SDN

Software Defined Networks

Harald Gustafsson

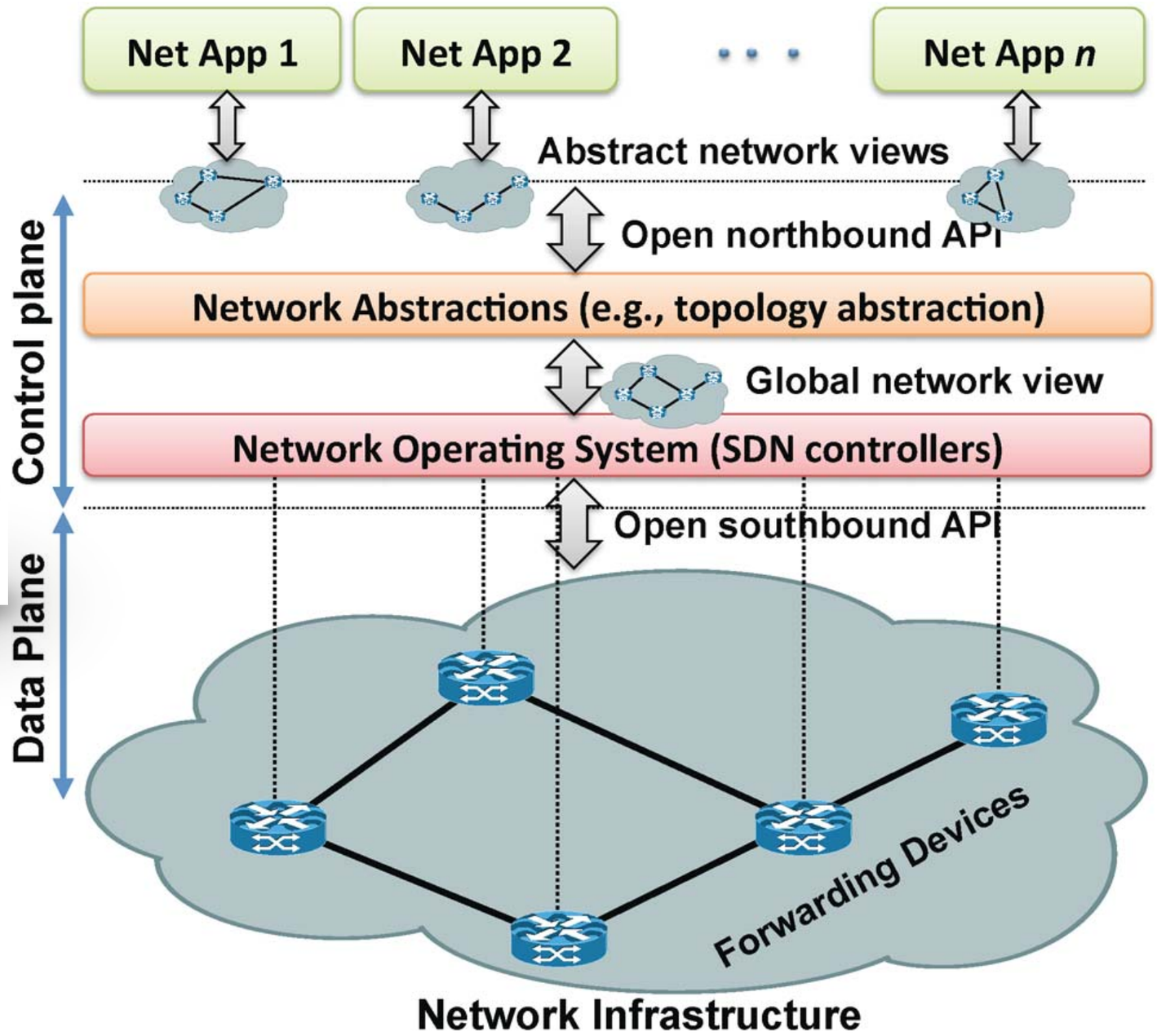
REFERENCES

- "Software-Defined Networking: A Comprehensive Survey", Kreutz et al, proceedings of the IEEE, vol. 103, 2015, pp 14-76.
- "B4: Experience with a globally-deployed software defined WAN", Jain et al, ACM SIGCOMM Computer Communication Review, vol. 43, 2013, pp 3-14

SDN analogy

VS

SDN concept



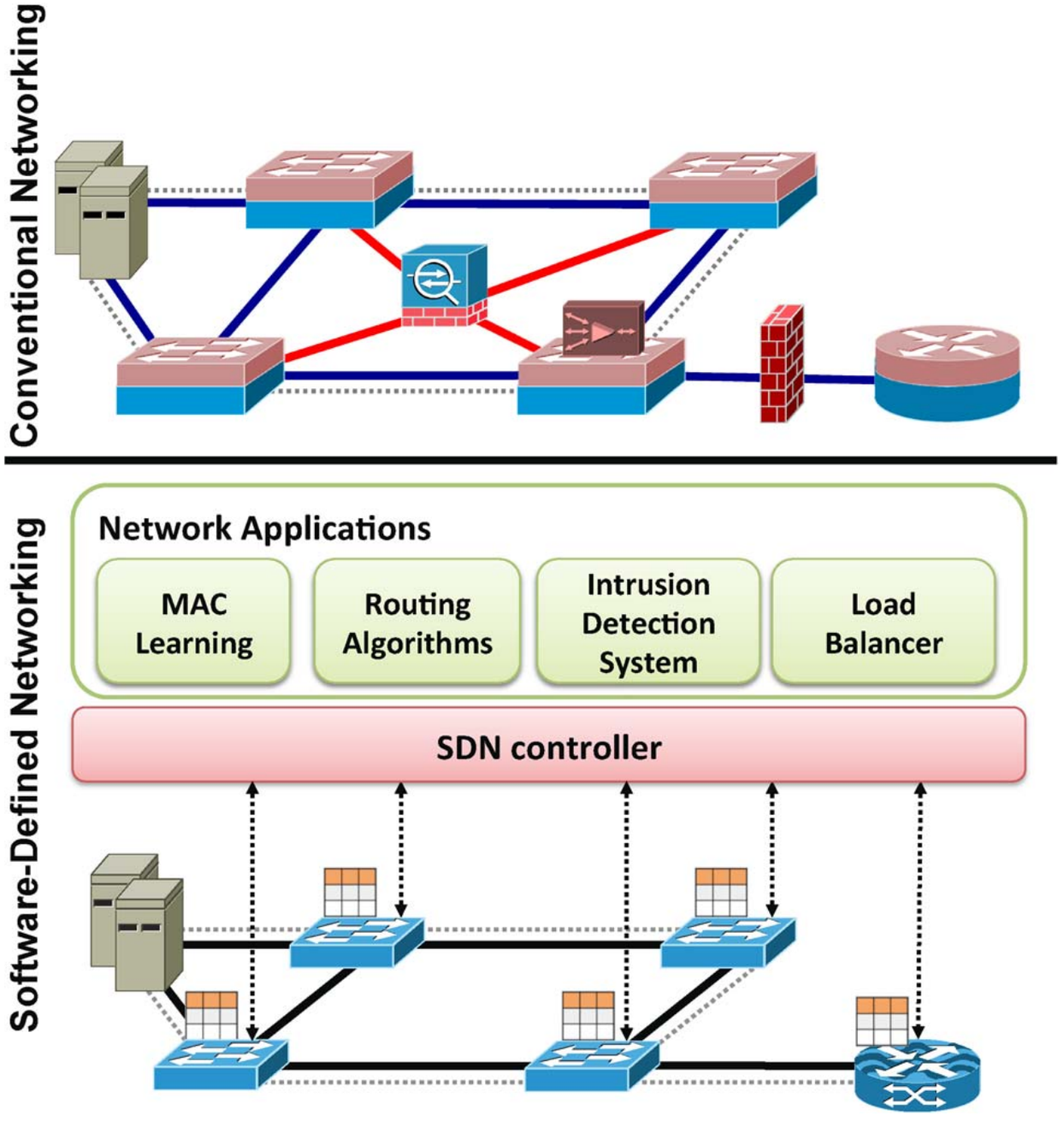
"User space"

"POSIX"

"Device driver"

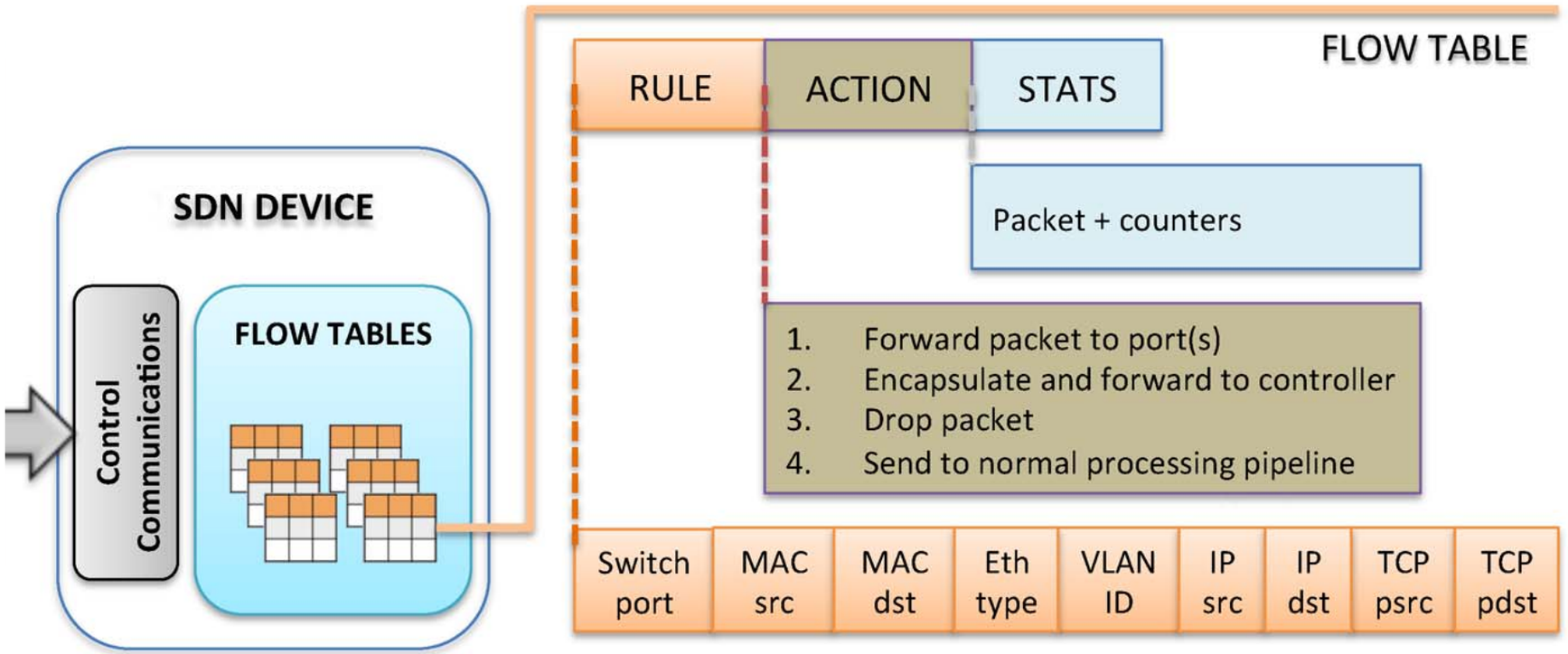
OpenFlow

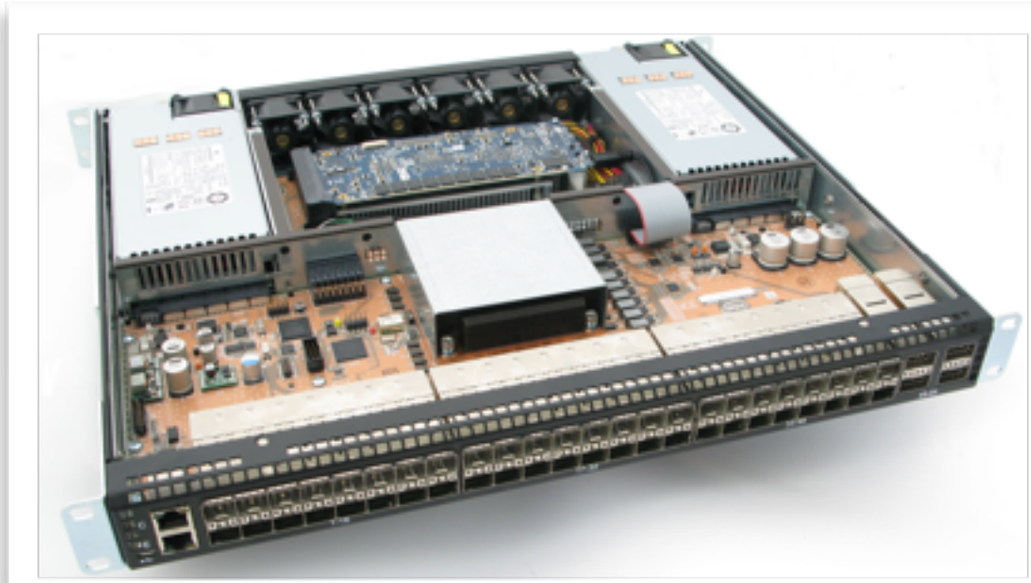
Example



- New features introduced as:
  - new box
    vs
  - new app

Forwarding device (switch)





- Run in order thru flow tables
- Match on bitmasked fields (flow)
- Do action based on flow
- Collect statistics

OpenFlow Protocol

- Three information sources for controller from forwarding device

    1. Event-based messages when a link or port change is triggered.

    2. Flow statistics are generated by the forwarding devices and collected by the controller.

    3. Packet-in messages when no flow rule or explicit ‘‘send to controller’’ action.

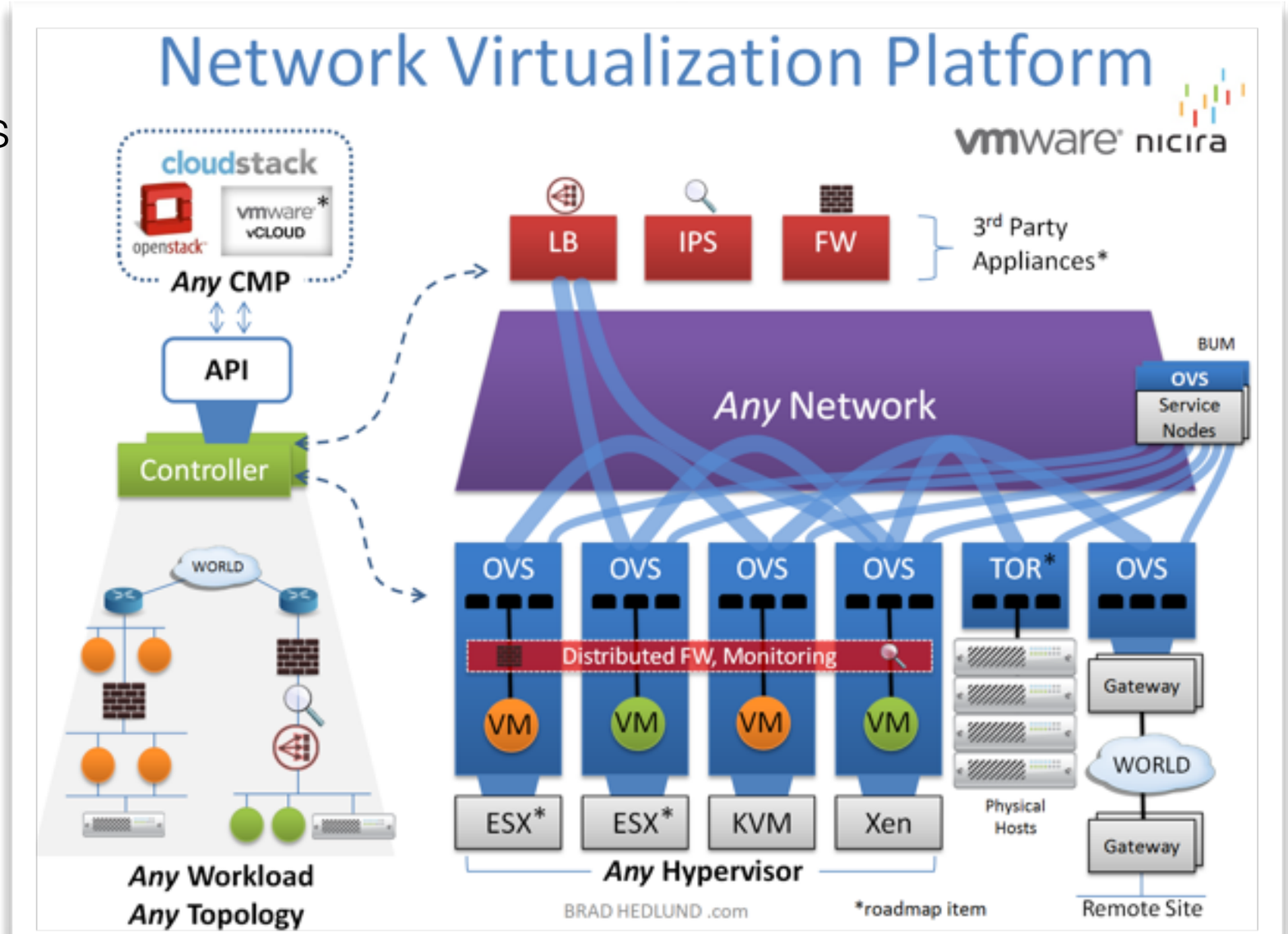
- Controller install flow rules in the forwarding device flow tables

Network Operating System

- The North-bound API (“POSIX”)
  - Still many contending APIs
  - Some pure SDN e.g. NOX, POX
  - Some legacy + SDN e.g. OpenDaylight
- Handles
  - Distributed/multi-threaded controllers
  - Conflicts between SDN-apps e.g. priority between security and routing flow rules
  - Support of several south-bound APIs
  - Simplify standard functionality e.g. collect statistics, topology, notifications, device mgmt, shortest path fwd, security mechanisms.

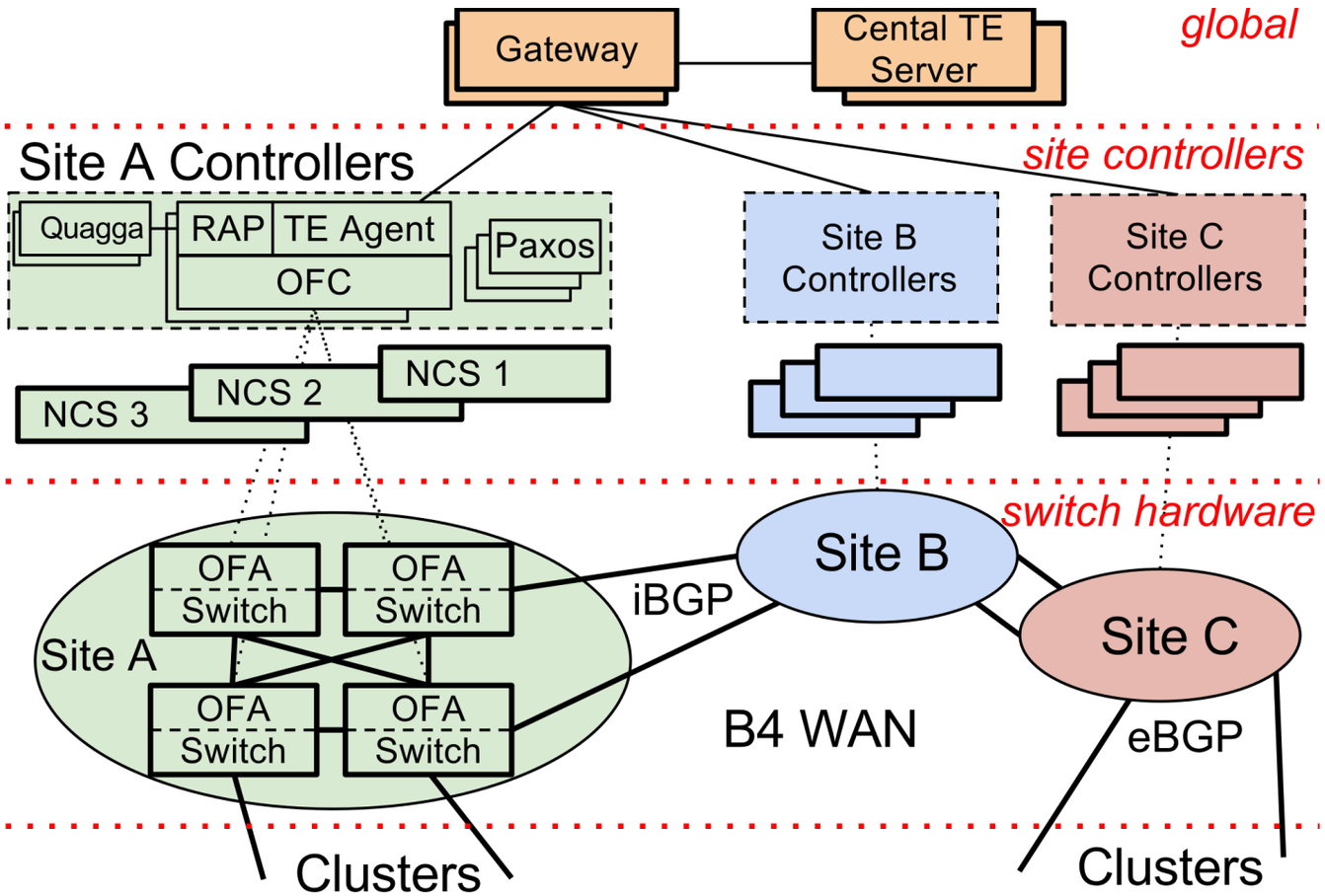
Cloud SDN - Intra DC

- Tenant specific topologies address and control function virtualization
- Forwarding devices (FD) in hypervisor
- Overlay network between hypervisors
  - Pseudo-SDN since physical switches not SDN
- Allow “moving” FD with VM



Cloud SDN - WAN (B4)

- Why:
  - Increased utilization from 30% to 70%-100%,
  - Traffic shaping e.g. user data prioritized over remote storage,
    - Elastic bandwidth needs
    - Failure handling support
- Routing and Traffic Engineering (TE) separate SDN applications
  - Allows falling back on standard protocols when TE fails
- TE use QoS levels for apps corresponding to fair-share BW – apps cooperate to allow shallow buffers
- Central TE makes multi-site-links QoS possible



OF: OpenFlow
OFA: OF Agent abstract FD
OFC: OF Controller
NCS: Network Control Server
Paxos: Leader election of site controller
Gateway: abstract cluster of controllers
Quagga, RAP: Routing software

SDN next

- Commercial switches only handles about 500 flow changes per second – needs improvement
- Move stateful local actions to switch, e.g. learning switch, threshold level rules, etc
- Reduce RTT latency between FD and controller
- Interoperability between controller applications e.g. Statesman automatically resolve conflicts
- Utilize high availability and scalability knowledge when designing distributed controllers
- Simplify fast fail-over flow rules
- Introduce hierarchal switches doing some of the controlling