SVEUČILIŠTE U ZAGREBU

Fakultet elektrotehnike i računarstva

Master Programme Computing

Advanced Architectures of Telecommunication Networks

Ac. year 2022/2023

Software Defined Networks and OpenFlow

Software-defined networking (SDN)

- Approach in networks that enables managing network services through abstraction of lower-level functionalities
  - Covers 3 problems in non-programmable networks:
    - Routing and constraints
    - Distributed services and segments
    - Configuration
- SDN is
  - Directly programmable: network control is programmable because it is decoupled from routing (forwarding) functions
  - Agile: administrator can dynamically adjust network-wide traffic flow to meet service needs
  - Centrally managed: network intelligence is logically centralized
  - Programmatically configured
  - Open standards-based and vendor-neutral

Network Operating System (NOS)

- NOS: distributed system that creates and maintains a network view
  - Think of abstraction of network in your Win/Mac/Linux machine!
- Communicates with forwarding elements
  - Get state information from forwarding elements
  - Communicates control directives to forwarding elements
- NOS in SDN?
  - NOS is placed in control layer
  - But also – physical hardware should be "programmable"
  - So, in SDN, NOS is an abstraction of lower physical networking devices

→ https://www.networkworld.com/article/2162773/the-return-of-the-network-operating-system--nos-.html

SDN definition – a bit more details

- Different definitions of SDN are used today
  - general: "everything that includes software is SDN"

- The term SDN refers to a network architecture in which:
  - the control and data planes are separated
  - packet forwarding (routing) is based on an extensible set of parameters to describe packet flows
  - the stream can be determined by different values from the packet header (and not just, e.g., source and destination IP addresses)
  - the control logic has been moved to a separate entity – SDN Controller
  - SDNC provides an abstract view of the network and its state
  - it is "programmable" for/by applications
  - adaptive control of network services (e.g., routing)

SDN: basic concept 1/2

- Separate control plane functions from network devices into a centralized control device
  - An SDN controller (SDNC) and "network operating system" (NOS)
    - Possible confusion!
  - SDNC centralization is logical, not physical!
  - There may be multiple (physical or virtual) SDNC instances in the network
    - distributed control plane

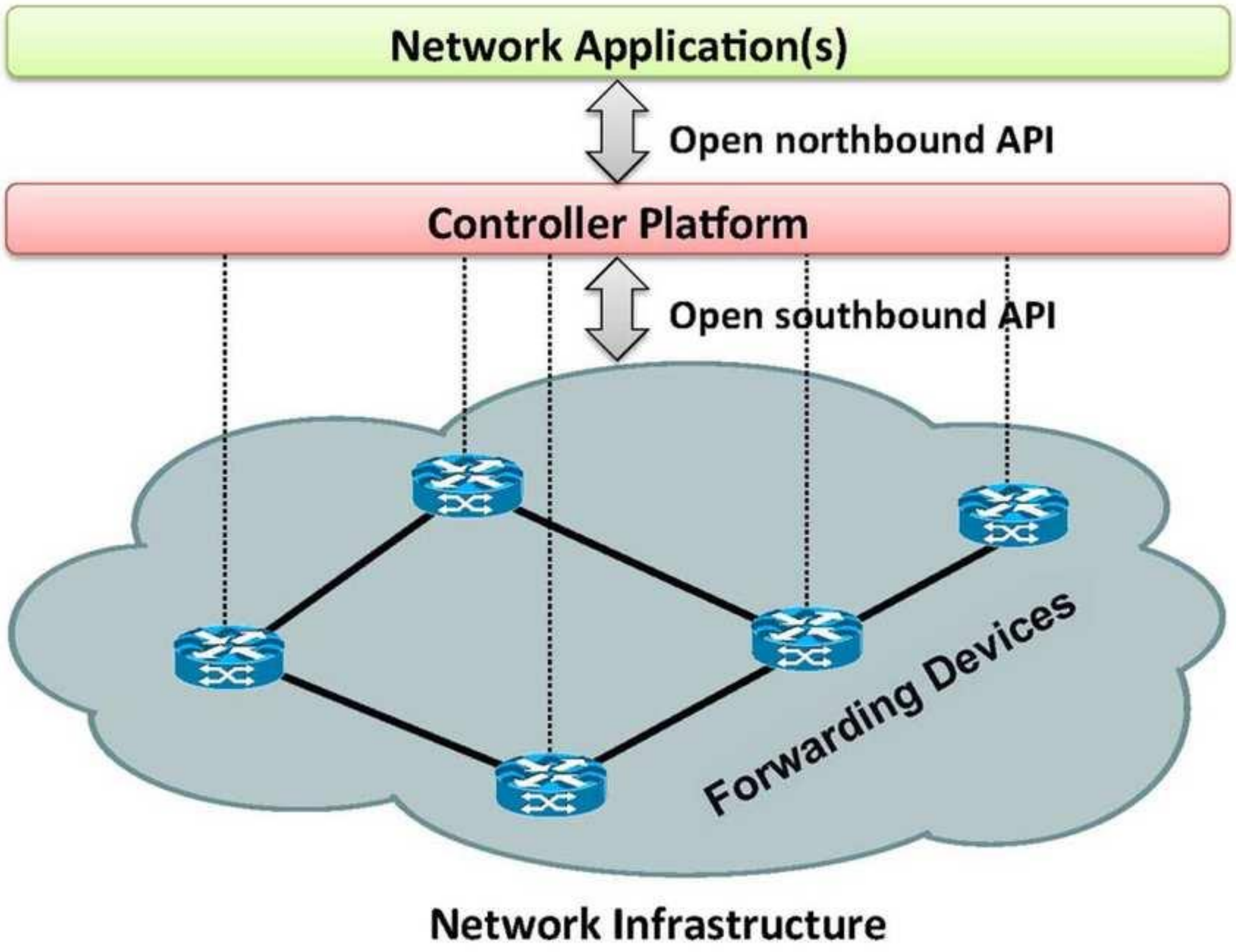
- Network devices retain only the data plane functions
  - Simpler network elements (NE) for packet forwarding
  - Decisions on how to forward are made by the SDNC

SDN: basic concept 2/2

- Plane separation requires a well-defined software interface between the network elements and the control device
  - With such an interface, SDNC can have direct control over the state of NEs
  - One such interface is the one based on the OpenFlow specification

- This separation of functions brings the necessary flexibility and facilitates technological evolution
- Communication network management challenges are now divided between adaptable entities

Simplified view of SDN



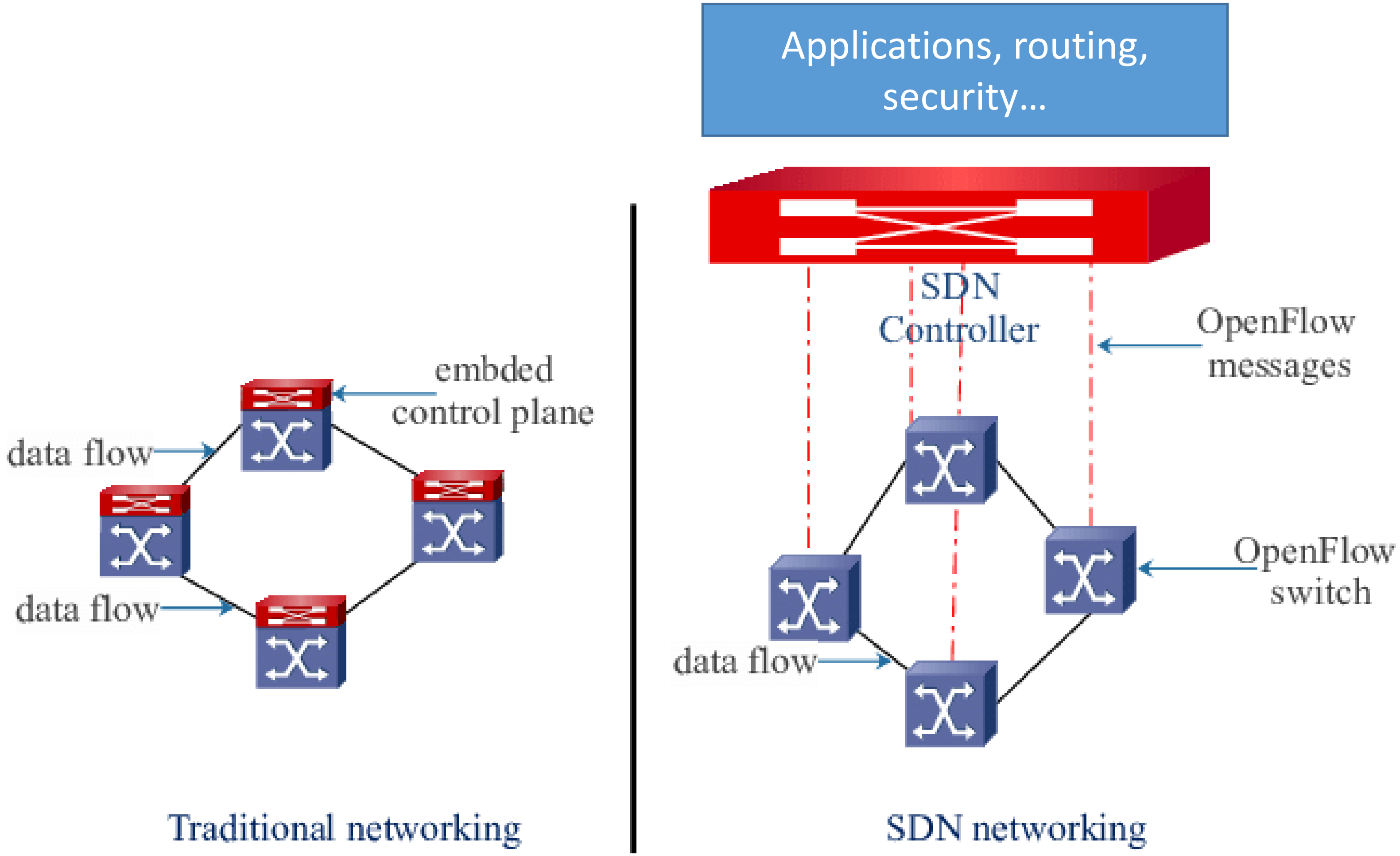
Application plane

Control plane

Data plane (infrastructure)

https://www.researchgate.net/publication/317999618_Bandwidth_and_Accuracy-Aware_State_Estimation_for_Smart_Grids_Using_Software_Defined_Networks

SDN vs traditional networks



Specifies behavior

Compiles to topology

NOS: Transmits to switches

https://www.researchgate.net/publication/324941281_Energy-Aware_Routing_in_Carrier-Grade_Ethernet_using_SDN_Approach

SDN architecture components 1/2

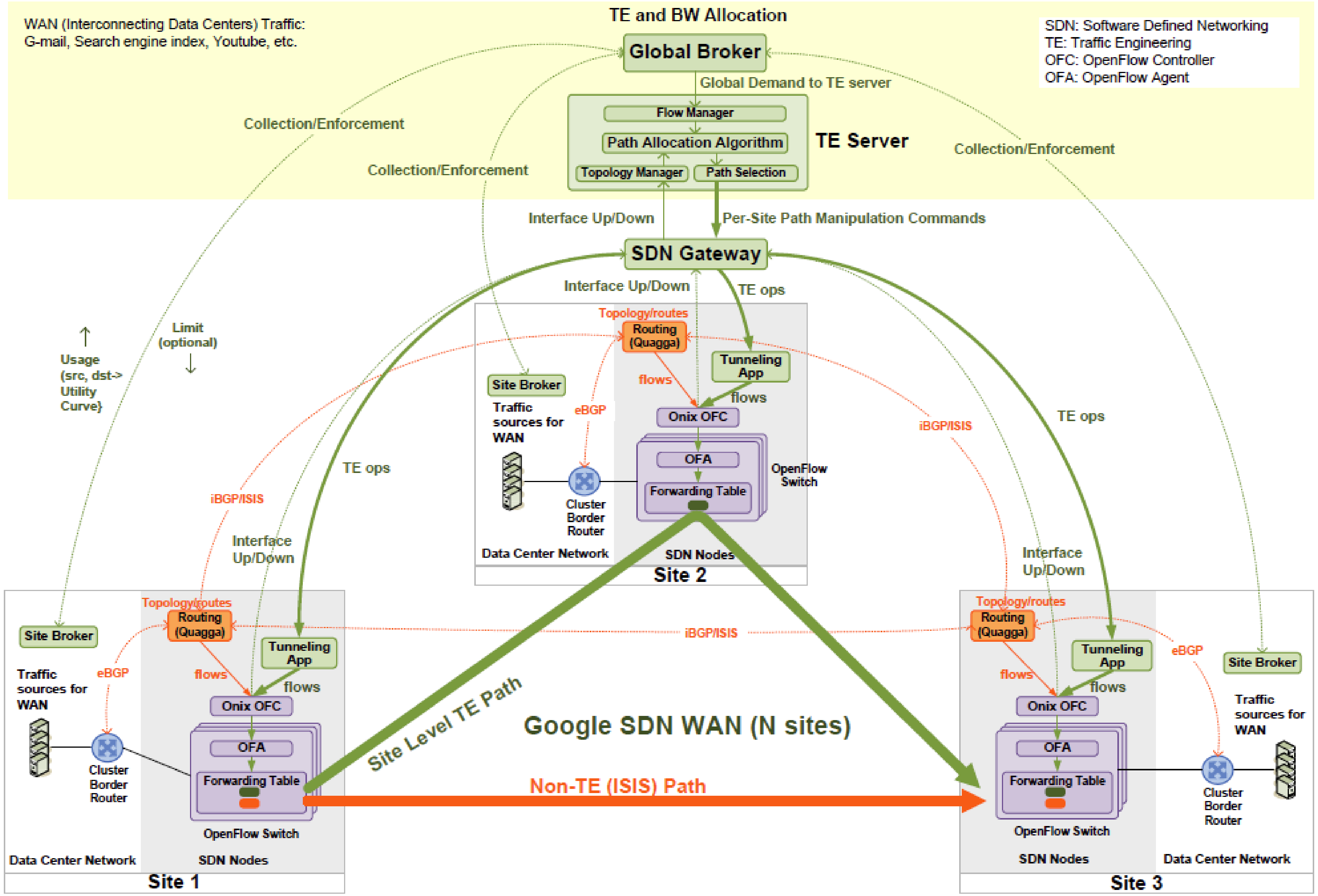
- Infrastructure (data) plane
  - Includes network elements (NEs), which allow (wired or wireless) connectivity between end computers
  - NEs can be hardware or software
  - NEs support a basic set of elementary packet operations (e.g., forward packet to SDNC or reject packet)
  - Supported types of operations depend on the performance of the Southbound interface

- Control plane
  - Includes control devices (SDNCs) that take care of network configuration ("control logic")
  - SDNC monitors the state of the network, decides on the method of packet forwarding and programs the corresponding forwarding rules in NEs
  - "Programming" is done through the Southbound interface

SDN architecture components 2/2

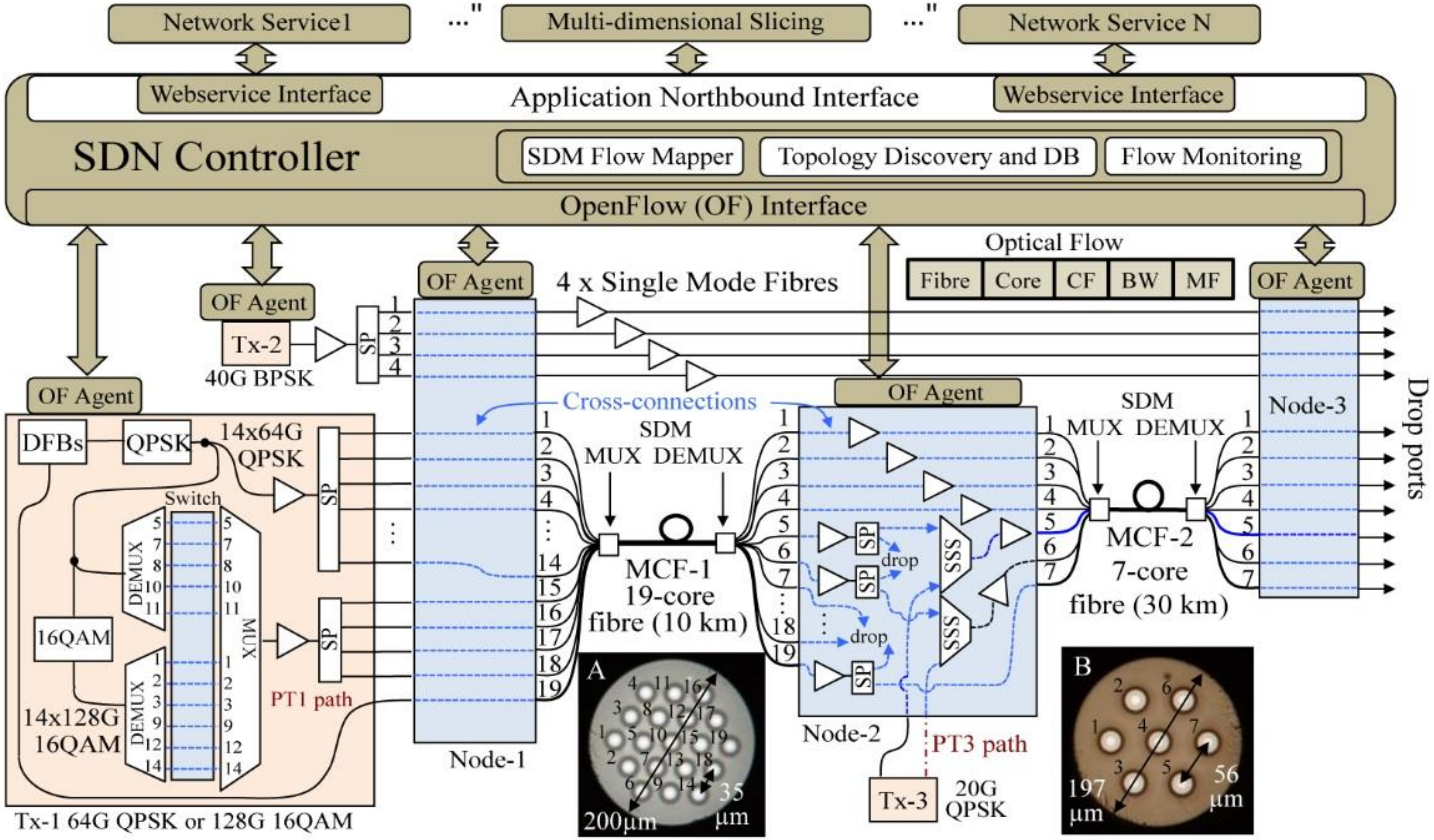
- Southbound interfaces
  - determine the mode of "interaction" between SDNCs and NEs
  - prescribe a basic set of operations on packages that NEs perform
  - also define the communication protocol between SDNCs and NEs
- Application plane
  - Includes performance of various network applications and services (e.g., routing, traffic load matching, firewall)
- Northbound interfaces
  - They are exposed by SDNCs to allow applications to specify requirements (e.g., on network performance)
  - They abstract the details of the performance of the Southbound interface and the infrastructure layer (i.e., the network resources it offers)

Example: Google's SDN WAN



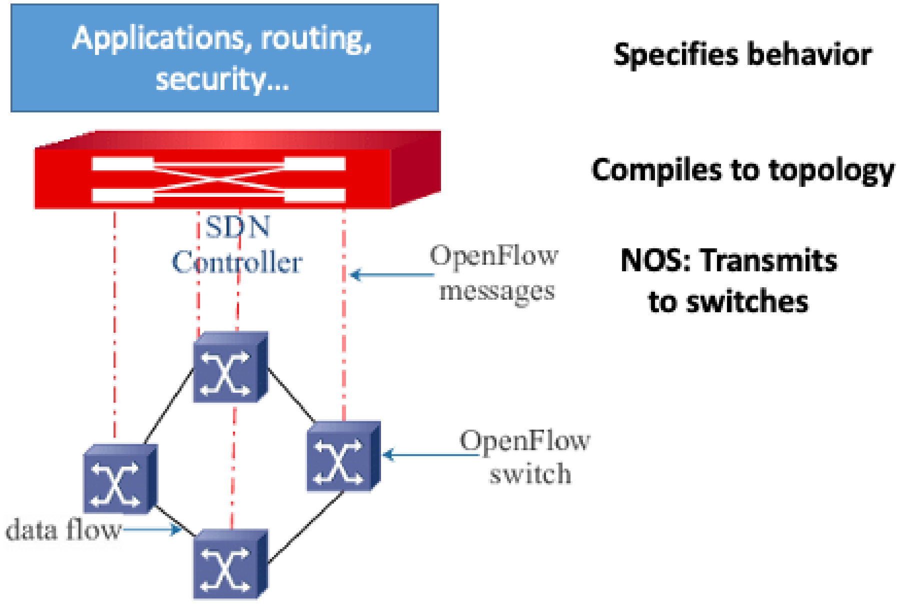
https://www.netmanias.com/en/post/oneshot/5939/google-network-architecture-sdn-nfv/google-sdn-wan-architecture

Example: SDN and optical infrastructure



How does programming work?

- Applications specify behavior, e.g.:
  - Access Control List implementation
  - Routing – which flow/packet goes where?
  - QoS assurance – traffic engineering
- Virtualization layer "compiles" these requirements
  - Produces suitable configuration of actual network devices
- NOS then transmits these settings to physical boxes



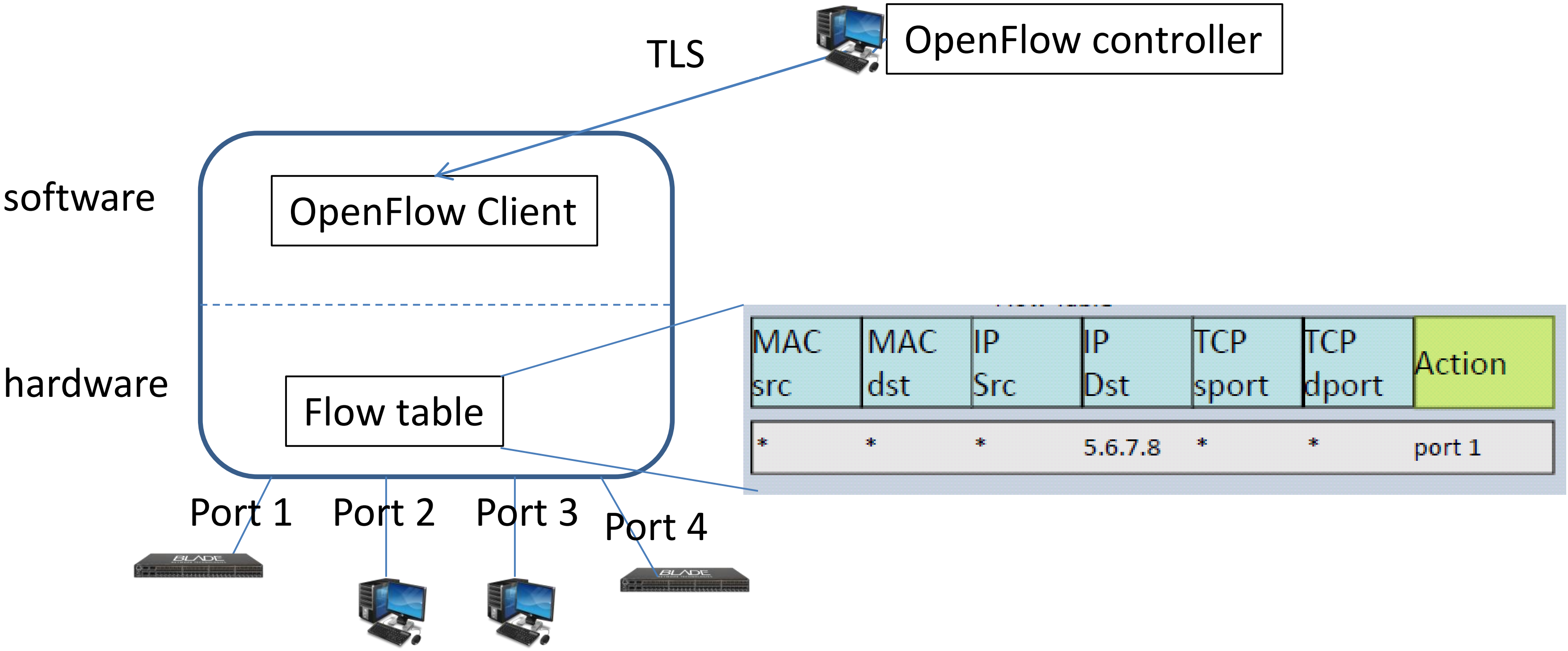
OpenFlow - basics

- OpenFlow is a communication protocol
- Defines the communication between SDN controller and the network device/agent
  - Network device/agent (typically called "switch") needs to be OpenFlow compliant
  - We use term "SDN switch" but in SDN this is really any network device capable of using OpenFlow protocol and not the device from OSI layer 2
- SDN controller takes the information from the applications and converts them into flow entries, which are fed to the switch via OF
- It can also be used for monitoring switch and port statistics in network management
- It uses TCP protocol, port 6653 (from controller to the switch)

OpenFlow – how switching is controlled

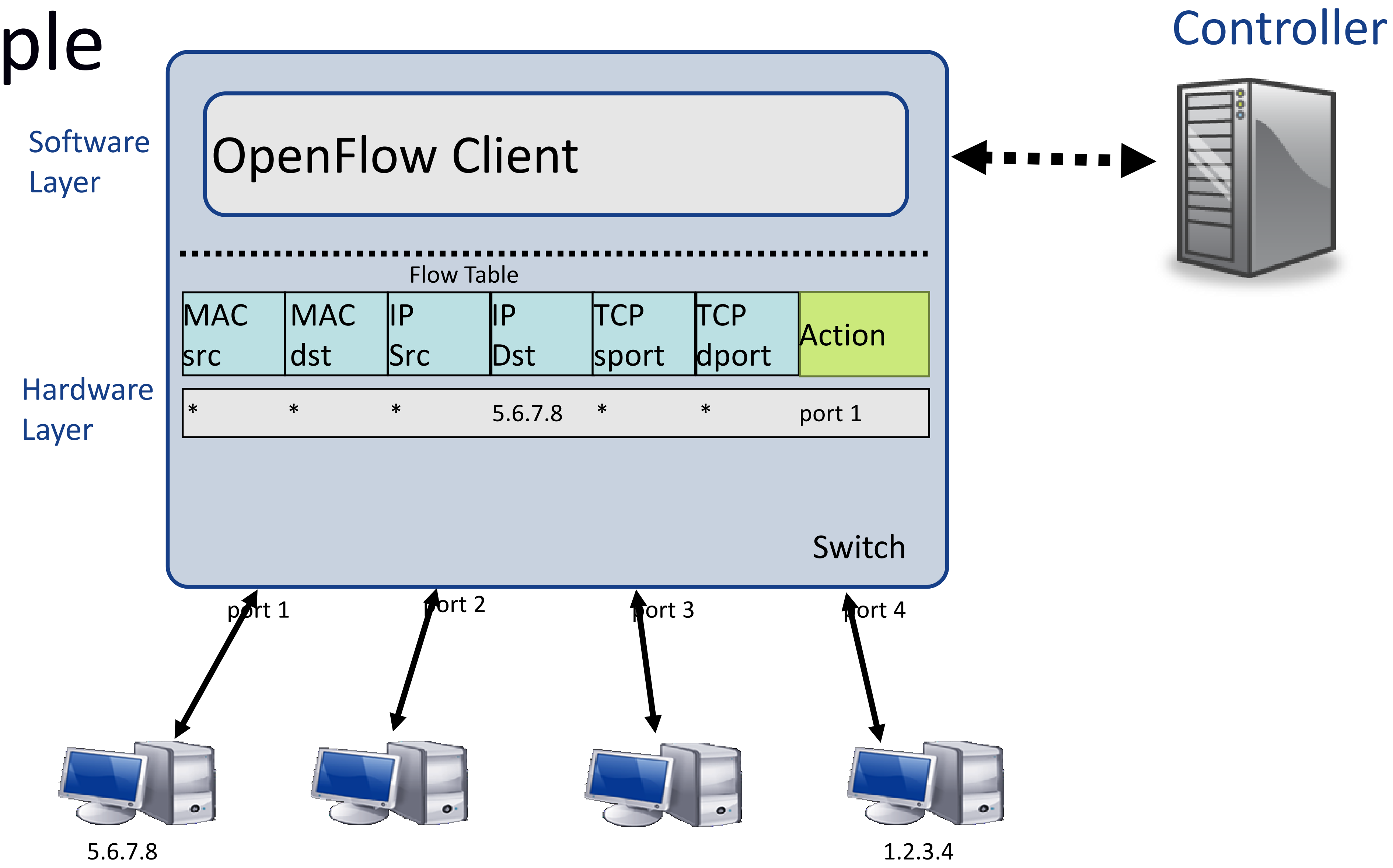
- An OpenFlow switch has an internal flow table (similar to L2 switch)
  - If a packet matches an entry in the flow table, perform the actions according to the flow table.
  - If a packet does not match any entry in the flow table, "ask" OpenFlow controller
    - the controller will know what to do with such packet
    - the controller will then respond to the switch, telling how to handle such a packet – from then, the switch "knows" how to handle these packets when they arrive again
    - for each flow, ideally the controller will be queried once.
- OpenFlow defines the standard interface to add and remove flow entries in the table

OpenFlow switch



https://www.cs.fsu.edu/~xyuan/cis5930/

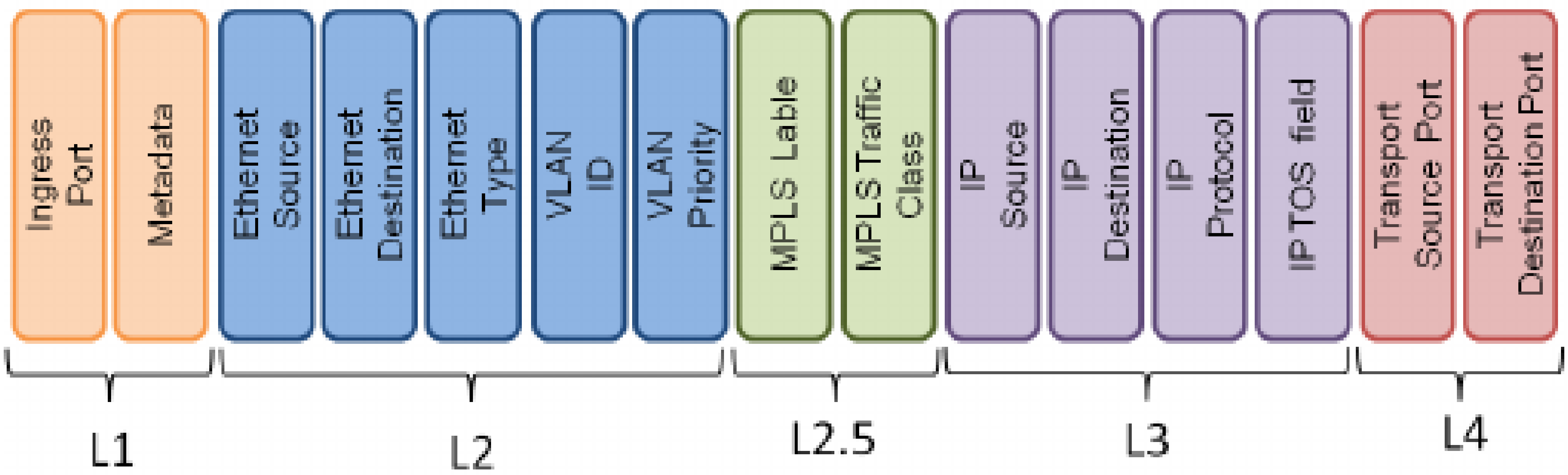
Example



MAC src	MAC dst	IP Src	IP Dst	TCP sport	TCP dport	Action
*	*	*	5.6.7.8	*	*	port 1

https://www.cs.fsu.edu/~xyuan/cis5930/

Flow identification fields



Layers (Lx) correspond to OSI stack!

https://www.researchgate.net/publication/261060581_OpenQoS_An_OpenFlow_controller_design_for_multimedia_delivery_with_end-to-end_Quality_of_Service_over_Software-Defined_Networks

Flow table entries and actions

Rule	Action	Stats

Packet + byte counters

1. Forward packet to zero or more ports
2. Encapsulate and forward to controller
3. Send to normal processing pipeline
4. Modify Fields
5. Any extensions you add!

Switch Port	VLAN ID	VLAN pcp	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP ToS	IP Prot	L4 sport	L4 dport

https://www.cs.fsu.edu/~xyuan/cis5930/

Examples 1/2

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

https://www.cs.fsu.edu/~xyuan/cis5930/

Examples 2/2

Routing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

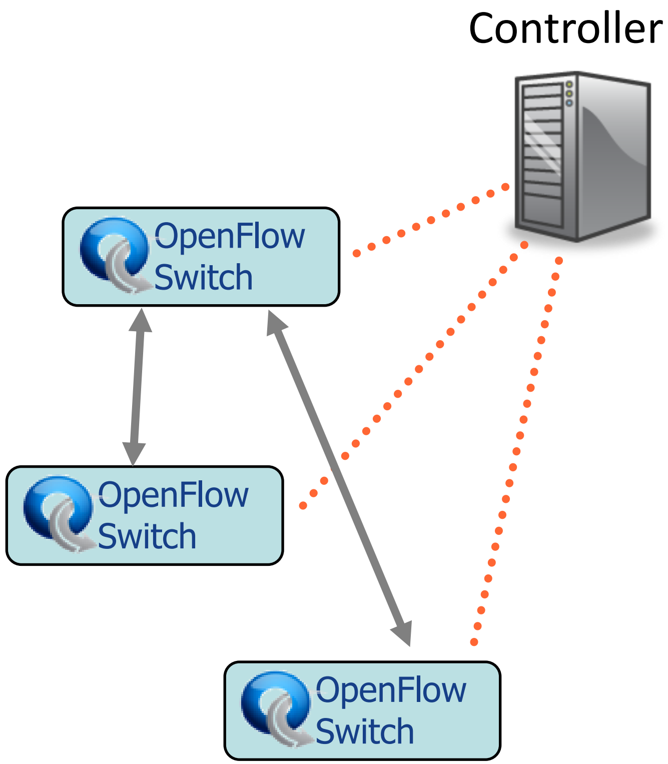
VLAN Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

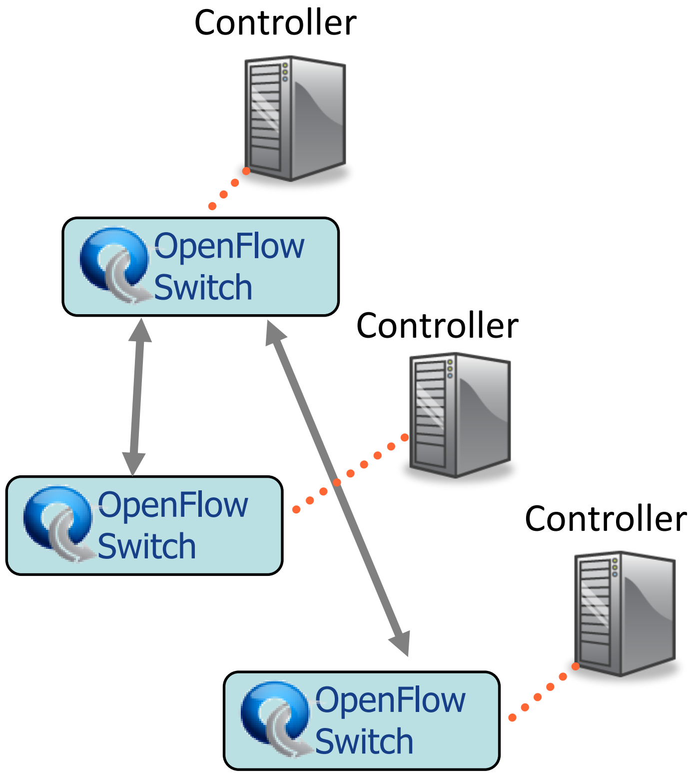
https://www.cs.fsu.edu/~xyuan/cis5930/

One or more controllers?

Centralized Control



Distributed Control



https://www.cs.fsu.edu/~xyuan/cis5930/

OpenFlow Protocol Messages

- Controller-to-switch: from the controller to manage or inspect the switch state
  - Features, config, modify state, read state, packet-out, etc.
- Asynchronous: send from switch without controller soliciting
  - Packet-in, flow removed/expired, port status, error, etc.
- Symmetric: symmetric messages without solicitation in either direction
  - Hello, Echo, etc.

Literature

- Course on SDN: https://www.cs.fsu.edu/~xyuan/cis5930/, Florida State University
- About OpenFlow: https://www.section.io/engineering-education/openflow-sdn/
- Open Networking Foundation: https://opennetworking.org/
- SDN tutorial: https://www.clear.rice.edu/comp529/www/papers/tutorial_4.pdf