BITS Pilani
Pilani Campus

Network Fundamentals for Cloud

Nishit Narang
WILPD-CSIS

innovate achieve lead

BITS Pilani
Pilani Campus

CC ZG503: Network Fundamentals for Cloud

Lecture No. 7: SDN and NFV

innovate achieve lead

BITS Pilani
Pilani Campus

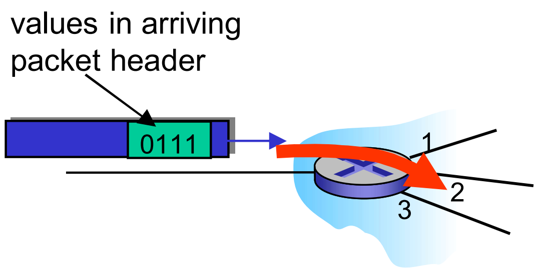
Software Defined Networking (SDN)

Some Slides Taken and Adapted from:
Computer Networking: A Top-Down Approach, 8th edition, Jim Kurose, Keith Ross, Pearson, 2020

Network layer: data plane, control plane

Data plane:

- *local*, per-router function
- determines how datagram arriving on router input port is forwarded to router output port



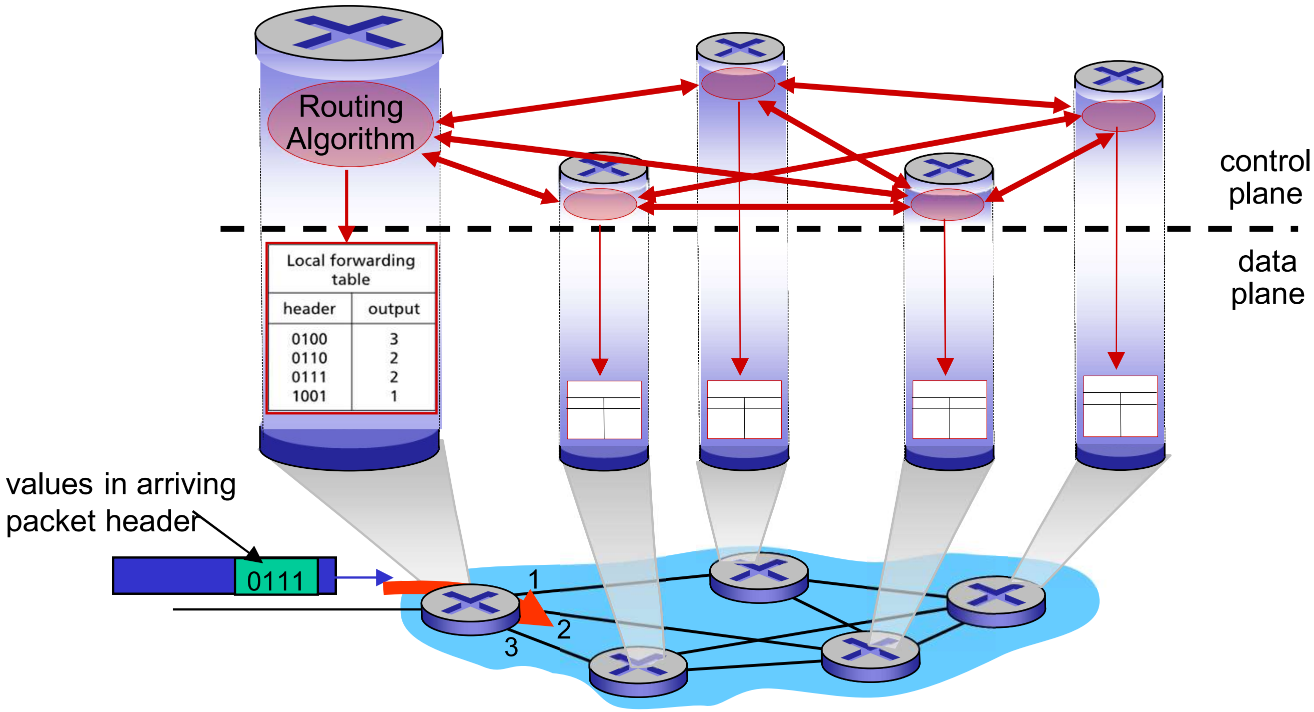
Control plane

- *network-wide* logic
- determines how datagram is routed among routers along end-end path from source host to destination host

- two control-plane approaches:
  - *traditional routing algorithms:* implemented in routers
  - *software-defined networking (SDN)*: implemented in (remote) servers

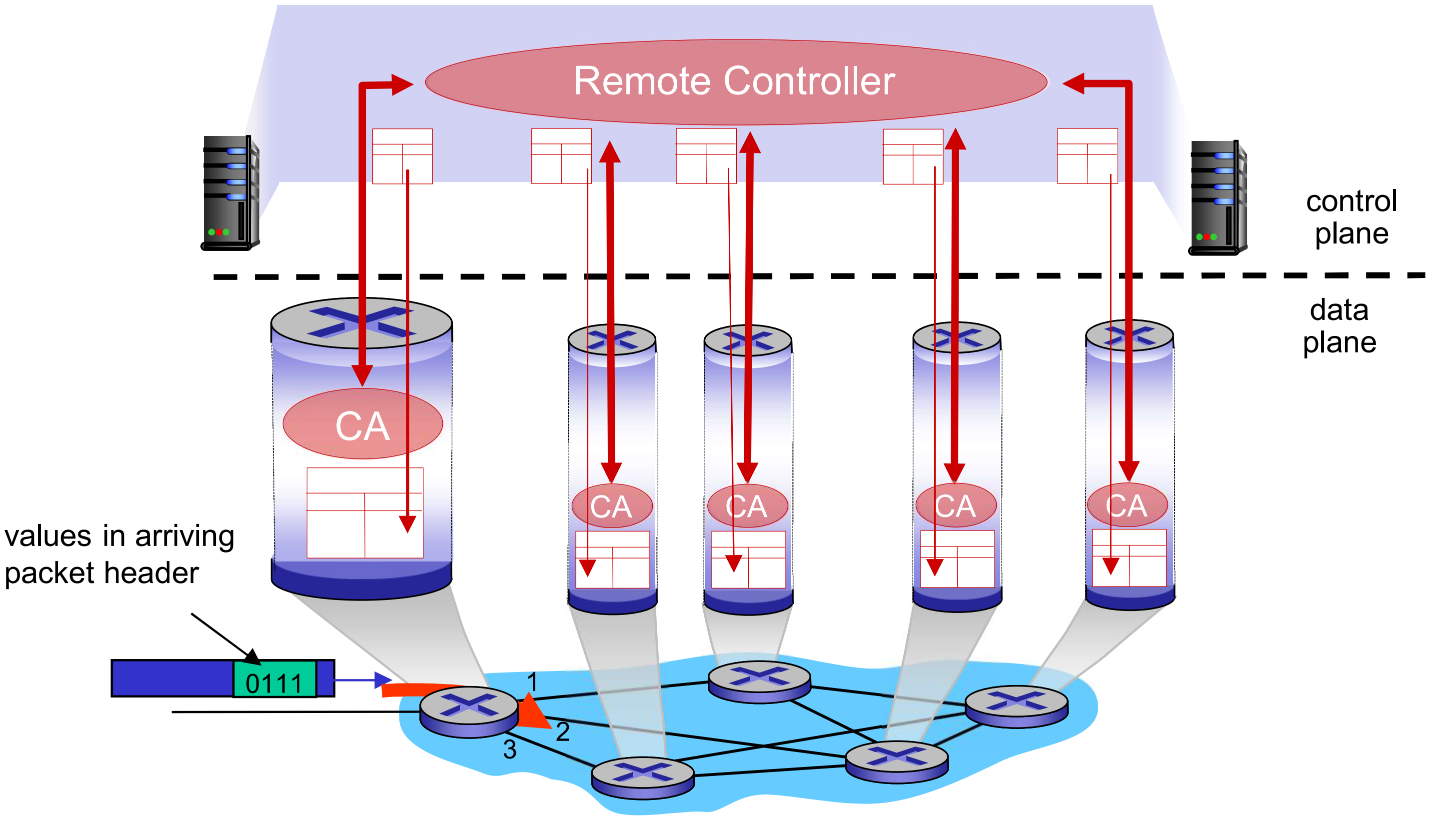
Per-router control plane

Individual routing algorithm components *in each and every router* interact in the control plane



Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



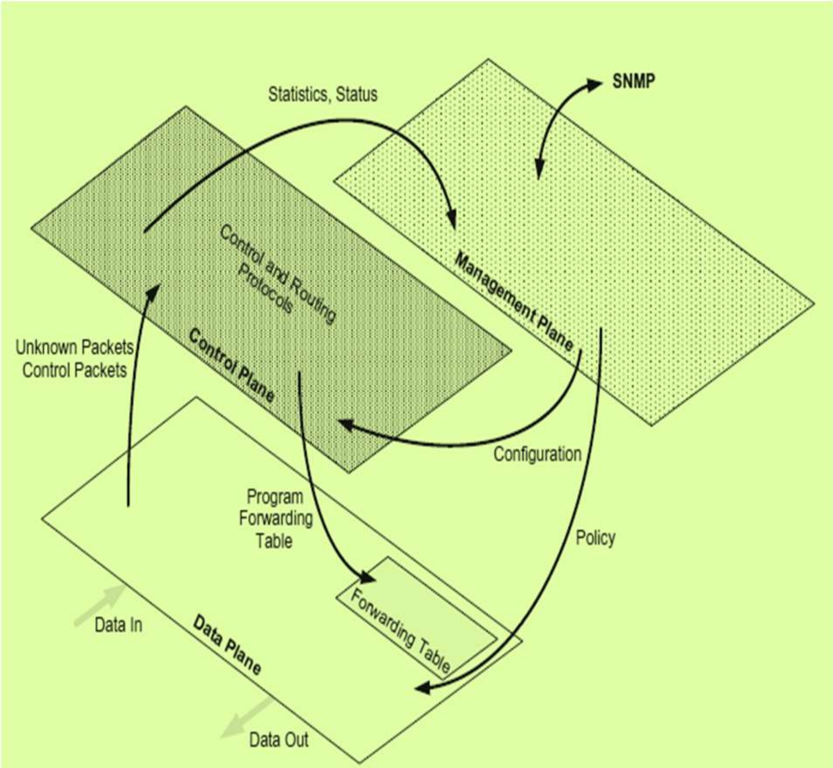
Lets Look into...,

Traditional Switch Architecture
Evolution of Switches, Control Planes
and needs of Modern Data Centers

Evolution of Switches and Control Planes

Traditional Switch Architecture

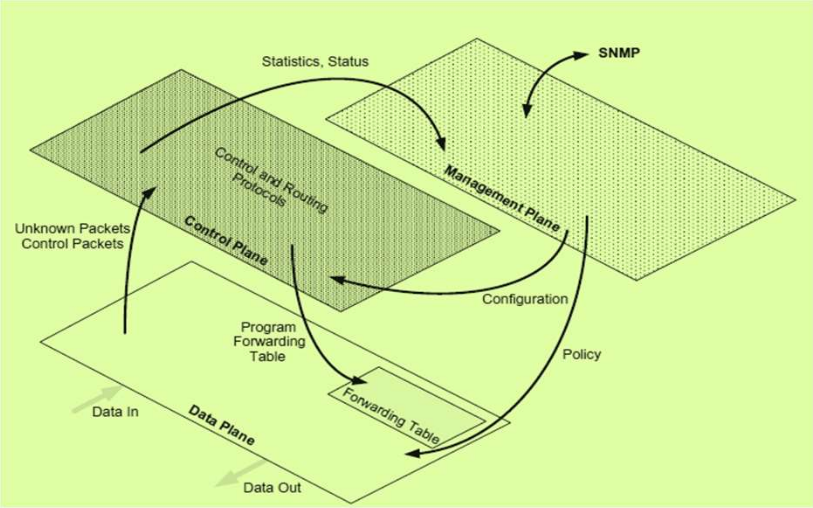
- **Data Plane**
  - Vast majority of packets only touch data plan
  - Contains
    - Reception & Transmission Ports
    - Forwarding Table
- What does the Data Plane do?
  - Packet buffering
  - Packet scheduling
  - Header modification and forwarding
- What happens if the packet information is not in the forwarding table?
  - Data Plane communicates vertically to Control Plane

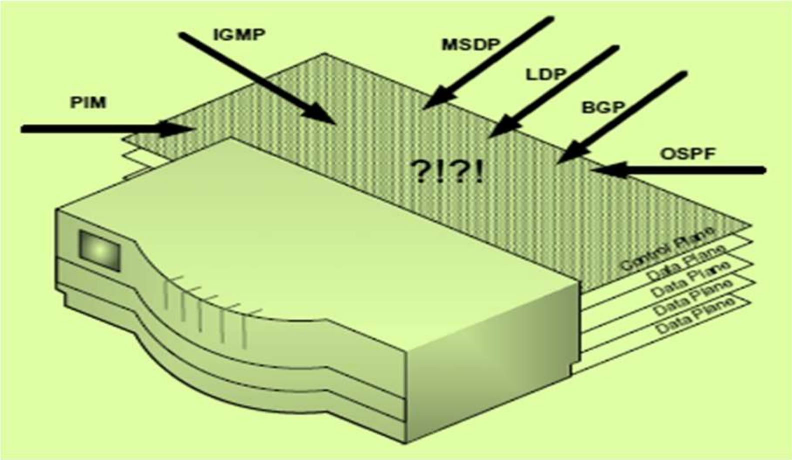


Evolution of Switches and Control Planes

Traditional Switch Architecture

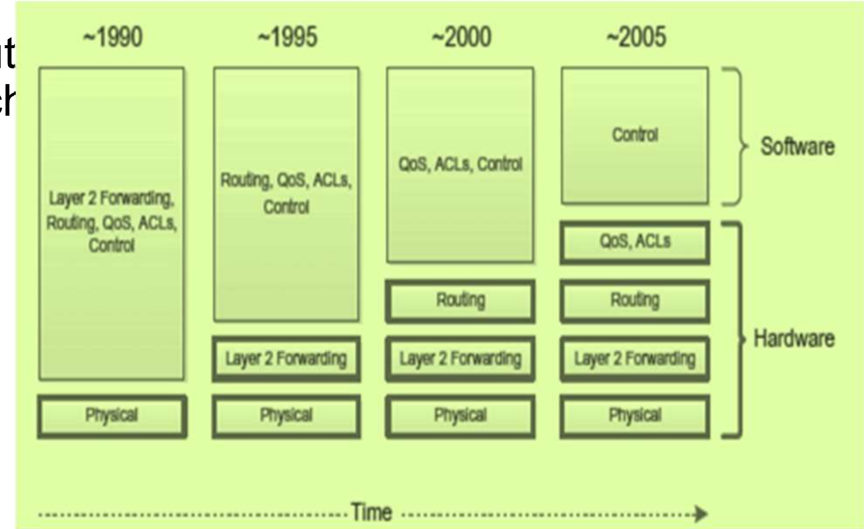
- **Control Plane**
  - Principal role - Keep forwarding table up-to-date
  - Control plane of the switch is bombarded with a constant barrage of control protocol traffic
  - Process control protocols - Control protocols collectively manage the topology of the network.
- **Management Plane**
  - Network administrators configure and monitor the switch through this plane
  - Interfaces vertically to collect or update information in other planes
  - Typically a NMS (network management system) communicates to the plane in the switch.





Evolution of Switches and Control Planes

- Developers → implemented distribut environment with intelligence in each device
- Coordination between devices → Collective decisions
- Goals
  - Simplicity
  - Ease of use
  - Automatic Recovery



- Distributed Intelligence in L2 and L3
  - Spanning Tree Protocol (STP) – IEEE 802.1D

Enforces a hierarchy on the network, Convergence latency

  - Rapid Spanning Tree Protocol (RSTP) – IEEE 802.1D-2004

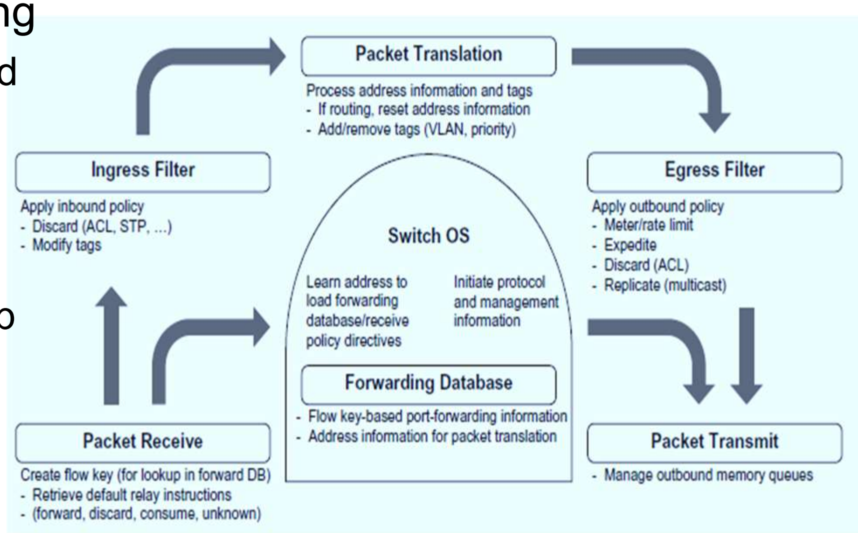
Improves latency, but not deployed

  - Shortest Path Bridging (SPB)
  - RIP, BGP, OSPF, and IS-IS

Layer 3 requires the cooperation between devices, Knowledge of which routers are attaching to which subnets in a network.

Evolution of Switches and Control Planes

- Software-Based Routing and Bridging
  - Ethernet interface speed increased through the 90's.
  - Hardware solutions leveraged to help routers and bridges keep up with increasing speed
  - Eventually software could not keep up with header inspection and routing table lookups
- Generically Programmable Forwarding Rules
  - Early routers – limited packet header field mods.
  - Switch features grew over time
    - Multi-cast, VLANs, MPLS, etc.
    - Pushing programmable rules to the hardware allows complex manipulation while maintaining line rates



This Programmable hardware gives life to the "concept of SDN"

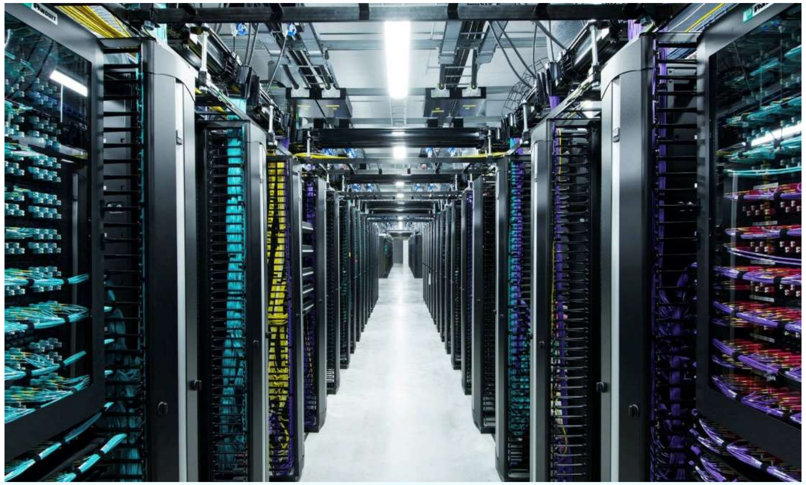
Modern Data Center

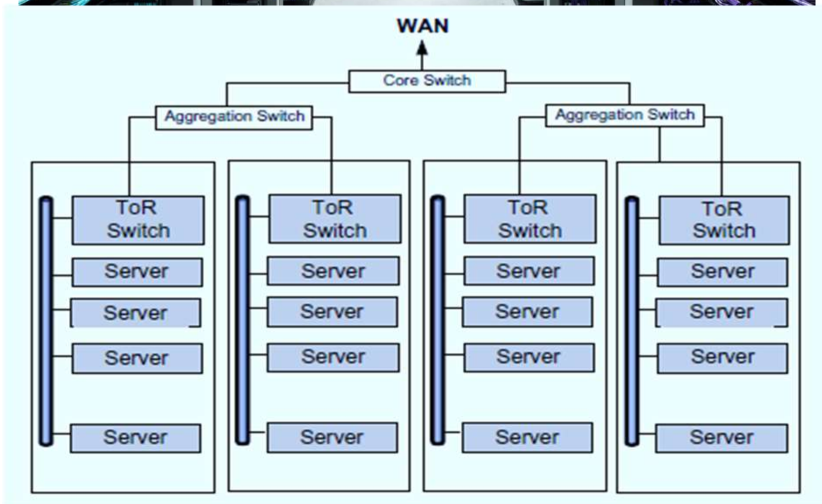
Modern Data Center

- WWW leads to

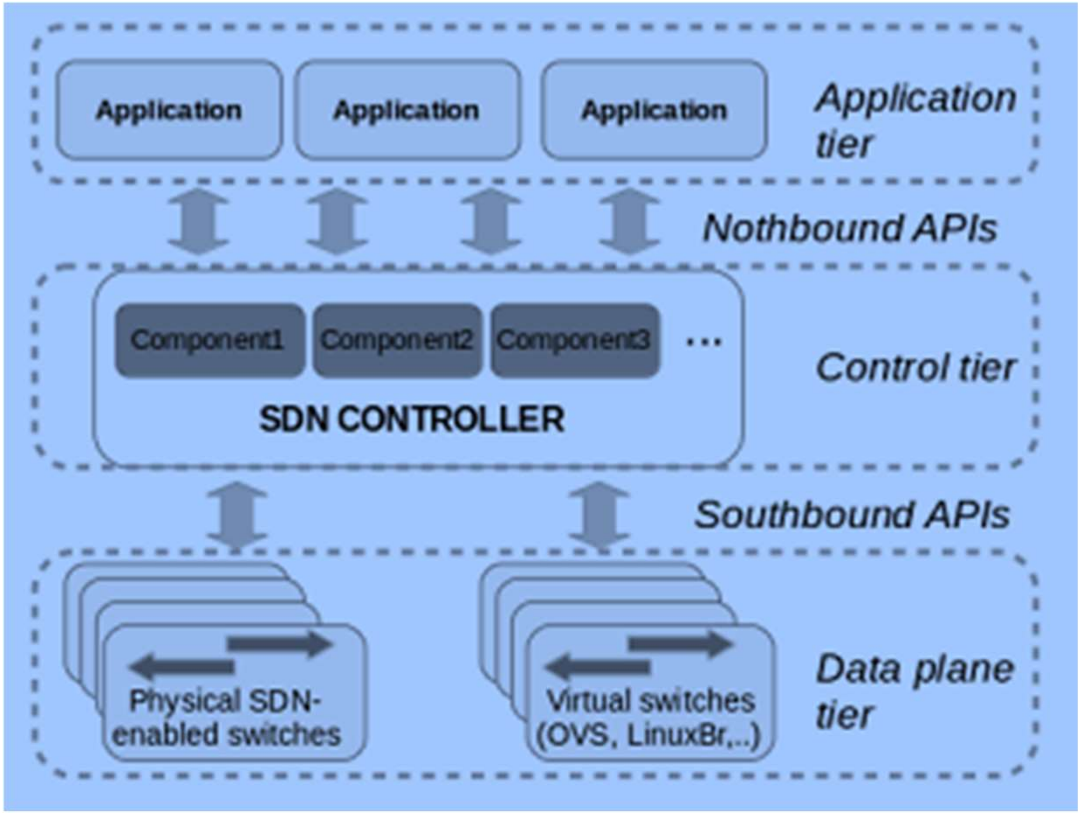
  Large data centers with huge numbers of servers and even greater VMs

- Protocols designed to provide robustness over geographic area not appropriate for the huge amount of traffic in the DC
  - Routers spend 30% of CPU cycles in rediscovery and recalculating routes
  - East / West traffic doesn't benefit from overhead of protocols.
  - SDN was designed to handle the network of the modern data center → fundamental shift from traditional Internet switching



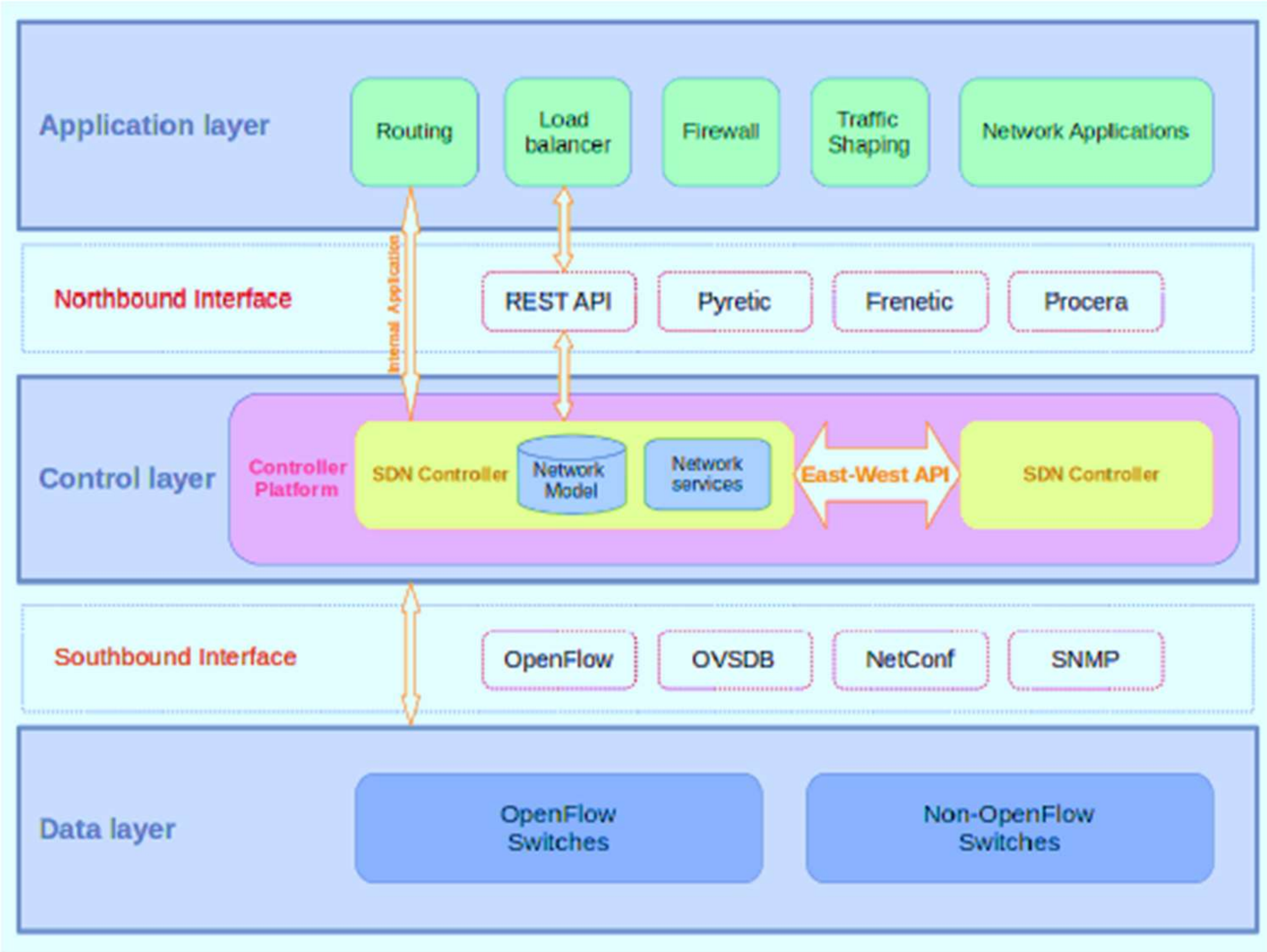


SDN Architecture



SDN general Architecture

SDN Architecture

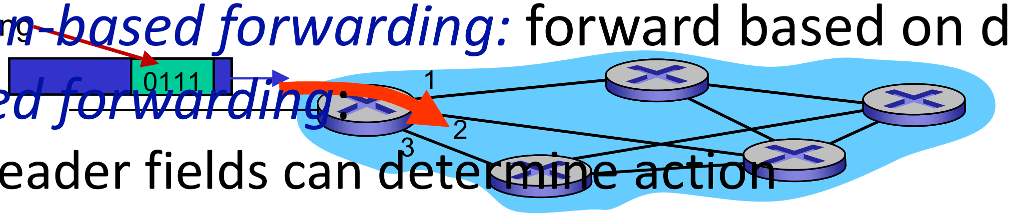


A three-layer distributed SDN Architecture

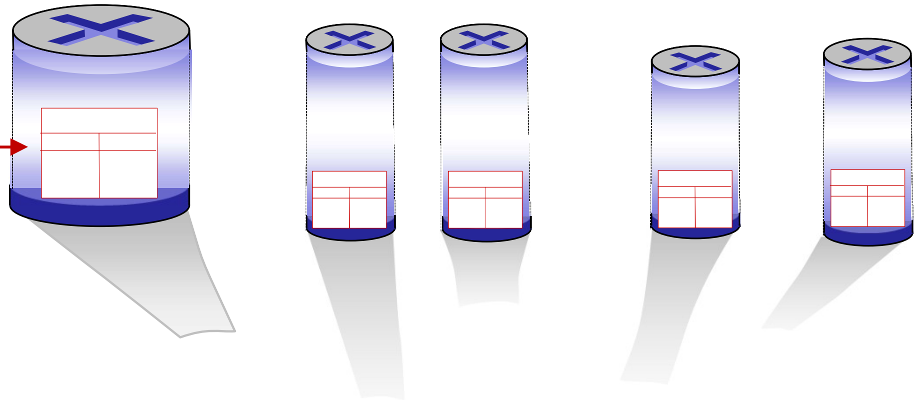
Generalized forwarding: match plus action

Review: each router contains a forwarding table (aka: flow table)

- "match plus action" abstraction: match bits in arriving packet, take action
  - *destination-based forwarding:* forward based on dest. IP address
  - *generalized forwarding:*
    - many header fields can determine action
    - many action possible: drop/copy/modify/log packet

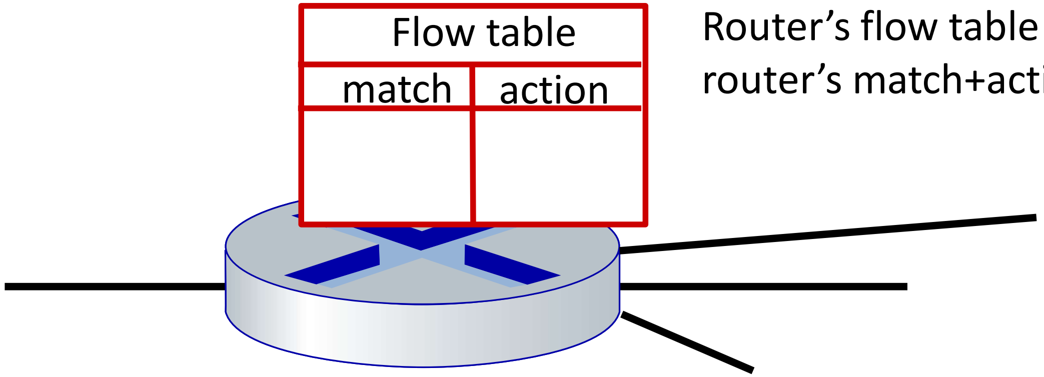


forwarding table
(aka: flow table)



Flow table abstraction

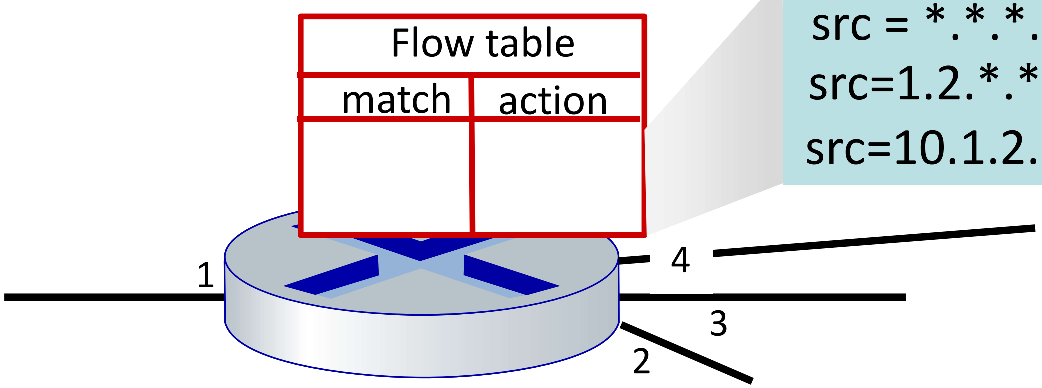
- flow: defined by header field values (in link-, network-, transport-layer fields)
- generalized forwarding: simple packet-handling rules
  - match: pattern values in packet header fields
  - actions: for matched packet: drop, forward, modify, matched packet or send matched packet to controller
  - priority: disambiguate overlapping patterns
  - counters: #bytes and #packets



Router's flow table define router's match+action rules

Flow table abstraction

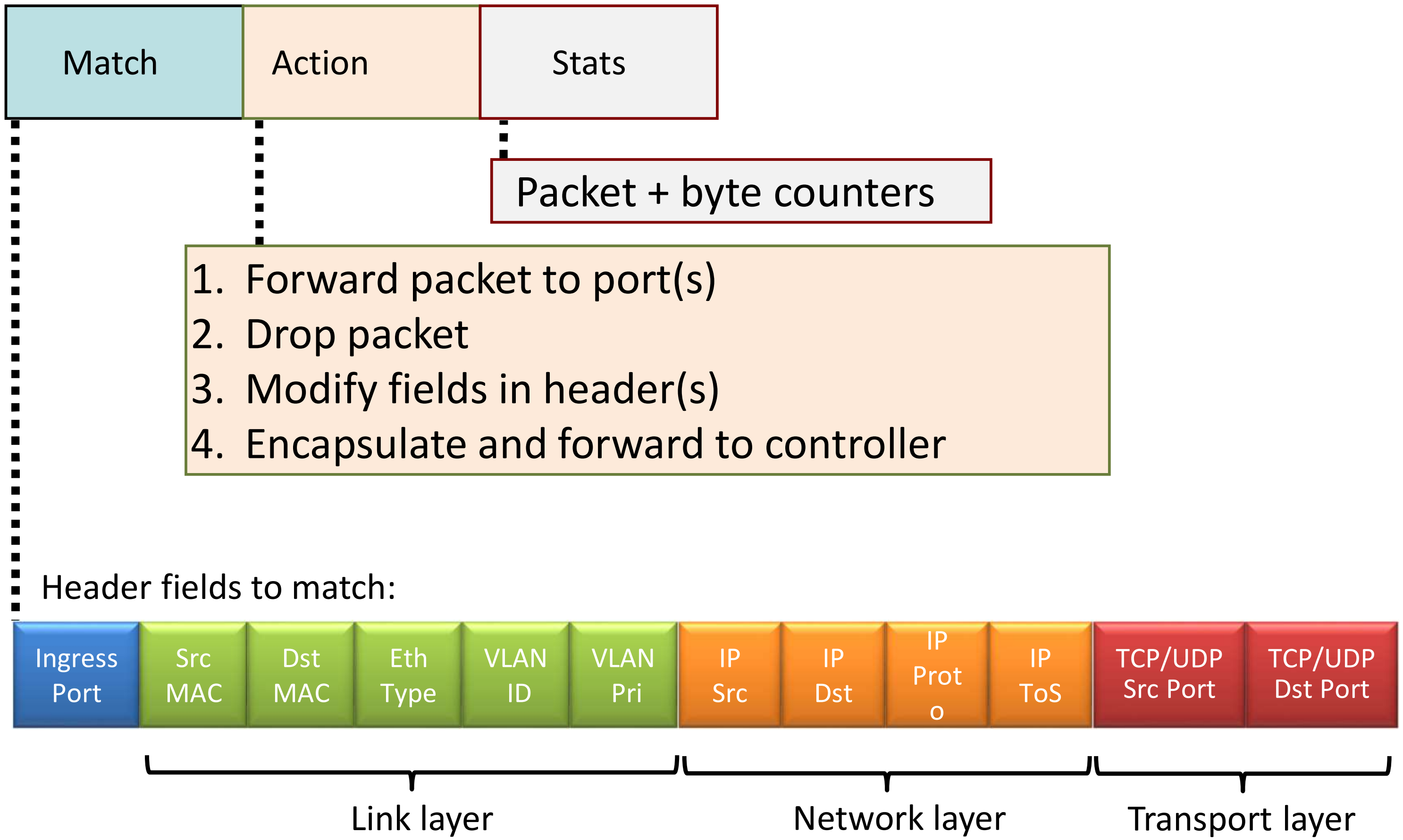
- flow: defined by header fields
- generalized forwarding: simple packet-handling rules
  - match: pattern values in packet header fields
  - actions: for matched packet: drop, forward, modify, matched packet or send matched packet to controller
  - priority: disambiguate overlapping patterns
  - counters: #bytes and #packets



src = *.*.*.*, dest=3.4.*.*	forward(2)
src=1.2.*.*, dest=*.*.*.*	drop
src=10.1.2.3, dest=*.*.*.*	send to controller

* : wildcard

OpenFlow: flow table entries



OpenFlow: examples

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	51.6.0.8	*	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	*	*	*	22	drop

Block (do not forward) all datagrams destined to TCP port 22 (ssh port #)

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	128.119.1.1	*	*	*	*	*	drop

Block (do not forward) all datagrams sent by host 128.119.1.1

OpenFlow: examples

Layer 2 destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	*	port3

layer 2 frames with destination MAC address 22:A7:23:11:E1:02 should be forwarded to output port 3

OpenFlow abstraction

- match+action: abstraction unifies different kinds of devices

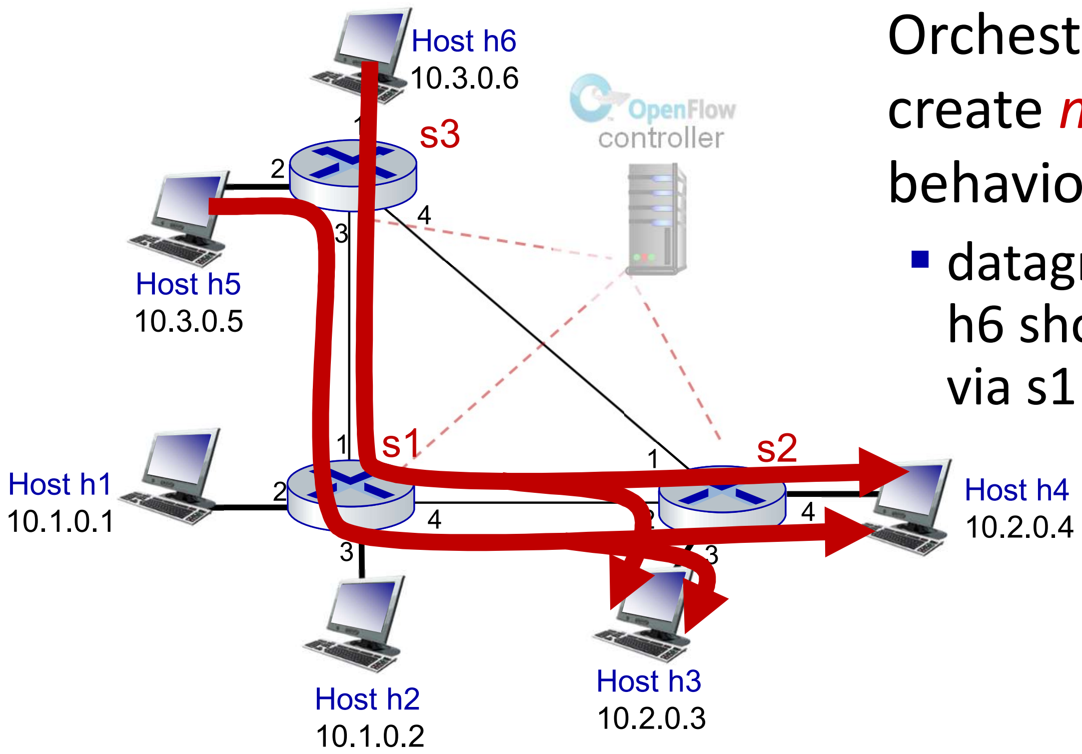
Router
- *match:* longest destination IP prefix
- *action:* forward out a link

Switch
- *match:* destination MAC address
- *action:* forward or flood

Firewall
- *match*: IP addresses and TCP/UDP port numbers
- *action:* permit or deny

NAT
- *match:* IP address and port
- *action:* rewrite address and port

OpenFlow example

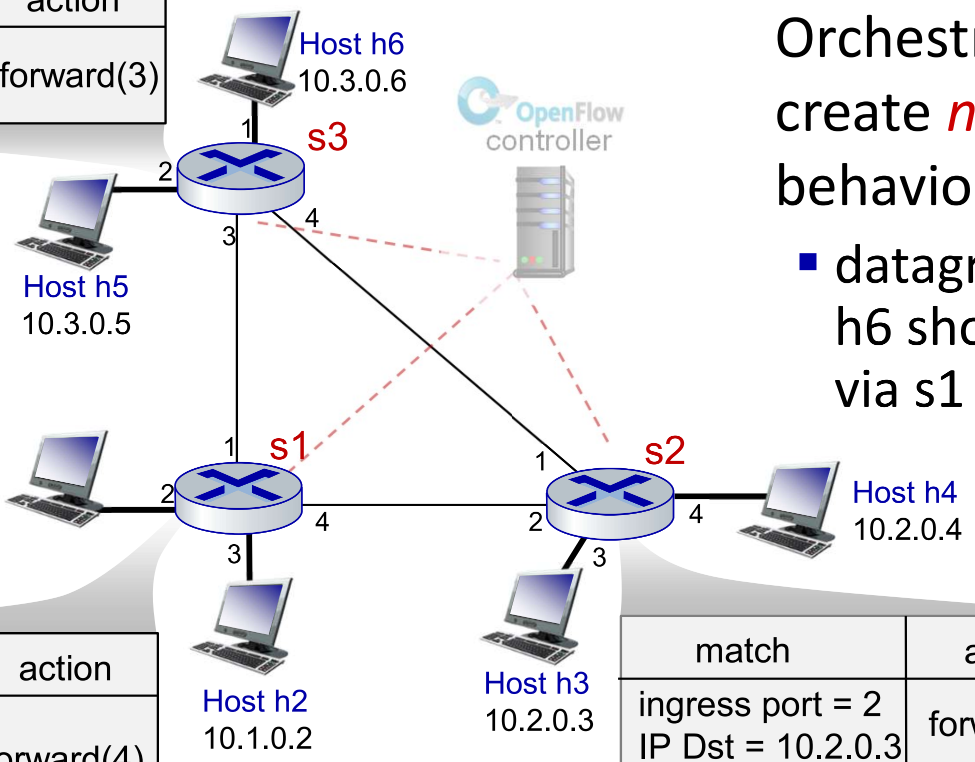


Orchestrated tables can create *network-wide* behavior, e.g.,:

- datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

OpenFlow example

match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)



match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)

Orchestrated tables can create *network-wide* behavior, e.g.,:

- datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

Generalized forwarding: summary

- "match plus action" abstraction: match bits in arriving packet header(s) in any layers, take action
  - matching over many fields (link-, network-, transport-layer)
  - local actions: drop, forward, modify, or send matched packet to controller
  - "program" *network-wide* behaviors
- simple form of "network programmability"
  - programmable, per-packet "processing"
  - *historical roots:* active networking
  - *today:* more generalized programming

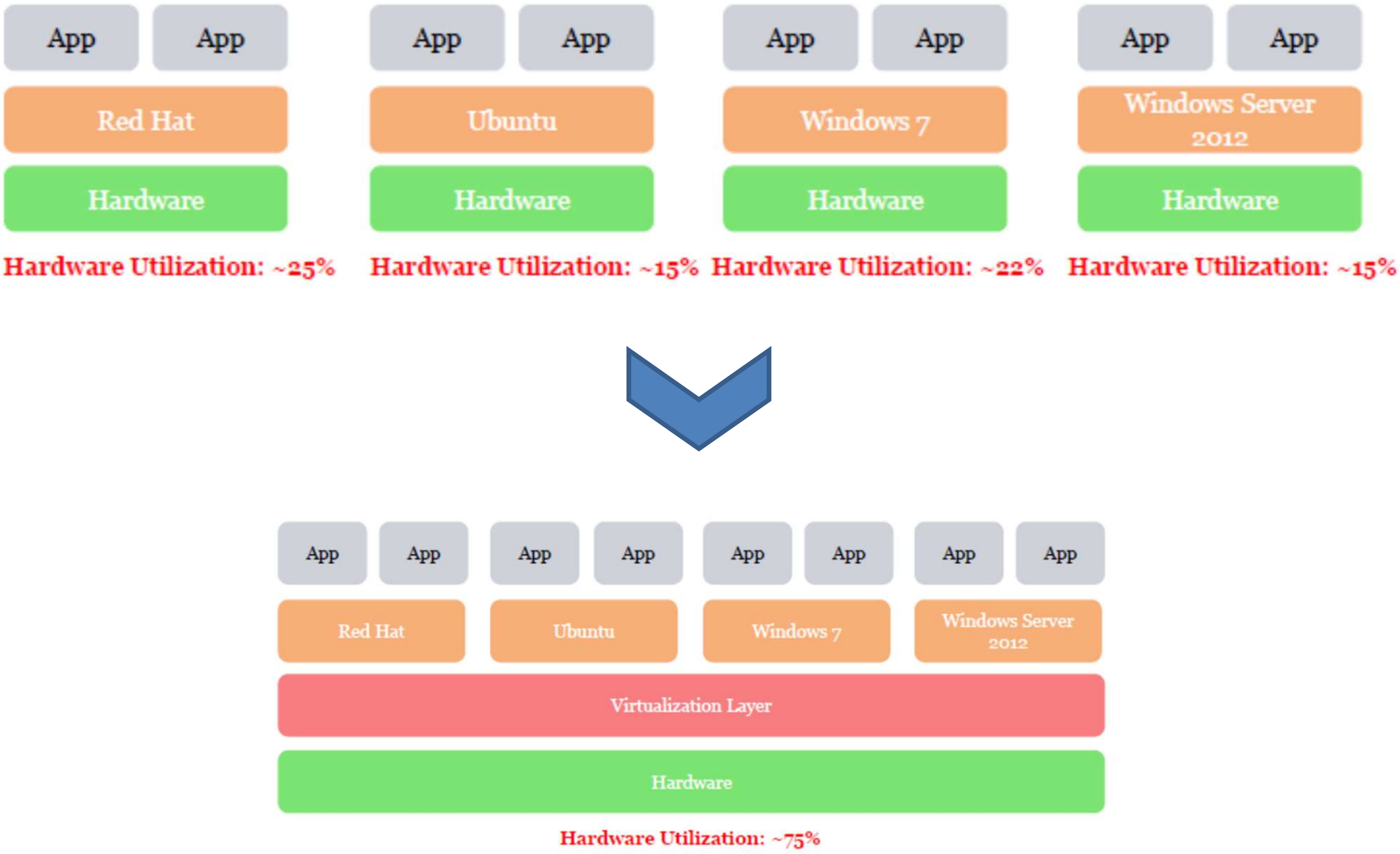
innovate achieve lead

BITS Pilani
Pilani Campus

Network Function Virtualization (NFV)

Some Slides Taken and Adapted from:
Computer Networking: A Top-Down Approach, 8th edition, Jim Kurose, Keith Ross, Pearson, 2020

Virtualization Concept



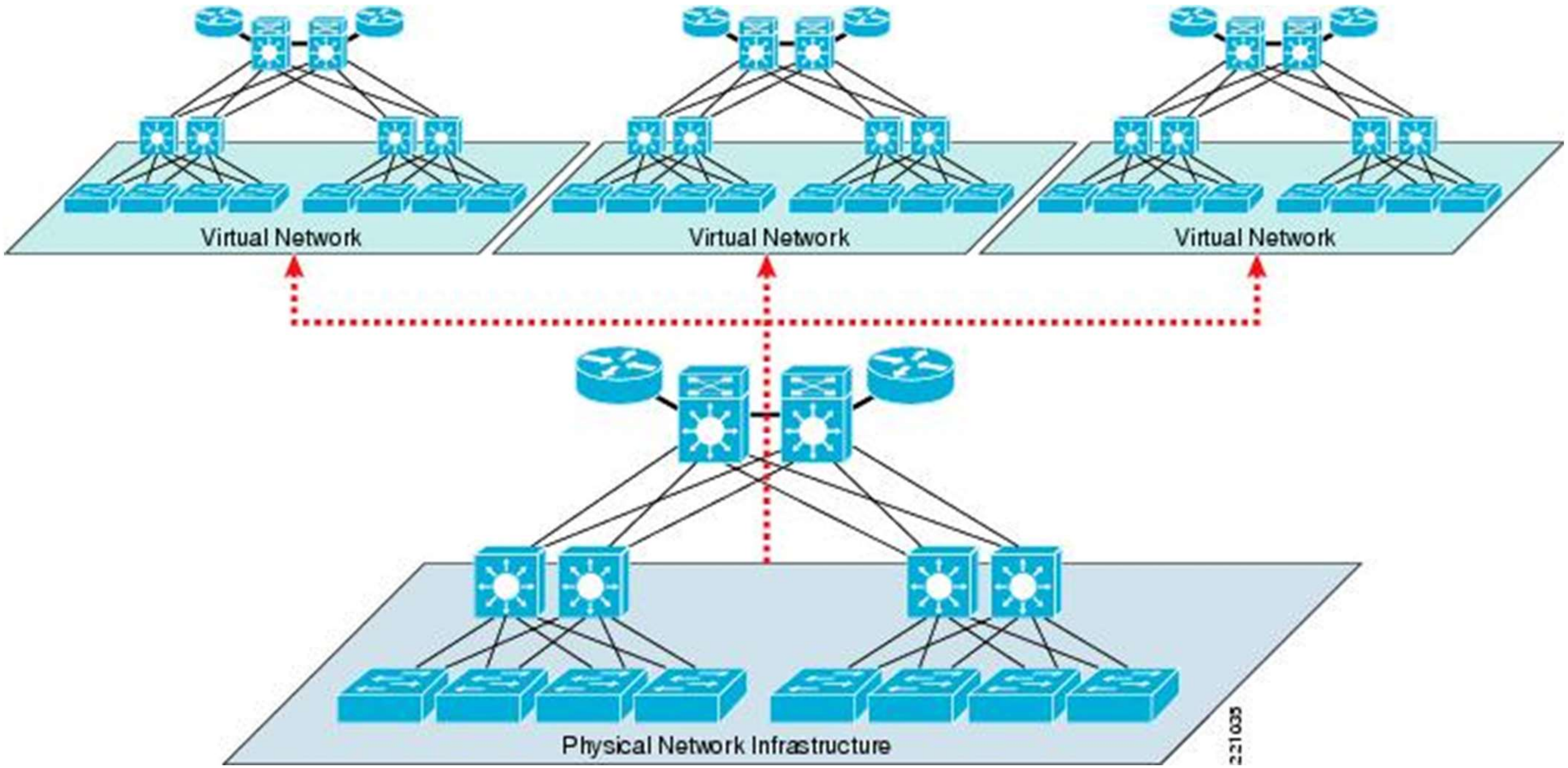
Network Virtualization

- What is network virtualization ?
  - In computing, Network Virtualization is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network.

- Two categories :
  - External network virtualization
    - Combining many networks, or parts of networks, into a virtual unit.
  - Internal network virtualization
    - Providing network-like functionality to the software containers on a single system.

Network Virtualization

- What is network virtualization ?



Network Virtualization

- Desirable properties of network virtualization :
  - Scalability
    - Easy to extend resources in need
    - Administrator can dynamically create or delete virtual network connection
  - Resilience
    - Recover from the failures
    - Virtual network will automatically redirect packets by redundant links
  - Security
    - Increased path isolation and user segmentation
    - Virtual network should work with firewall software
  - Availability
    - Access network resource anytime

Network Virtualization

- External network virtualization in different layers :
  - Layer 1
    - Seldomly implemented in physical data transmission layer.
  - Layer 2
    - Use some tags in MAC address packet to provide virtualization.
    - Example, VLAN.
  - Layer 3
    - Use some tunnel techniques to form a virtual network.
    - Example, VPN.
  - Layer 4 or higher
    - Build up some overlay network for some application.
    - Example, P2P.

Network Virtualization

- Internal network virtualization in different layers :
  - Layer 1
    - Hypervisor usually do not need to emulate the physical layer.
  - Layer 2
    - Implement virtual L2 network devices, such as switch, in hypervisor.
  - Layer 3
    - Implement virtual L3 network devices, such as router, in hypervisor.
  - Layer 4 or higher
    - Layer 4 or higher layers virtualization is usually implemented in guest OS.