbound interfaces that are used to interconnect SDN application with the SDN infrastructure. The most used northbound interface protocol is HTTP REST.
  + one or several Southbound interfaces that are used to control SDN network nodes. Most used southbound interface protocols are OpenFlow and XMPP.
  + the SDN engine, made up of SDN Control Logic and some databases.
* **Infrastructure Layer**: is containing the SDN network nodes. This is the working part of a SDN solution. SDN network nodes are either physical or virtual nodes. On each SDN node are located:
  + a SDN agent: which is handling the communication between each SDN network node and the SDN controller.
  + A flow/routing information table filled by the SDN Agent.
  + A forwarding plane engine

## changes (comparing with traditional networking)

In a traditional infrastructure, the route calculation is made on each individual router. Routing path is the result of routing information exchange, and of a distributed calculation.

Traditional networks are very robust but very hard to manage due to the high number of points to configure. Traditional network nodes are requiring expensive components are they are implementing high end routing protocols.

|  |  |
| --- | --- |
|  |  |

New Cloud infrastructures are requiring:

* a single configuration point
* the ability to distribute at a higher scale network elements, at least in each Cloud compute, and not only at the network infrastructure level.
* a simplified network node in order to be able to implement it into each compute node.

In order to get a single configuration point, a centralized network controller is proposed by the SDN Architecture. In order to be able to simplify network nodes, the smartest part has been moved onto a controller.

A southbound network protocol is the last piece needed to allow routing information between the SDN controller and each controlled element. A network infrastructure is allowing the communication between SDN controller and SDN network nodes, and data packet transfer between SDN nodes. This underlay network infrastructure is playing the same role that the local switch fabric is doing inside a standalone router between the control processor card and lines cards.

In a SDN infrastructure route calculation is done centrally onto the controller and distributed into each SDN network node. It makes the controller the weakest point of this new kind of infrastructure.

Lots of efforts are done by each SDN solution supplier to make this centralized point:

* highly resilient: using clustered architecture to build the controller
* highly scalable: using distributed compute and storage architectures

## underlay vs overlay

In SDN architecture, each network node is connected to a physical network infrastructure. This physical network which is providing connectivity between network nodes is called the underlay network infrastructure.

However, customer data packet collected by SDN nodes have to be able to cross transparently this underlay network infrastructure. Therefore, a packet encapsulation mechanism is needed in SDN networks.



Indeed, without such an encapsulation mechanism, traditional segmentation solutions (VLAN, VRF) would have to be provided by the physical infrastructure and implemented up to each SDN node, in order to provide an isolated transportation channel for each customer network connected to the SDN infrastructure.

Encapsulation protocols used in SDN networks have to provide:

* network segmentation: ability to build several different network connectivity between 2 SDN network nodes.
* ability to carry transparently Ethernet frames and IP packets
* ability to be carried over an IP connectivity

Several encapsulation protocols are used into SDN networks; they are:

* VxLAN
* Geneve
* STT
* NVGRE
* MPLS over GRE
* MPLS over UDP

These encapsulation protocols are providing Overlay connectivity which is required between customers workload connected to the SDN infrastructure.



Each SDN node is call a VTEP (Virtual Tunnel End Point) as it is starting and terminating the overlay tunnels.

## interfaces between layers

## SDN vs openstack

## NFV

## SDN controllers

## SDN controller reports

## SDN controller

## opendaylight

## southbound interface

## openflow

## OVSDB

## northbound interface

## other terms