SDN FS2022 - Zusammenfassung

Luzia Kündig

June 29, 2022

1 Introduction and Concepts

Traditional Networking Architecture is divided into planes, depending on the layer

	Control Plane	Data Plane	Management Plane
Layer 2	Spanning Tree Overlays (VLANs)	Forward *Ethernet Frames*	
Layer 3	Routing Protocols Overlays (MPLS)	Forward *IP Packets*	

This results in some drawbacks such as

- – Limited decision making "intelligence"
- – Difficult administration
- – Missing overall analytics

1.1 Vision of SDN

- Hardware: cheaper
- Software: features frequent releases, decoupled from hardware
- Functionality: driven by software and controller. Aiming for a programmable network
- Simplicity: from manual to automated, from box centric to network wide, from provisioning in months to provisioning in hours
- Innovations: from closed systems to open and programmable

Virtualization of Computing needs virtualization of Network!

1.2 SDN Devices

All Information from FIB to Config can be updated via API calls (support depending: RESTCONF, NETCONF, ...)

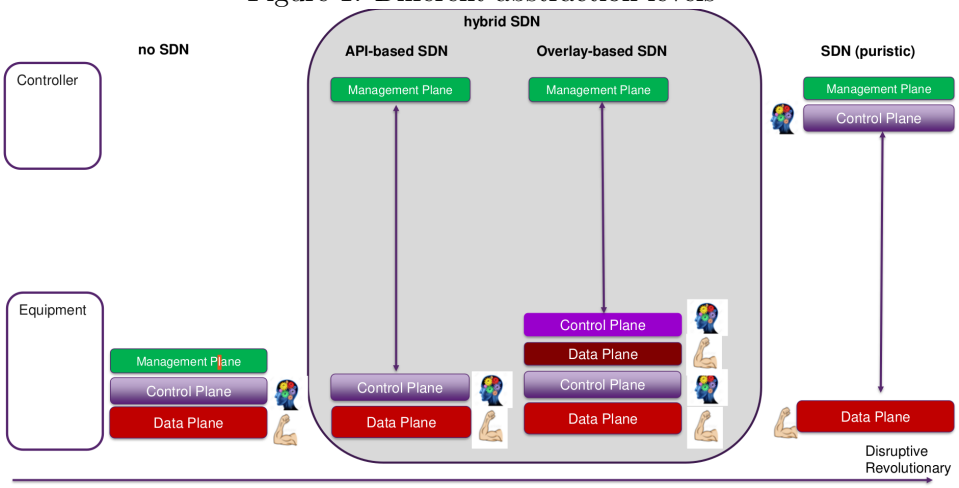
White box switches

- support OpenFlow 1.3
- Third Party OS Support

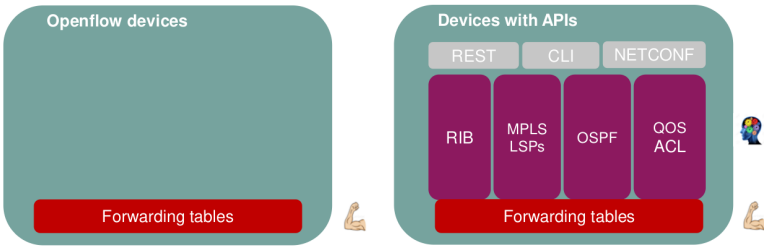
White box OS

- Open Compute Project OCP

Figure 1: Different abstraction levels



- Pica8
- Nvidia Cumulus Linuz
- opennetlinux.org
- fboss

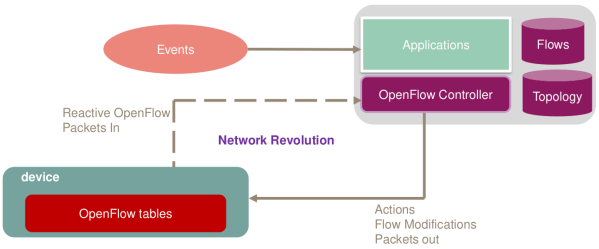


1.3 SDN Approaches

Different levels of abstraction can be applied to the topology, resulting in mainly three different SDN approaches.

1.3.1 Pure SDN

Academic approach. Only Data Plane on each device. Management and Control Plane centralized, resulting in full decoupling. OpenFlow Protocol distributes information, unknown Packets are sent to controller. Flow table is used for forwarding decisions.



Positive

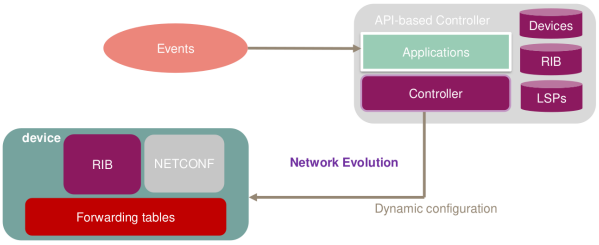
- Independent evolution and development
  - software control of the network can evolve independently from hardware
- control from high-level software program
  - debug/check behaviour more easily
  - testing/troubleshooting

Negative

- controller could be single point of failure
- no topology change without controller
- migration
- high risk

1.3.2 Hybrid SDN, API based

Data and Control Plane on each device, Management Plane centralized. Similar to snmp, ssh scripts.



Positive

- Faster provisioning of new customers and services
- Low impact in case of controller loss:
  - Provisioning delayed

Figure 2: Pure SDN schema example

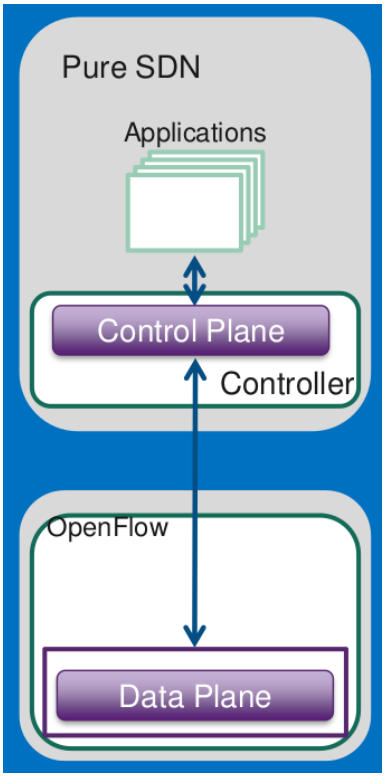
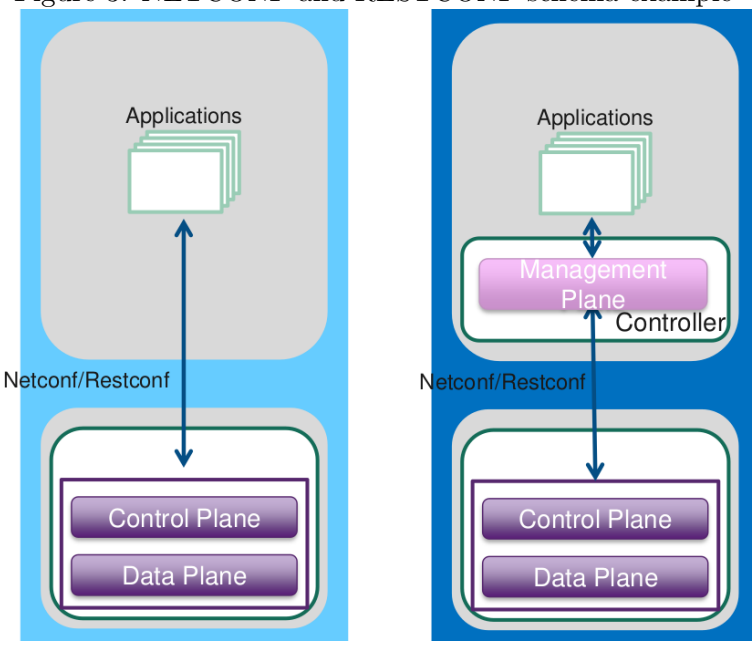


Figure 3: NETCONF and RESTCONF schema example



  - Visibility loss
  - Equivalent to any orchestration system failure
- Network partitioning: low impact
- Increased flexibility and speed

Neutral

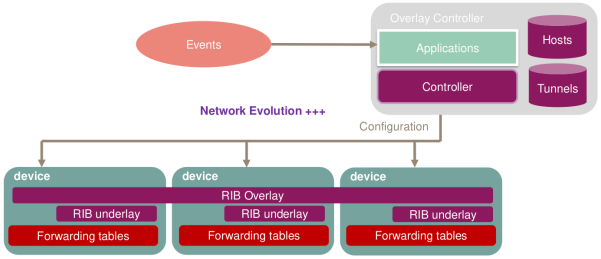
- Normal hardware cost
- No control plane change
- Transactional consistency important (all or nothing commands on devices)

Negative

- Static Management
- Not suited for multivendor environments
- software dependencies

1.3.3 Hybrid SDN, Overlay based

- Underlay Network: optimized, traditional Architecture
- Overlay Network: flexible, virtual network, centralized Management Plane
- Encapsulation necessary (VXLAN, NVGRE, IPSEC, ...)



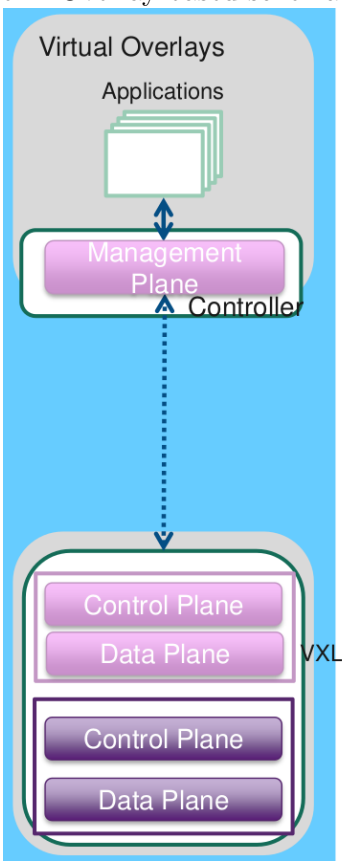
Positive

- Decoupling of services and network
  - Service provisioning in the edge elements
- No impact on the transport core

Negative

- overhead in
  - Encapsulation
  - Processing power
  - Complexity (additional control plane)
- Overlay-to-physical gateways
- End-to-end monitoring and troubleshooting

Figure 4: Overlay based schema example



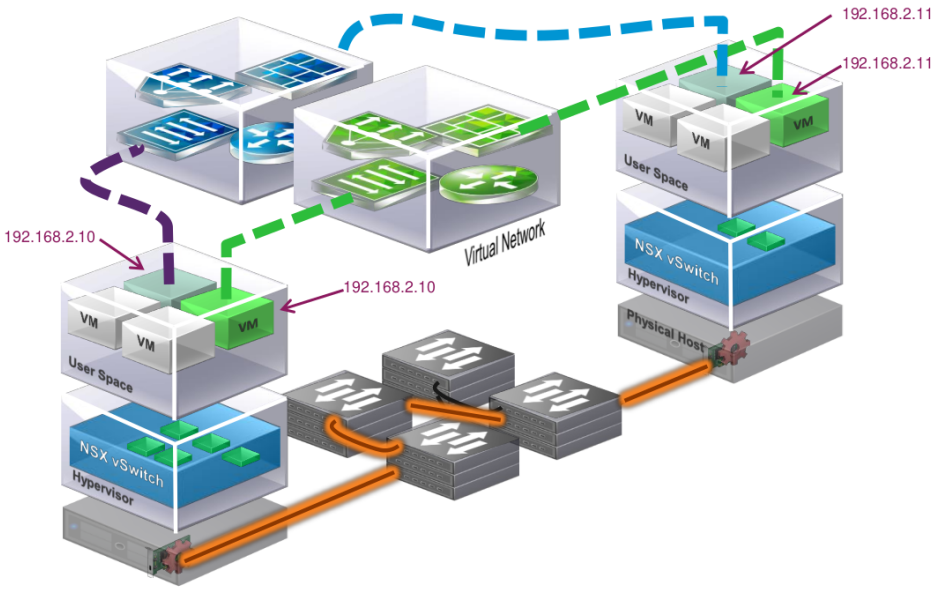


Figure 5: Datacenter example of an overlay solution

2 Segment Routing

RFC 8402: `https://datatracker.ietf.org/doc/html/rfc8402`

Segment Routing (SR) leverages the source routing paradigm. A node steers a packet through an ordered list of instructions, called "segments". A segment can represent any instruction, topological or service based. A segment can have a semantic local to an SR node or global within an SR domain. SR provides a mechanism that allows a flow to be restricted to a specific topological path, while maintaining per-flow state only at the ingress node(s) to the SR domain.

2.1 Source Routing

The entire path is calculated as a *Segment List* by the source router, or received by a PCE (Path Computation Element). This path defines either a route inside a topology or some network services (Networks Function Virtualization NFV).
Segment = an Instruction

- Shortest Path
- specific interface
- ...

Each node has one or several *Segment IDs*.

- MPLS Label or IPv6 Address
- Global Segment - Each node in the SR domain installs this instruction
- Local Segment - only the node that originates this segment installs the associated instruction.
  all nodes must know about other nodes' local segments in order to use them.

A packet header contains an **SID** as shortest path destination or **SID List** as defined path.
Every hop runs the instructions inside the packet header - there is no per-flow state information. An instruction can be one of the following three.''

- PUSH – insert segment(s) at the packet head and set first as active
- CONTINUE – active segment is not completed and remains active
- NEXT – active segment is completed, make next item in SID list active

2.2 IGP Extensions to support Segment Routing

The following segment types must be known to the IGP protocol in order to distribute SIDs.

2.2.1 Prefix Segment

Shortest path to any known network (prefix). Multi hop, equal cost.
Prefix SID is domain-wide unique, assigned manually to the loopback address of each node
Algorithm id specifies the method of choosing a path - default is 0, shortest path.
A Prefix Segment can be of two different types

- Node Segment
  Associated with a /32 prefix which is a node address.
  Sets **N-Flag** in Segment ID.

- Anycast Segment
  Associated with an anycast prefix, which routes to the geographically closest out of a group of hosts.
  N-Flag is unset!
  Macro-Engineering: can be used to steer traffic via specific region, or make it pass some router performing special network functions.
  Offers ECMP load balancing and high availability.

2.2.2 Adjacency Segment *(local)*

Unidirectional Adjacency, traffic is steered explicitly over an interface / link.
Overrides shortest path routing decisions.
SID list contains node prefix first, then Adjacency-ID.

- Layer-2 Adjacency can address one specific link inside a Link Aggregation Group (LAG).

- Group Adjacency

2.2.3 BGP Segments

- BGP Prefix Segment
  Global segment, associated with a BGP Prefix
  "steer traffic along the ECMP-aware BGP multi-path to the prefix associated with this segment"

- BGP Anycast Segment
  Traffic steering capabilities such as *"steer traffic via spine nodes in group A"*

- BGP Peer Segment
  Associated with BGP Peering sessions to specific neighbor
  Local segments that are signaled via BGP link-state address-family
  "steer traffic to the specific BGP peer node via ECMP multi-path towards that peer router"
  Overrides the traditional BGP mechanism

- BGP Peer Adjacency Segment
  "steer traffic to the specific BGP peer node via the specified interface towards that peer router"

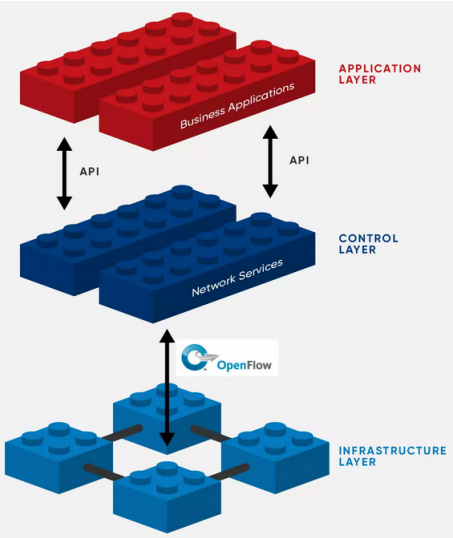
Combining segments can create any kind of end-to-end path.
Traffic steering only happens on source nodes to enable per-flow load balancing.
For **traffic engineering** a policy defines the path (SID-List) to be used.

HELLO	Sent by the switch, reply by the controller
FEATURE_REQUEST	Sent by controller, as supported OF capabilities
FEATURE_REPLY	Sent by switch to advertise

3 OpenFlow

Managed by the Open Networking Foundation.

Standard Southbound Protocol used between the SDN controller and the switch - *management only!*



OpenFlow operates as TCP Protocol (6644 / 6653) and can be secured by TLS using certificates.
Components of an OpenFlow Switch

- Flow Table(s)
- Group Table
- OpenFlow channel(s) to external controller

3.1 Controller

OpenFlow messages - for OF Channel setup between switch and controller
Controller manages *Flow Entries* in every switches flow tables (add, update, delete).

3.2 Flow Tables

Flow entry consists of

- Match fields

- Counter
- Instructions

Replaces traditional MAC/CAM table that stores hosts' hardware addresses. A flow entry is selected by IP packet matching fields, first matching entry is used ordered by priority.

- 39 fields possible to match on in OpenFlow 1.3, BUT must be supported by the hardware used
- usually in routing: most specific match

Instructions can be actions or modify pipeline processing. Possible actions are

- Forward on port
- Drop
- Flood
- Send to controller

If no match in any flow table is found: TABLE_MISS rule configuration: send to controller or drop.