Recent Advances in Networking: Network *Virtualization and Software Defined Networking for Cloud Computing*



RAJ JAIN

Washington University in Saint Louis
Saint Louis, MO 63130, Jain@cse.wustl.edu

Tutorial at IEEE International Conference on Sensing, Communication, and Networking (SECON), Singapore, June 30, 2014

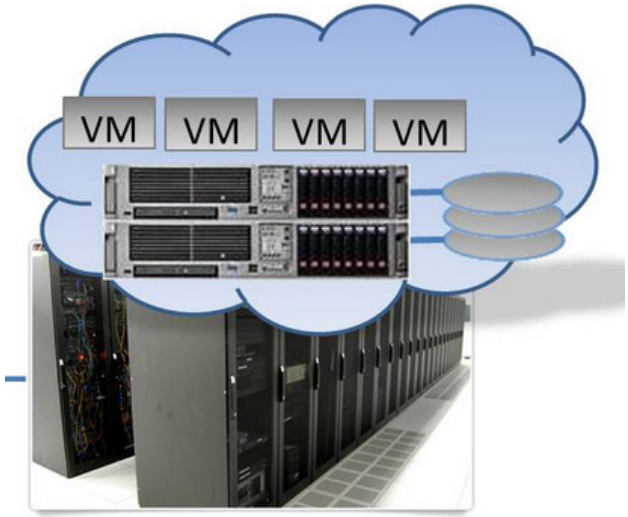
These slides and a video recording of the tutorial are at:

http://www.cse.wustl.edu/~jain/tutorials/secon14.htm

Overview

1. Part I: Network Virtualization
2. Part II: Data Center Bridging
3. Part III: Carrier Ethernet for Data Centers
4. Part IV: Virtual Bridging
5. Part V: LAN Extension and Partitioning
6. Part VI: OpenFlow and Tools
7. Part VII: Software Defined Networking
8. Part VIII: Network Function Virtualization

Part I: Network Virtualization



1. Virtualization
2. Why Virtualize?
3. Network Virtualization
4. Names, IDs, Locators
5. Interconnection Devices

Part II: Data Center Bridging

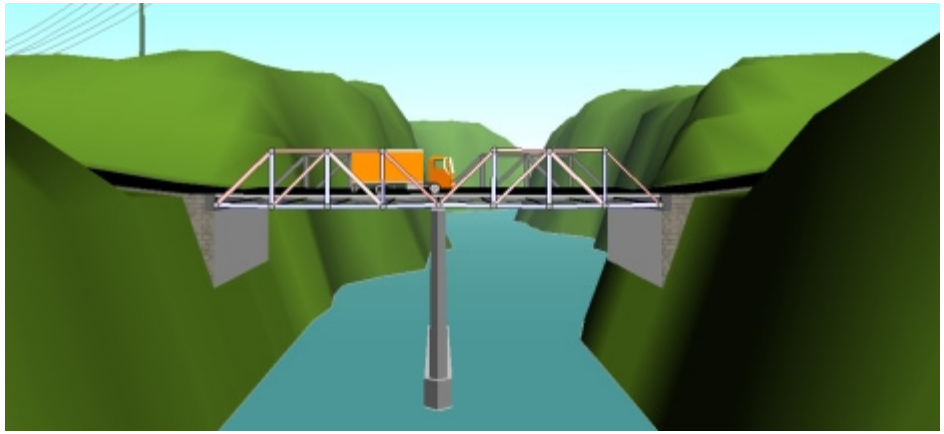
1. Residential vs. Data Center Ethernet
2. Review of Ethernet devices and algorithms
3. Enhancements to Spanning Tree Protocol
4. Virtual LANs
5. Data Center Bridging Extensions

Part III: Carrier Ethernet for Data Centers

1. Provider Bridges (PB) or Q-in-Q
2. Provider Backbone Bridges (PBB) or MAC-in-MAC
3. Provider Backbone Bridges with Traffic Engineering (PBB-TE)

Note: Although these technologies were originally developed for carriers, they are now used inside multi-tenant data centers (clouds)

Part IV: Virtual Bridging



1. Virtual Bridges to connect virtual machines
2. IEEE Virtual Edge Bridging Standard
3. Single Root I/O Virtualization (SR-IOV)
4. Aggregating Bridges and Links: VSS and vPC
5. Bridges with massive number of ports: VBE

Part V: LAN Extension and Partitioning

1. Transparent Interconnection of Lots of Links (TRILL)
2. Network Virtualization using GRE (NVGRE)
3. Virtual eXtensible LANs (VXLAN)
4. Stateless Transport Tunneling Protocol (STT)

Part VI: OpenFlow and Tools

- Planes of Networking
- OpenFlow
- OpenFlow Switches including Open vSwitch
- OpenFlow Evolution
- OpenFlow Configuration Protocol (OF-Config)
- OpenFlow Notification Framework
- OpenFlow Controllers

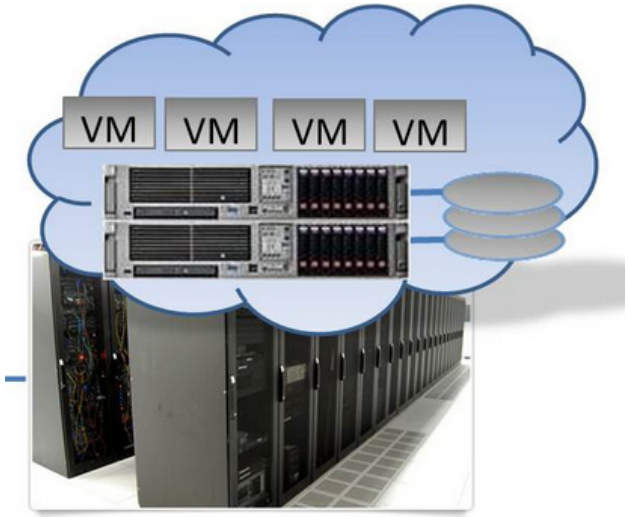
Part VII: Software Defined Networking

- What is SDN?
- Alternative APIs: XMPP, PCE, ForCES, ALTO
- OpenDaylight SDN Controller Platform and Tools

Part VIII: Network Function Virtualization

- What is NFV?
- NFV and SDN Relationship
- ETSI NFV ISG Specifications
- Concepts, Architecture, Requirements, Use cases
- Proof-of-Concepts and Timeline

Part I: Network Virtualization



1. Virtualization
2. Why Virtualize?
3. Network Virtualization
4. Names, IDs, Locators
5. Interconnection Devices

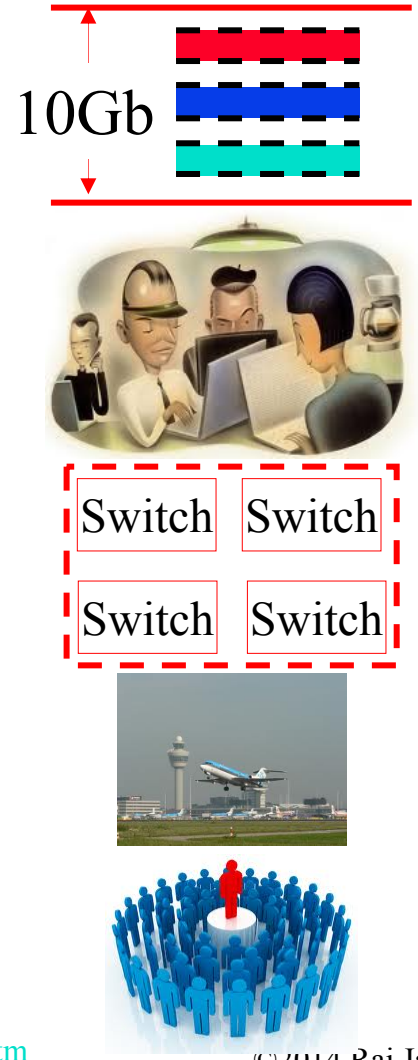
Virtualization

"*Virtualization means that Applications can use a resource without any concern for where it resides, what the technical interface is, how it has been implemented, which platform it uses, and how much of it is available.*"

-Rick F. Van der Lans
in Data Virtualization for Business Intelligence Systems

5 Reasons to Virtualize

1. Sharing: Break up a large resource
   Large Capacity or high-speed
   E.g., Servers
2. Isolation: Protection from other tenants
   E.g., Virtual Private Network
3. Aggregating: Combine many resources in to one, e.g., storage
4. Dynamics: Fast allocation, Change/Mobility, Follow the sun (active users) or follow the moon (cheap power)
5. Ease of Management ⇒ Easy distribution, deployment, testing



Virtualization in Computing

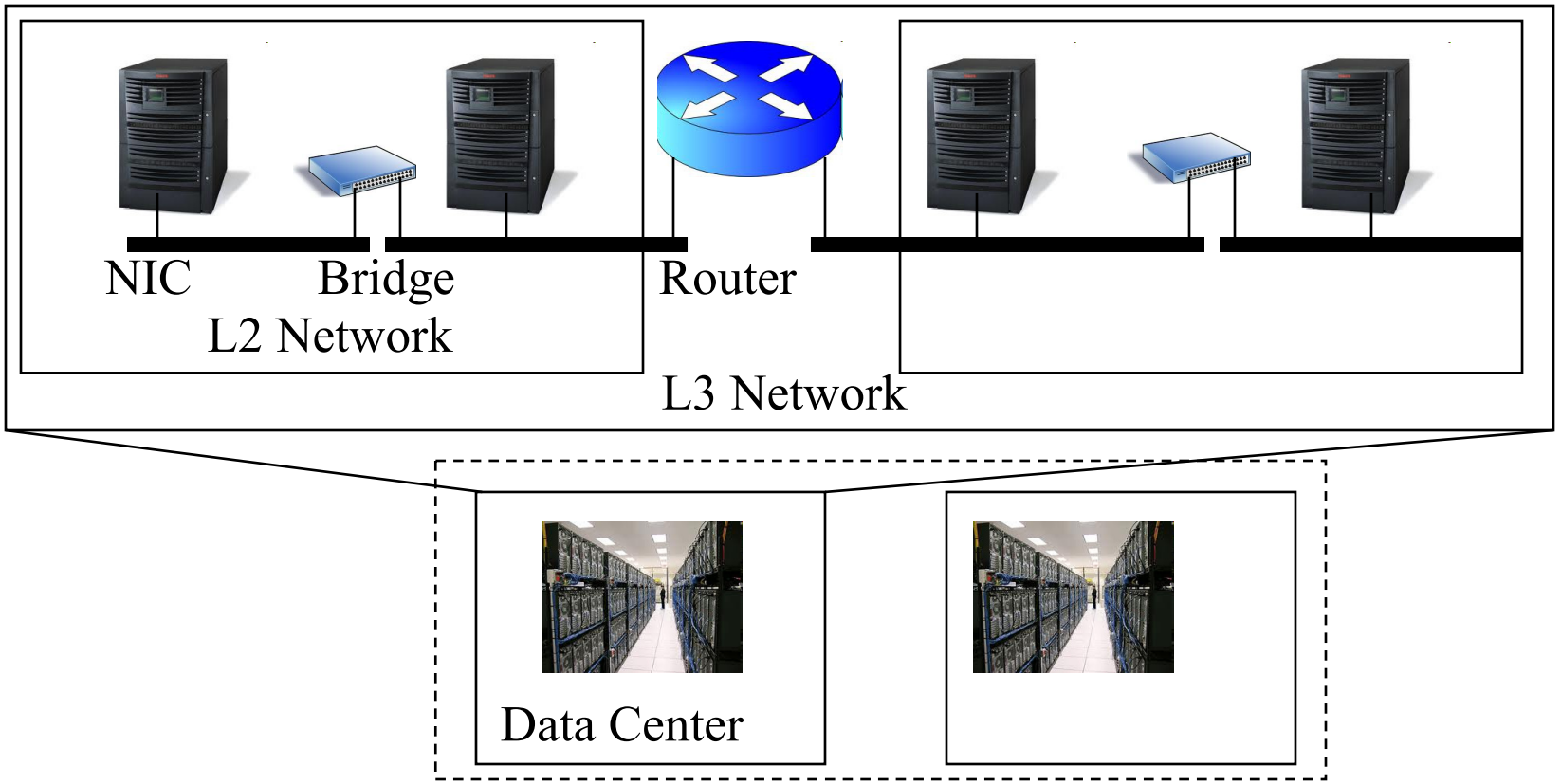
- **Storage**:
  - Virtual Memory ⇒ L1, L2, L3, ... ⇒ Recursive
  - Virtual CDs, Virtual Disks (RAID), Cloud storage
- **Computing**:
  - Virtual Desktop ⇒ Virtual Server ⇒ Virtual Datacenter

    Thin Client ⇒ VMs ⇒ Cloud
- **Networking**: Plumbing of computing
  - Virtual Channels, Virtual LANs, Virtual Private Networks

Network Virtualization

1. Network virtualization allows tenants to form an overlay network in a multi-tenant network such that tenant can control:
   1. Connectivity layer: Tenant network can be L2 while the provider is L3 and vice versa
   2. Addresses: MAC addresses and IP addresses
   3. Network Partitions: VLANs and Subnets
   4. Node Location: Move nodes freely
2. Network virtualization allows providers to serve a large number of tenants without worrying about:
   1. Internal addresses used in client networks
   2. Number of client nodes
   3. Location of individual client nodes
   4. Number and values of client partitions (VLANs and Subnets)
3. Network could be a single physical interface, a single physical machine, a data center, a metro, … or the global Internet.
4. Provider could be a system owner, an enterprise, a cloud provider, or a carrier.

Levels of Network Virtualization



- Networks consist of: **Network Interface Card (NIC)** – **L2 Links** - **L2 Bridges** - **L2 Networks** - L3 Links - L3 Routers - L3 Networks – **Data Centers** – **Global Internet.**
- Each of these needs to be virtualized

Network Virtualization Techniques

Entity	Partitioning	Aggregation/Extension/Interconnection**
NIC	SR-IOV	MR-IOV
Switch	VEB, VEPA	VSS, VBE, DVS, FEX
L2 Link	VLANs	LACP, Virtual PortChannels
L2 Network using L2	VLAN	PB (Q-in-Q), PBB (MAC-in-MAC), PBB-TE, Access-EPL, EVPL, EVP-Tree, EVPLAN
L2 Network using L3	NVO3, VXLAN, NVGRE, STT	MPLS, VPLS, A-VPLS, H-VPLS, PWoMPLS, PWoGRE, OTV, TRILL, LISP, L2TPv3, EVPN, PBB-EVPN
Router	VDCs, VRF	VRRP, HSRP
L3 Network using L1		GMPLS, SONET
L3 Network using L3*	MPLS, GRE, PW, IPSec	MPLS, T-MPLS, MPLS-TP, GRE, PW, IPSec
Application	ADCs	Load Balancers

*All L2/L3 technologies for L2 Network partitioning and aggregation can also be used for L3 network partitioning and aggregation, respectively, by simply putting L3 packets in L2 payloads.

**The aggregation technologies can also be seen as partitioning technologies from the provider point of view.

Names, IDs, Locators

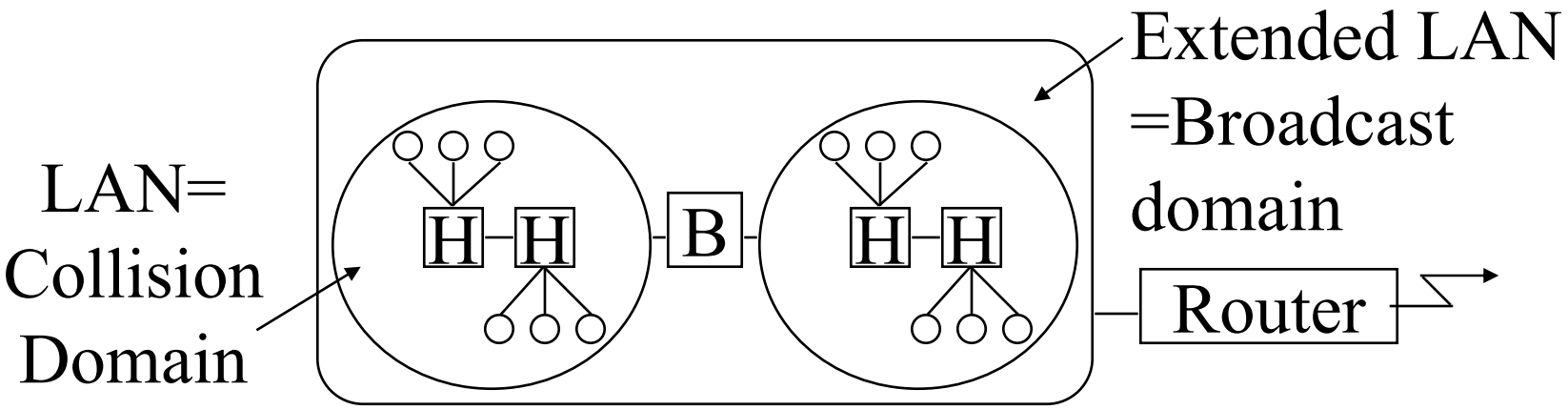
Name: John Smith

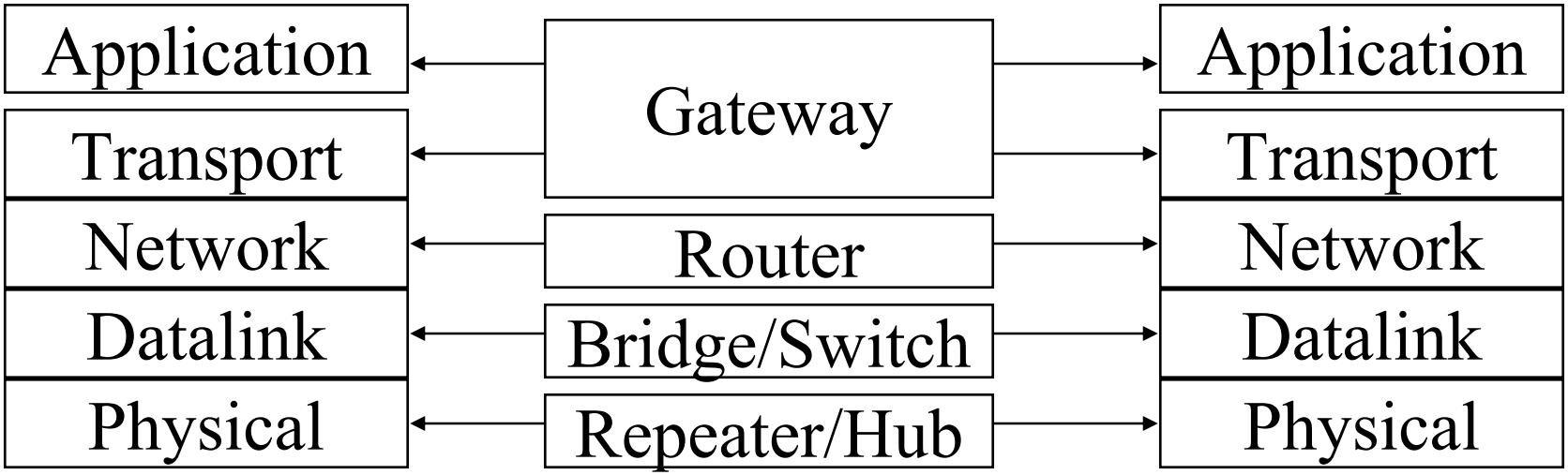
ID: 012-34-5678

Locator:
1234 Main Street
Big City, MO 12345
USA

- Locator changes as you move, ID and Names remain the same.
- **Examples**:
  - Names: Company names, DNS names (Microsoft.com)
  - IDs: Cell phone numbers, 800-numbers, Ethernet addresses, Skype ID, VOIP Phone number
  - Locators: Wired phone numbers, IP addresses

Interconnection Devices





Interconnection Devices (Cont)

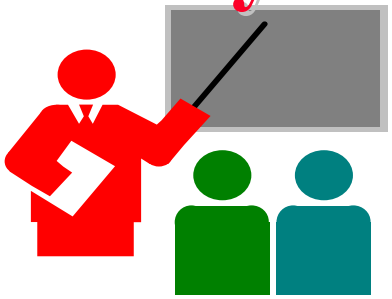
- **Repeater**: PHY device that restores data and collision signals
- **Hub**: Multiport repeater + fault detection and recovery
- **Bridge**: Datalink layer device connecting two or more collision domains. MAC multicasts are propagated throughout "extended LAN."
- **Router**: Network layer device. IP, IPX, AppleTalk. Does not propagate MAC multicasts.
- **Switch**: Multiport bridge with parallel paths
- These are functions. Packaging varies.
- No CSMA/CD in 10G and up
- No CSMA/CD in practice now even at home or at 10 Mbps

Fallacies Taught in Networking Classes

1. Ethernet is a local area network (Local $\leq$ 2km)
2. Token ring, Token Bus, and CSMA/CD are the three most common LAN access methods.
3. Ethernet uses CSMA/CD.
4. Ethernet bridges use spanning tree for packet forwarding.
5. Ethernet frames are limited to 1518 bytes.
6. Ethernet does not provide any delay guarantees.
7. Ethernet has no congestion control.
8. Ethernet has strict priorities.

Ethernet has changed.
All of these are now false or are becoming false.

Summary of Part I



1. Virtualization allows applications to use resources without worrying about its location, size, format etc.
2. Ethernet's use of IDs as addresses makes it very easy to move systems in the data center ⇒ Keep traffic on the same Ethernet
3. Cloud computing requires Ethernet to be extended globally and partitioned for sharing by a very large number of customers who have complete control over their address assignment and connectivity
4. Many of the previous limitations of Ethernet have been overcome in the last few years.

Part II: Data Center Bridging

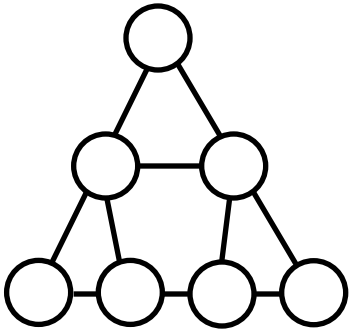
1. Residential vs. Data Center Ethernet
2. Review of Ethernet devices and algorithms
3. Enhancements to Spanning Tree Protocol
4. Virtual LANs
5. Data Center Bridging Extensions

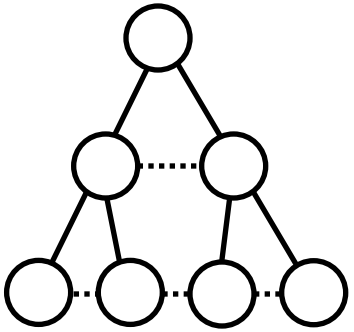
Residential vs. Data Center Ethernet

Residential	Data Center/Cloud
❑ Distance: up to 200m	❑ No limit
❑ Scale: ➢ Few MAC addresses ➢ 4096 VLANs	❑ Millions of MAC Addresses ❑ Millions of VLANs Q-in-Q
❑ Protection: Spanning tree	❑ Rapid spanning tree, … (Gives 1s, need 50ms)
❑ Path determined by spanning tree	❑ Traffic engineered path
❑ Simple service	❑ Service Level Agreement. Rate Control.
❑ Priority ⇒ Aggregate QoS	❑ Need per-flow/per-class QoS
❑ No performance/Error monitoring (OAM)	❑ Need performance/BER

Spanning Tree and its Enhancements

- Helps form a tree out of a mesh topology
- A topology change can result in 1 minute of traffic loss with STP ⇒ All TCP connections break
- Rapid Spanning Tree Protocol (RSTP)
  IEEE 802.1w-2001 incorporated in IEEE 802.1D-2004
- One tree for all VLANs
  ⇒ Common spanning tree
- Many trees
  ⇒ Multiple spanning tree (MST) protocol
  IEEE 802.1s-2002 incorporated in IEEE 802.1Q-2005
- One or more VLANs per tree.





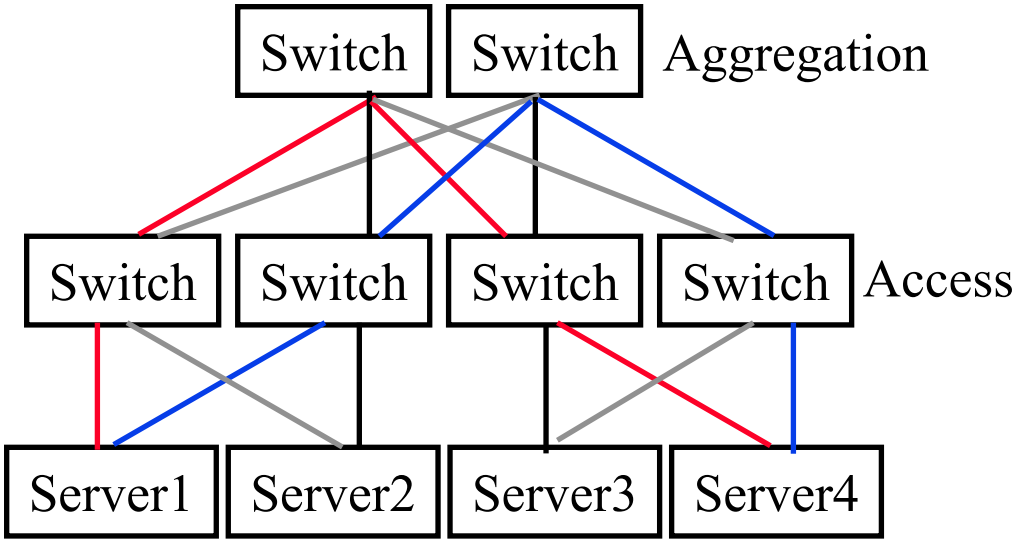
IS-IS Protocol

- Intermediate System to Intermediate System (IS-IS) is a protocol to build routing tables. Link-State routing protocol
  ⇒ Each nodes sends its connectivity (link state) information to all nodes in the network
- Dijkstra's algorithm is then used by each node to build its routing table.
- Similar to OSPF (Open Shortest Path First).
- OSPF is designed for IPv4 and then extended for IPv6.
  IS-IS is general enough to be used with any type of addresses
- OSPF is designed to run on the top of IP
  IS-IS is general enough to be used on any transport
  ⇒ Adopted by Ethernet

Ref: http://en.wikipedia.org/wiki/IS-IS

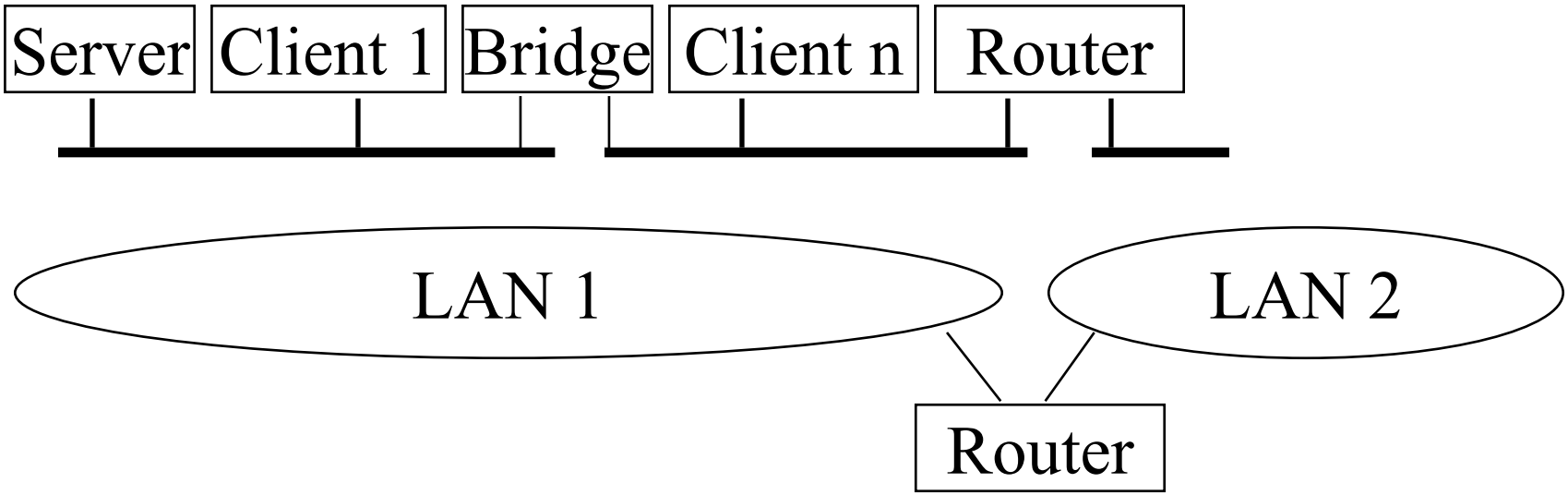
Shortest Path Bridging

- IEEE 802.1aq-2012
- Allows all links to be used ⇒ Better CapEx
- IS-IS link state protocol (similar to OSPF) is used to build shortest path trees for each node to every other node within the SPB domain
- Equal-cost multi-path (ECMP) used to distribute load



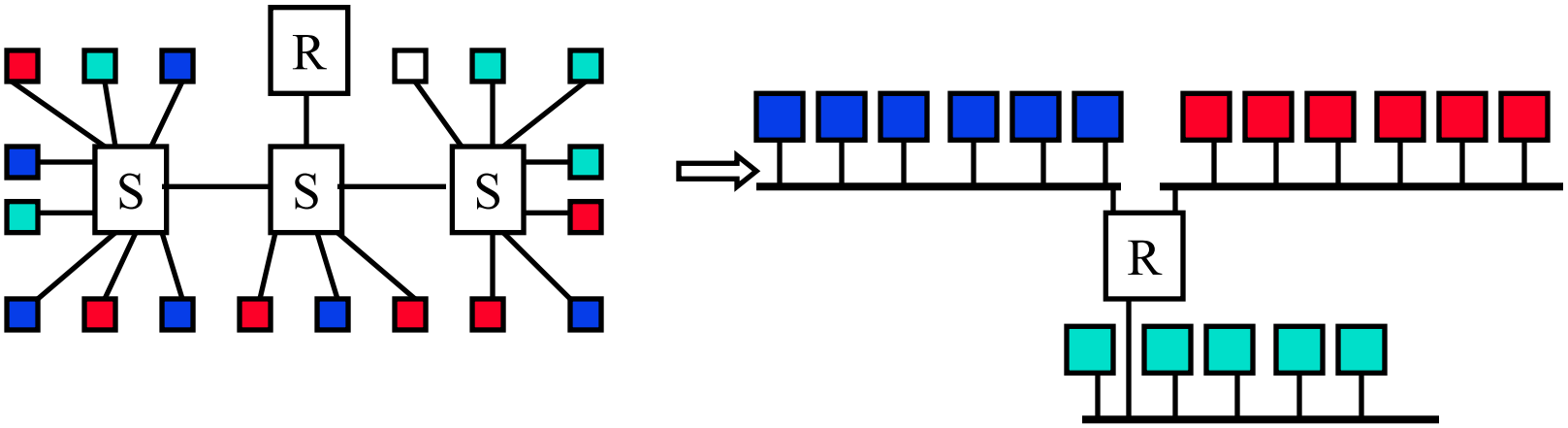
Ref: http://en.wikipedia.org/wiki/Shortest_Path_Bridging

What is a LAN?



- LAN = Single broadcast domain = Subnet
- No routing between members of a LAN
- Routing required between LANs

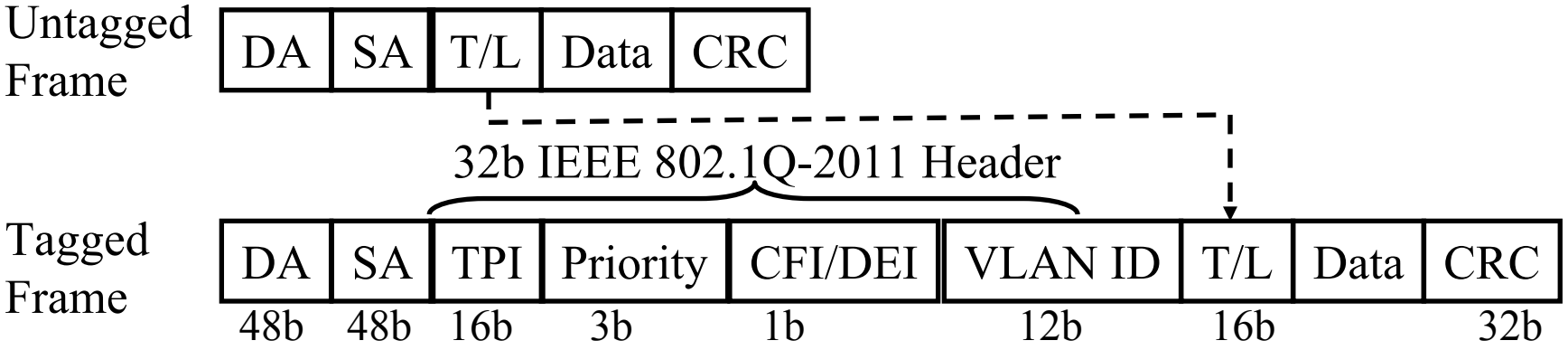
Virtual LAN



- Virtual LAN = Broadcasts and multicast goes only to the nodes in the virtual LAN
- LAN membership defined by the network manager ⇒ Virtual

IEEE 802.1Q-2011 Tag

- Tag Protocol Identifier (TPI)
- Priority Code Point (PCP): 3 bits = 8 priorities 0..7 (High)
- Canonical Format Indicator (CFI): 0 ⇒ Standard Ethernet, 1 ⇒ IBM Token Ring format (non-canonical or non-standard)
- CFI now replaced by Drop Eligibility Indicator (DEI)
- VLAN Identifier (12 bits ⇒ 4095 VLANs)
- Switches forward based on MAC address + VLAN ID Unknown addresses are flooded.



Ref: Canonical vs. MSB Addresses, http://support.lexmark.com/index?page=content&id=HO1299&locale=en&userlocale=EN_US

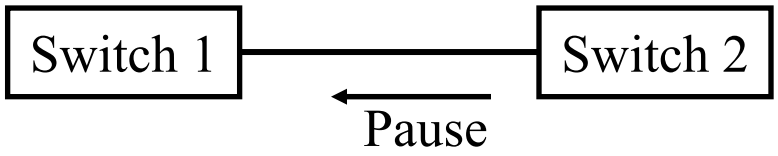
Ref: G. Santana, "Data Center Virtualization Fundamentals," Cisco Press, 2014, ISBN:1587143240

Data Center Bridging (DCB)

- Goal: To enable storage traffic over Ethernet
- Four Standards:
  - Priority-based Flow Control (IEEE 802.1Qbb-2011)
  - Enhanced Transmission Selection (IEEE 802.1Qaz-2011)
  - Congestion Control (IEEE 802.1Qau-2010)
  - Data Center Bridging Exchange (IEEE 802.1Qaz-2011)

Ref: M. Hagen, "Data Center Bridging Tutorial," http://www.iol.unh.edu/services/testing/dcb/training/DCB-Tutorial.pdf

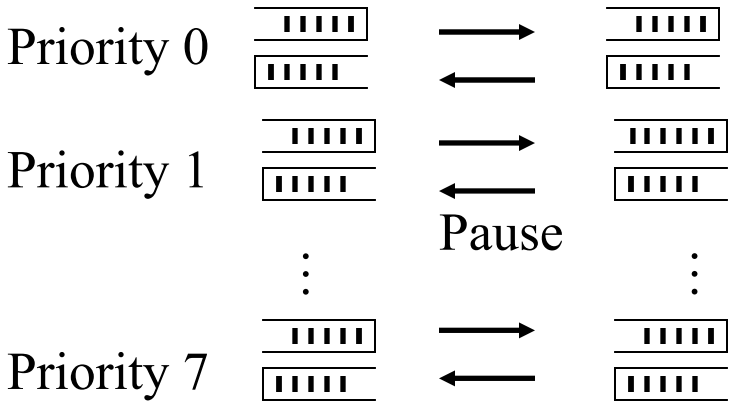
Ethernet Flow Control: Pause Frame



- Defined in IEEE 802.3x-1997. A form of on-off flow control.
- A receiving switch can stop the adjoining sending switch by sending a "Pause" frame.
  Stops the sender from sending any further information for a time specified in the pause frame.
- The frame is addressed to a standard (well-known) multicast address. This address is acted upon but not forwarded.
- Stops all traffic. Causes congestion backup.

Ref: http://en.wikipedia.org/wiki/Ethernet_flow_control

Priority-based Flow Control (PFC)

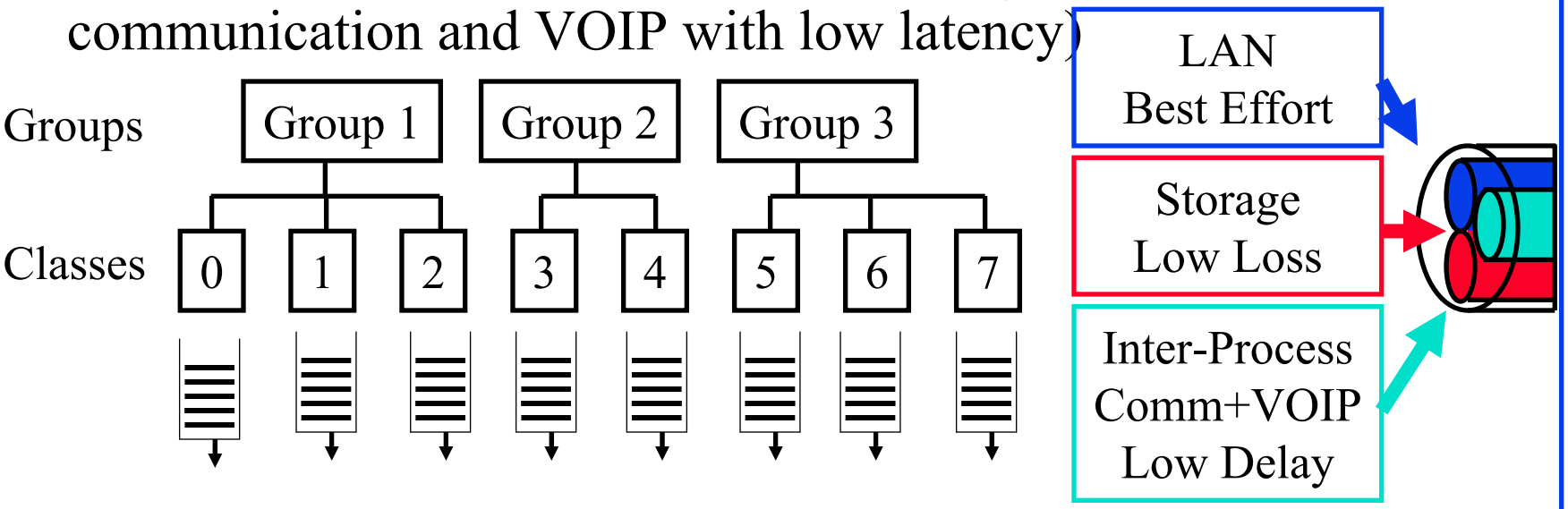


- IEEE 802.1Qbb-2011
- IEEE 802.1Qbb-2011 allows any single priority to be stopped. Others keep sending

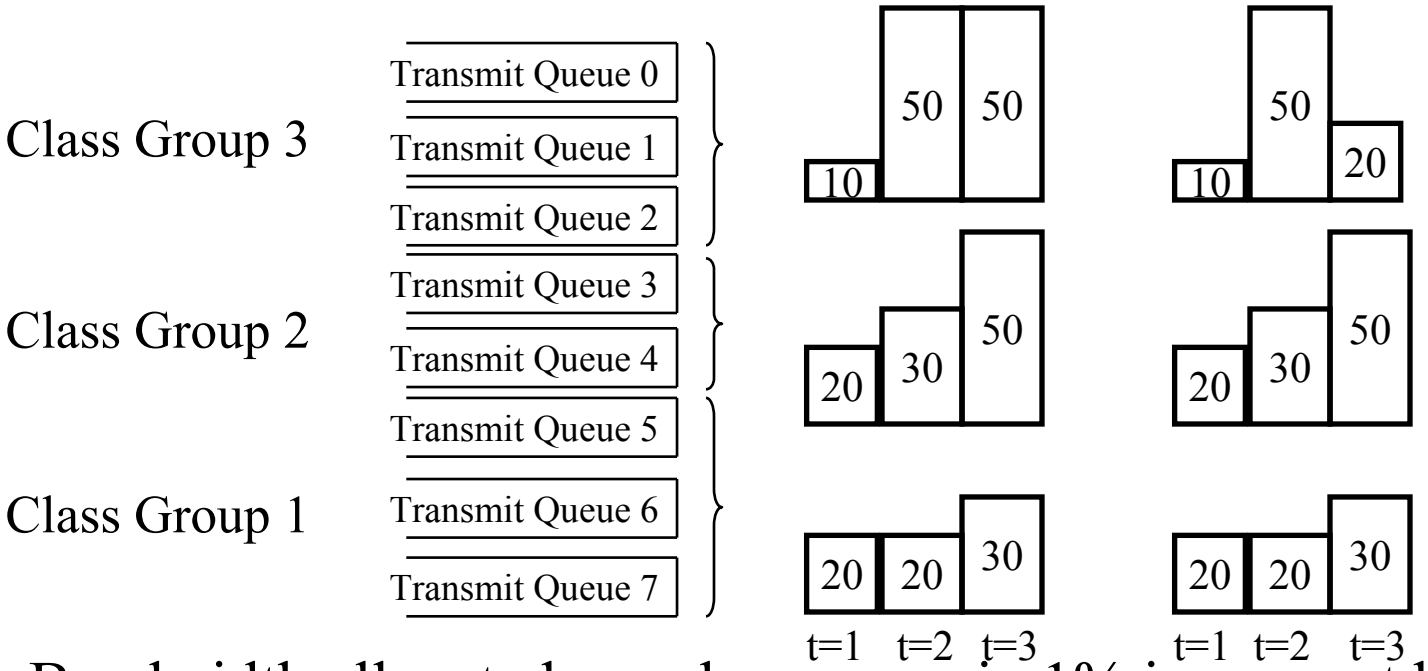
Ref: J. L. White, "Technical Overview of Data Center Networks," SNIA, 2013, http://www.snia.org/sites/default/education/tutorials/2012/fall/networking/JosephWhite_Technical%20Overview%20of%20Data%20Center%20Networks.pdf

Enhanced Transmission Selection

- IEEE 802.1Qaz-2011
- Goal: Guarantee bandwidth for applications sharing a link
- Traffic is divided in to 8 classes (not priorities)
- The classes are grouped.
- Standard requires min 3 groups: 1 with PFC (Storage with low loss), 1 W/O PFC (LAN), 1 Strict Priority (Inter-process communication and VOIP with low latency)

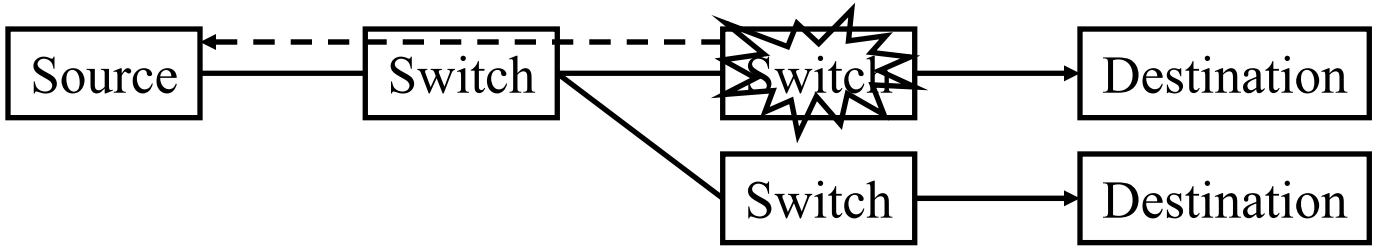


ETS (Cont)



- Bandwidth allocated per class group in 1% increment but 10% precision (±10% error).
- Max 75% allocated ⇒ Min 25% best effort
- Fairness within a group
- All unused bandwidth is available to all classes wanting more bandwidth. Allocation algorithm **not** defined.
- Example: Group 1=20%, Group 2=30%

Quantized Congestion Notification (QCN)



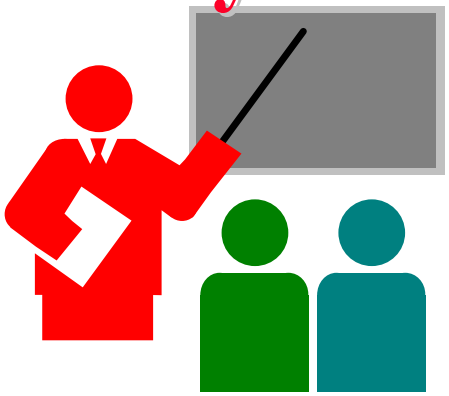
- IEEE 802.1Qau-2010 Dynamic Congestion Notification
- A source quench message is sent by the congested switch direct to the source. The source reduces its rate for that flow.
- Sources need to keep per-flow states and control mechanisms
- Easy for switch manufacturers but complex for hosts.
  Implemented in switches but not in hosts ⇒ Not effective.
- The source may be a router in a subnet and not the real source
  ⇒ Router will drop the traffic. QCN does not help in this case.

Ref: *I. Pepelnjak,* "DCB Congestion Notification (802.1Qau)," http://blog.ipspace.net/2010/11/data-center-bridging-dcb-congestion.html

DCBX

- Data Center Bridging eXchange, IEEE 802.1Qaz-2011
- Uses LLDP (Link Level Discovery Protocol) to negotiate quality metrics and capabilities for Priority-based Flow Control, Enhanced Transmission Selection, and Quantized Congestion Notification
- New TLV's
  - Priority group definition
  - Group bandwidth allocation
  - PFC enablement per priority
  - QCN enablement
  - DCB protocol profiles
  - FCoE and iSCSI profiles

Summary of Part II



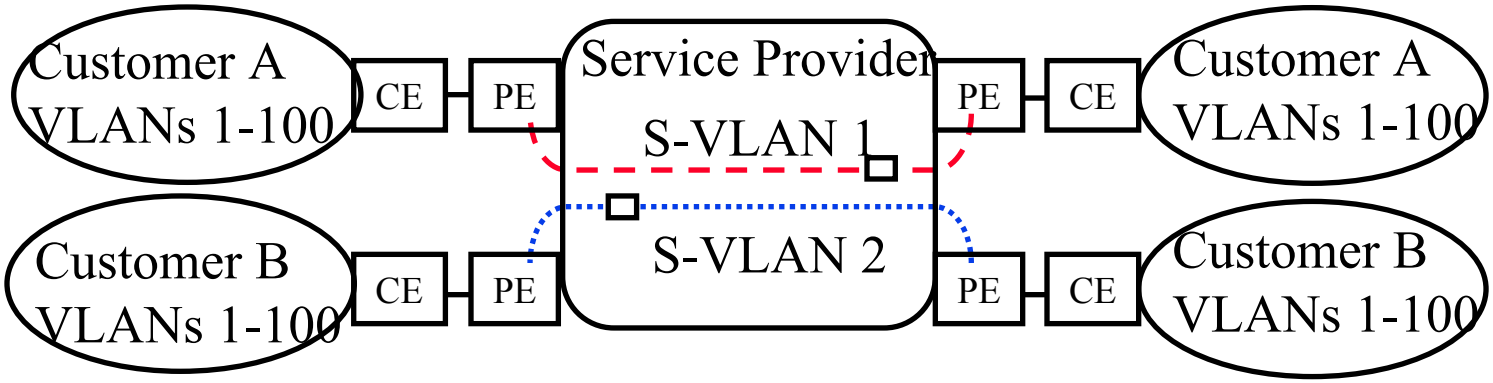
1. Ethernet's use of IDs as addresses makes it very easy to move systems in the data center ⇒ Keep traffic on the same Ethernet
2. Spanning tree is wasteful of resources and slow.
   Ethernet now uses shortest path bridging (similar to OSPF)
3. VLANs allow different non-trusting entities to share an Ethernet network
4. Data center bridging extensions reduce the packet loss by enhanced transmission selection and Priority-based flow control

Part III: Carrier Ethernet for Data Centers

1. Provider Bridges (PB) or Q-in-Q
2. Provider Backbone Bridges (PBB) or MAC-in-MAC
3. Provider Backbone Bridges with Traffic Engineering (PBB-TE)

Note: Although these technologies were originally developed for carriers, they are now used inside multi-tenant data centers (clouds)

Ethernet Provider Bridge (PB)

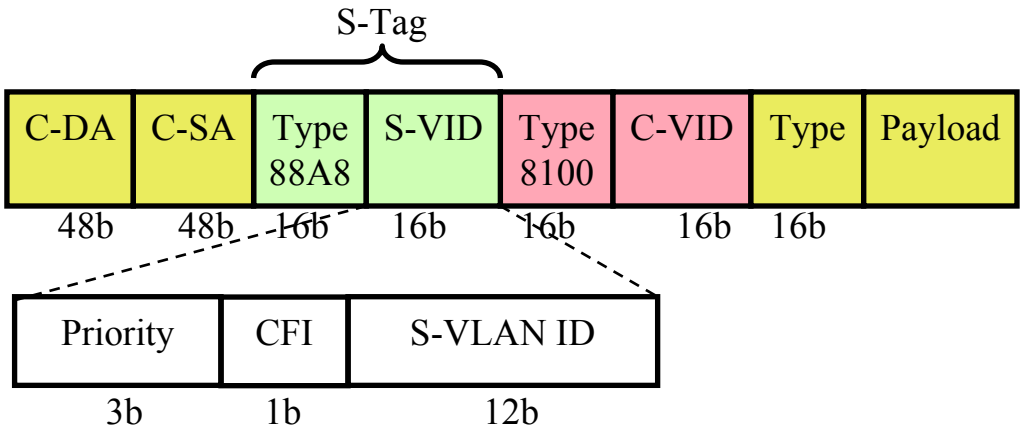


- IEEE 802.1ad-2005 incorporated in IEEE 802.1Q-2011
- Problem: Multiple customers may have the same VLAN ID. How to keep them separate?
- Solutions:
  1. VLAN translation: Change customer VLANs to provider VLANs and back
  2. VLAN Encapsulation: Encapsulate customer frames

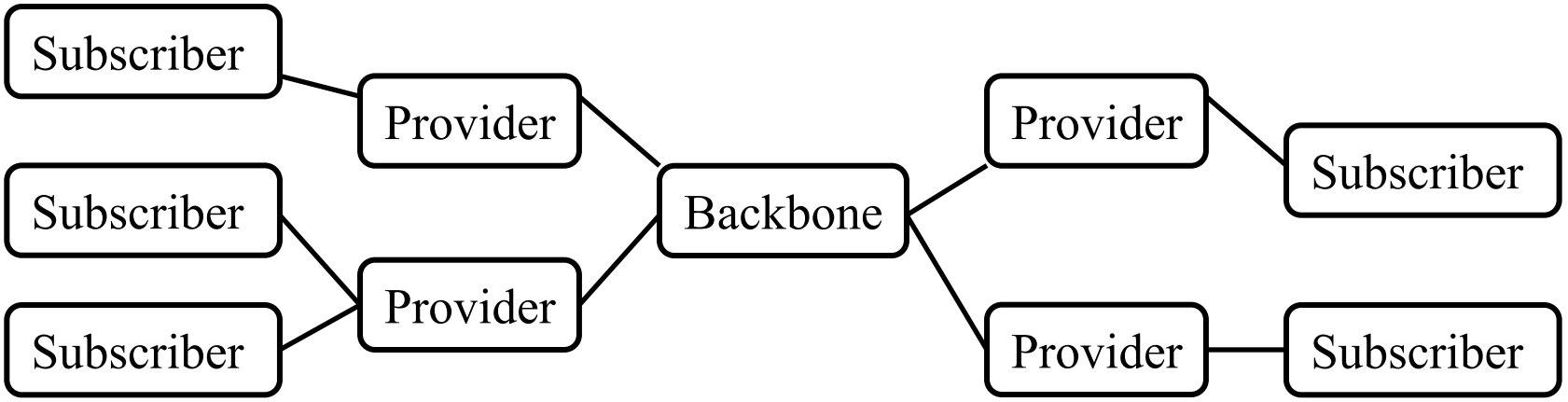
Ref: D. Bonafede, "Metro Ethernet Network," http://www.cicomra.org.ar/cicomra2/asp/TUTORIAL-%20Bonafede.pdf
Ref: P. Thaler, et al., "IEEE 802.1Q," IETF tutorial, March 10 2013, http://www.ietf.org/meeting/86/tutorials/86-IEEE-8021-Thaler.pdf

Provider Bridge (Cont)

- Q-in-Q Encapsulation: Provider inserts a service VLAN tag VLAN translation Changes VLANs using a table
- Allows 4K customers to be serviced. Total 16M VLANs
- 8 Traffic Classes using Differentiated Services Code Points (DSCP) for Assured Forwarding



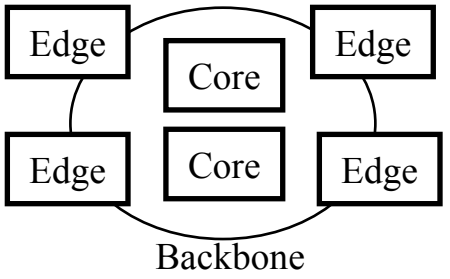
Provider Backbone Network (PBB)

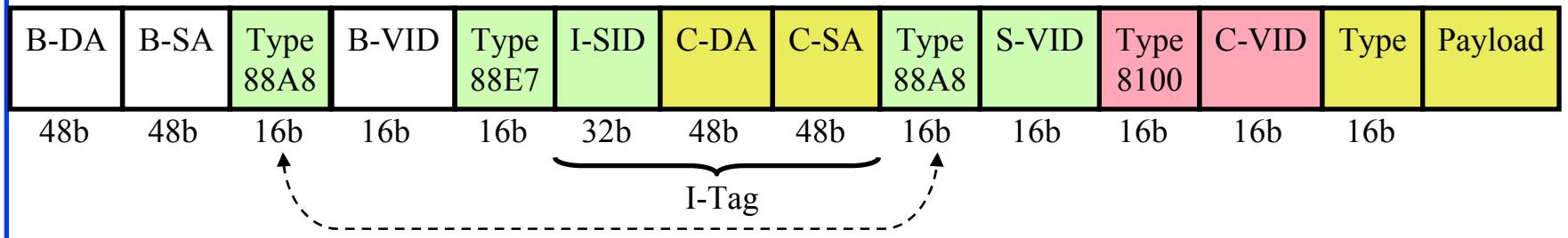


- Problem: Number of MAC addresses passing through backbone bridges is too large for all core bridge to remember
  Broadcast and flooded (unknown address) frames give unwanted traffic and security issues
- Solution: IEEE 802.1ah-2008 now in 802.1Q-2011
- Add new source/destination MAC addresses pointing to ingress backbone bridge and egress backbone bridge
  ⇒Core bridges only know edge bridge addresses

MAC-in-MAC Frame Format

- Provider backbone edge bridges (PBEB) forward to other PBEB's and learn customer MAC addresses
  ⇒ PB *core* bridges do not learn customer MACs
- B-DA = Destination backbone bridge address Determined by Customer Destination Address
- Backbone VLANs delimit the broadcast domains in the backbone



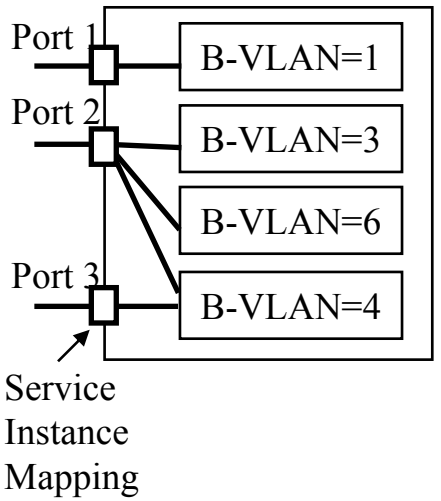


- PBB Core switches forward based on Backbone Destination Bridge Address and Backbone-VLAN ID (60 bits)
  Similar to 802.1ad Q-in-Q. Therefore, same EtherType.

PBB Service Instance

- Service instance ID (I-SID) indicates a specific flow
  - All frames on a specific port, or
  - All frames on a specific port with a specific *service* VLAN, or
  - All frames on a specific port with a specific service VLAN and a specific *customer* VLAN

SID	Definition	B-VLAN
1	Port 1	1
20	Port 2, S-VLAN=10	3
33	Port 2, S-VLAN=20	6
401	Port 2, S-VLAN=30, C-VLAN=100	4
502	Port 3, S-VLAN=40, C-VLAN=200	4



Connection Oriented Ethernet

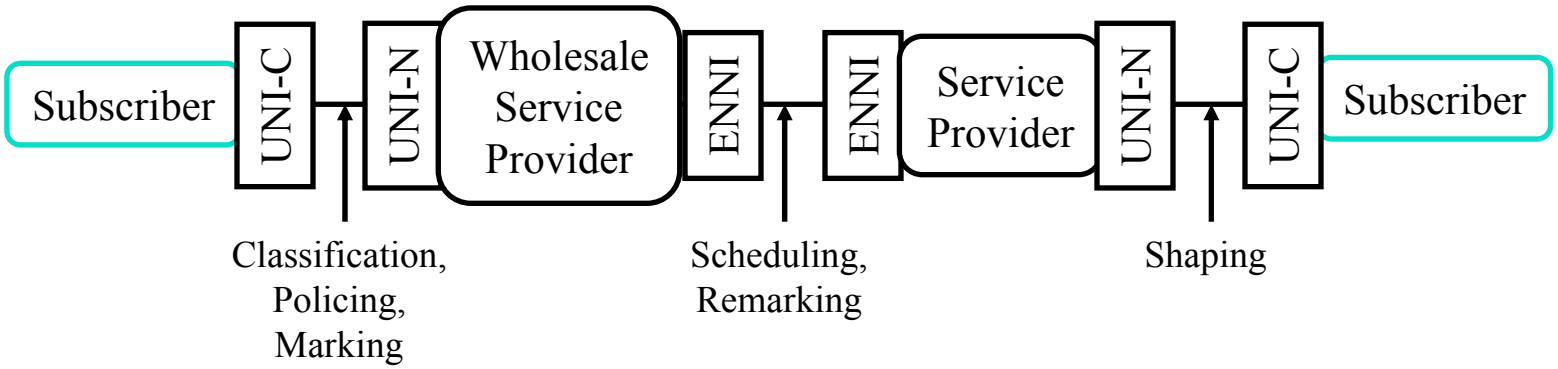
- Connectionless: Path determined at forwarding
  ⇒ Varying QoS
- Connection Oriented: Path determined at provisioning
  - Path provisioned by management ⇒ Deterministic QoS
    - No spanning tree, No MAC address learning,
    - Frames forwarded based on VLAN Ids and Backbone bridges addresses
    - Path not determined by customer MAC addresses and other customer fields ⇒ More Secure
  - Reserved bandwidth per EVC
  - Pre-provisioned Protection path ⇒ Better availability

PBB-TE

- Provider Backbone Bridges with Traffic Engineering (PBB-TE)
- IEEE 802.1Qay-2009 now in 802.1Q-2011
- Provides connection oriented P2P (E-*Line*) Ethernet service
- For PBB-TE traffic VLANs:
  - Turn off MAC learning
  - Discard frames with unknown address and broadcasts.
    ⇒ No flooding
  - Disable Spanning Tree Protocol.
  - Add protection path switching for each direction of the trunk
- Switch forwarding tables are administratively populated using management
- Same frame format as with MAC-in-MAC. No change.

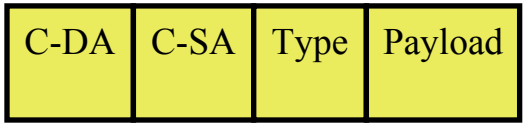
PBB-TE QoS

- Guarantees QoS ⇒ No need for MPLS or SONET/SDH
- UNI traffic is classified by Port, Service VLAN ID, Customer VLAN ID, priority, Unicast/Multicast
- UNI ports are *policed* ⇒ Excess traffic is dropped
  No policing at NNI ports. Only remarking, if necessary.
- Traffic may be marked and remarked at both UNI and NNI

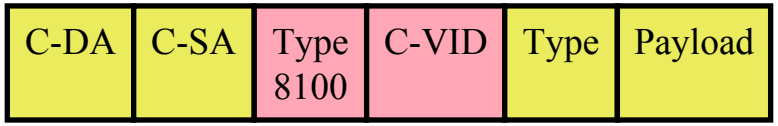


Ethernet Tagged Frame Format Evolution

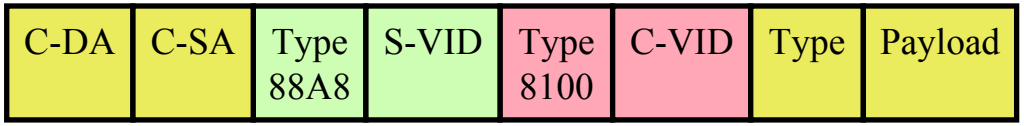
- Original Ethernet



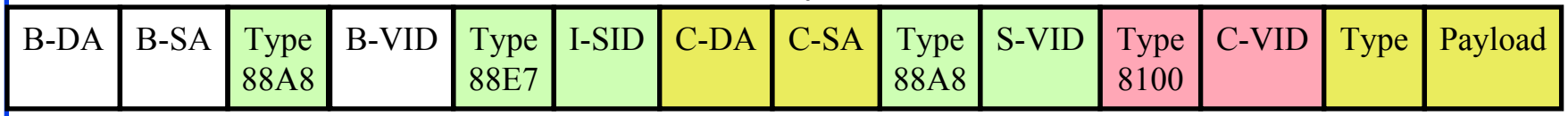
- IEEE 802.1Q VLAN



- IEEE 802.1ad PB



- IEEE 802.1ah PBB or 802.1Qay PBB-TE

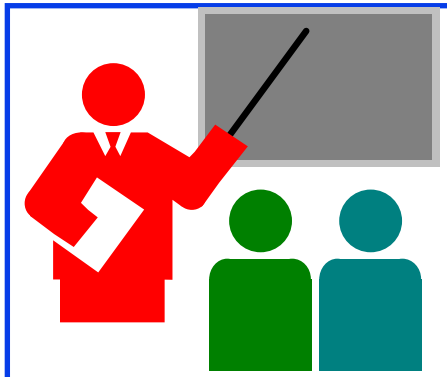


Tag Type	Value
Customer VLAN	8100
Service VLAN or Backbone VLAN	88A8
Backbone Service Instance	88E7

Comparison of Technologies

	Basic Ethernet	MPLS	PB	PBB-TE
Resilience	No	Protection Fast Reroute	SPB/LAG	Protection Fast Reroute
Security	No	Circuit Based	VLAN	Circuit Based
Multicast	Yes	Inefficient	Yes	No. P2P only
QoS	Priority	Diffserve	Diffserve+ Guaranteed	Diffserve+ Guaranteed
Legacy Services	No	Yes (PWE3)	No	No
Traffic Engineering	No	Yes	No	Yes
Scalability	Limited	Complex	Q-in-Q	Q-in-Q+ Mac-in-MAC
Cost	Low	High	Medium	Medium
OAM	No	Some	Yes	Yes

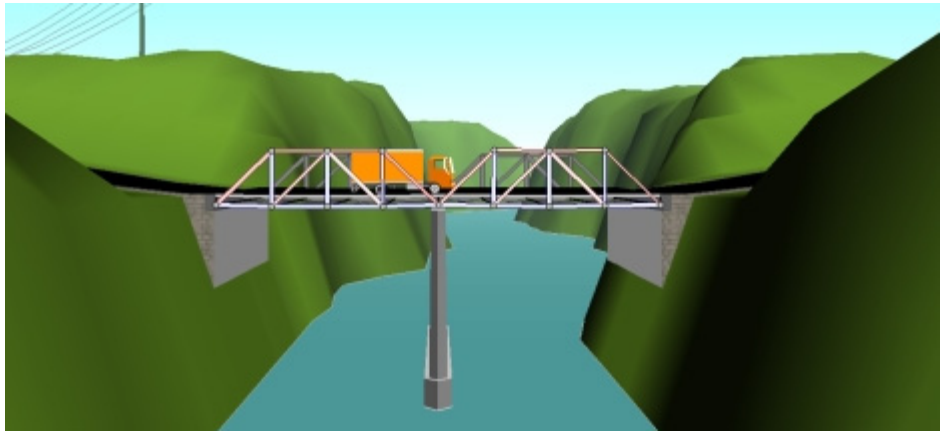
Ref: Bonafede
Washington University in St. Louis



Summary of Part III

1. PB Q-in-Q extension allows Internet/Cloud service providers to allow customers to have their own VLAN IDs
2. PBB MAC-in-MAC extension allows customers/tenants to have their own MAC addresses and allows service providers to not have to worry about them in the core switches
3. PBB allows very large Ethernet networks spanning over several backbone carriers
4. PBB-TE extension allows connection oriented Ethernet with QoS guarantees and protection

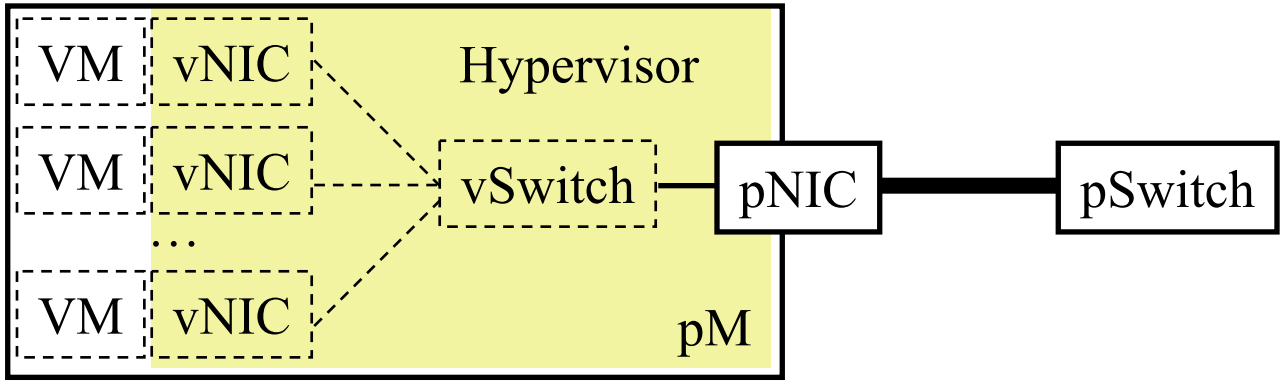
Part IV: Virtual Bridging



1. Virtual Bridges to connect virtual machines
2. IEEE Virtual Edge Bridging Standard
3. Single Root I/O Virtualization (SR-IOV)
4. Aggregating Bridges and Links: VSS and vPC
5. Bridges with massive number of ports: VBE

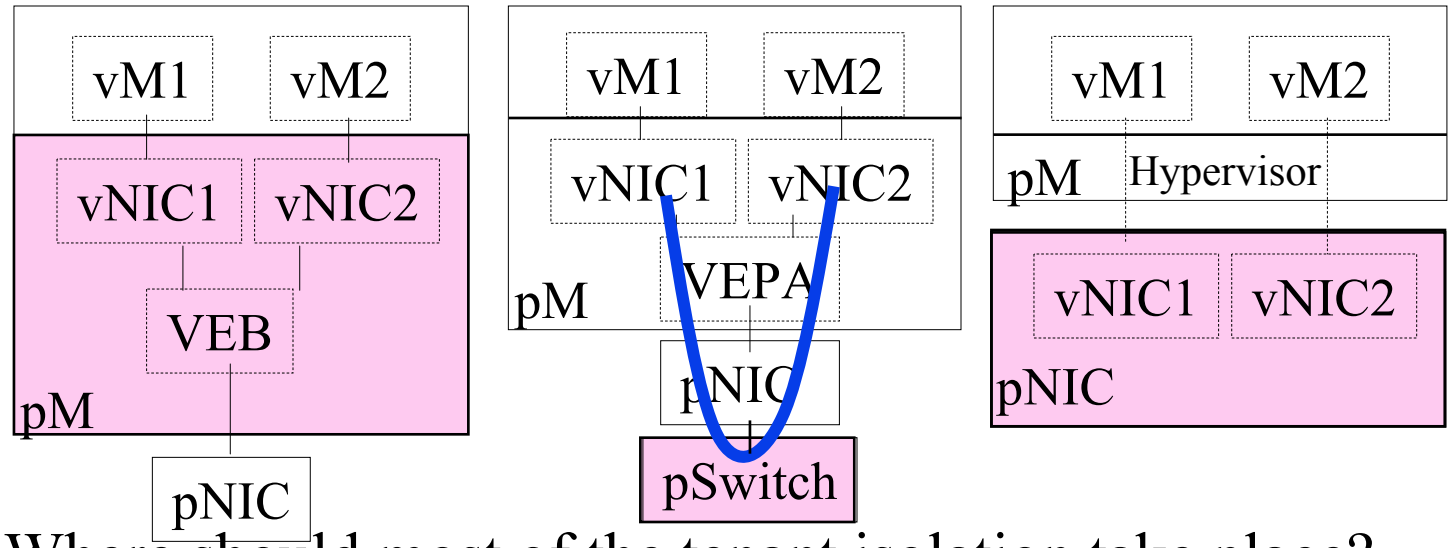
vSwitch

- **Problem**: Multiple VMs on a server need to use one physical network interface card (pNIC)
- **Solution**: Hypervisor creates multiple vNICs connected via a virtual switch (vSwitch)
- pNIC is controlled by hypervisor and not by any individual VM
- **Notation**: From now on prefixes p and v refer to physical and virtual, respectively. For VMs only, we use upper case V.



Ref: G. Santana, "Datacenter Virtualization Fundamentals," Cisco Press, 2014, ISBN: 1587143240

Virtual Bridging

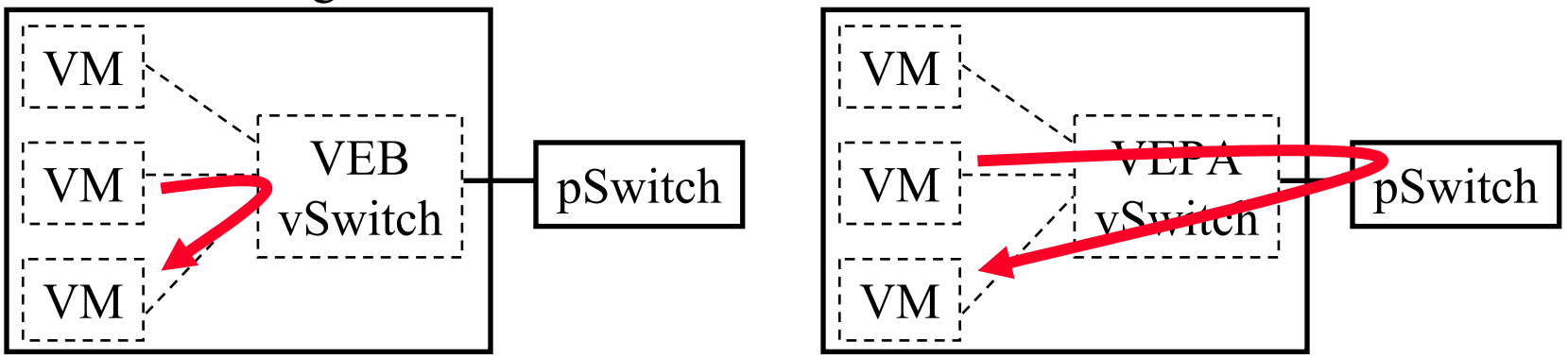


Where should most of the tenant isolation take place?

1. VM vendors: S/W NICs in Hypervisor w Virtual Edge Bridge (**VEB**)(overhead, not ext manageable, not all features)
2. Switch Vendors: Switch provides virtual channels for inter-VM Communications using virtual Ethernet port aggregator (**VEPA**): **802.1Qbg** (s/w upgrade)
3. NIC Vendors: NIC provides virtual ports using Single-Route I/O virtualization (**SR-IOV**) on PCI bus

Virtual Edge Bridge

- IEEE 802.1Qbg-2012 standard for vSwitch
- Two modes for vSwitches to handle *local* VM-to-VM traffic:
  - **Virtual Edge Bridge (VEB):** Switch internally.
  - **Virtual Ethernet Port Aggregator (VEPA):** Switch externally
- VEB
  - could be in a hypervisor or network interface card
  - may learn or may be configured with the MAC addresses
  - VEB may participate in spanning tree or may be configured\
  - Advantage: No need for the external switch in some cases



Virtual Ethernet Port Aggregator (VEPA)

- VEPA simply relays all traffic to an external bridge
- External bridge forwards the traffic. Called "*Hairpin Mode*."
  Returns local VM traffic back to VEPA
  Note: Legacy bridges do not allow traffic to be sent back to the incoming port within the same VLAN
- **VEPA Advantages**:
  - Visibility: External bridge can see VM to VM traffic.
  - Policy Enforcement: Better. E.g., firewall
  - Performance: Simpler vSwitch ⇒ Less load on CPU
  - Management: Easier
- Both VEB and VEPA can be implemented on the same NIC in the same server and can be cascaded.

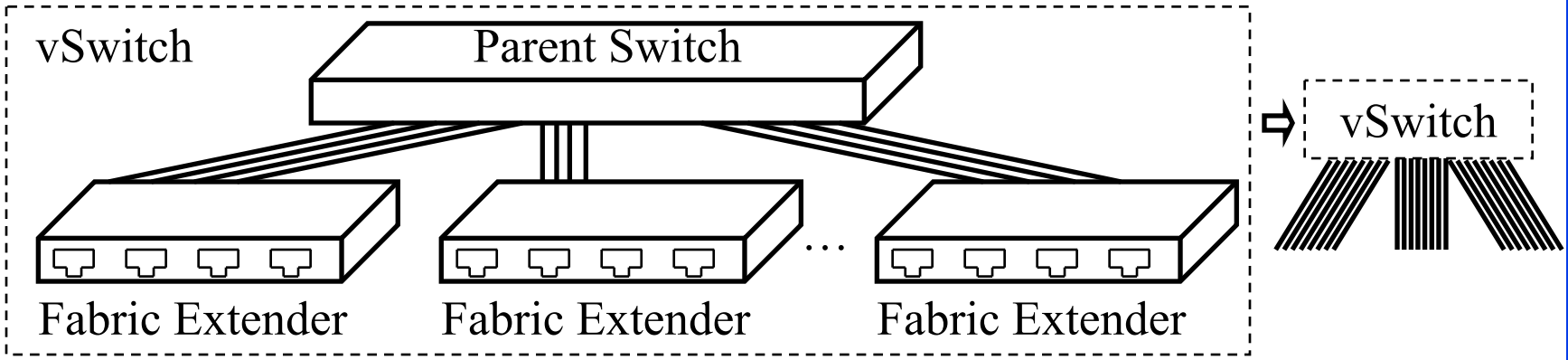
Ref: HP, "Facts about the IEEE 802.1Qbg proposal," Feb 2011, 6pp.,
http://h20000.www2.hp.com/bc/docs/support/SupportManual/c02877995/c02877995.pdf

Combining Bridges

- **Problem**:
  - Number of VMs is growing very fast
  - Need switches with very large number of ports
  - Easy to manage one bridge than 100 10-port bridges
  - How to make very large switches ~1000 ports?
- **Solutions**: Multiple pSwitches to form a single switch
  1. Fabric Extension (FEX)
  2. Virtual Bridge Port Extension (VBE)

Fabric Extenders

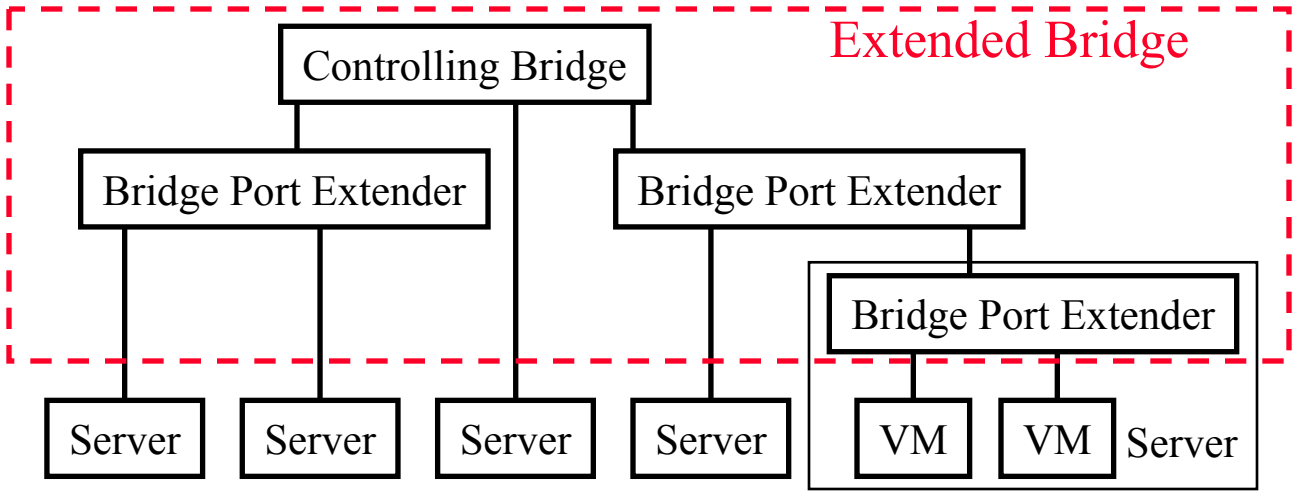
- Fabric extenders (FEX) consists of ports that are managed by a remote parent switch
- 12 Fabric extenders, each with 48 host ports, connected to a parent switch via 4-16 10 Gbps interfaces to a parent switch provide a virtual switch with 576 host ports
  ⇒ **Chassis Virtualization**
- All software updates/management, forwarding/control plane is managed centrally by the parent switch.
- A FEX can have an active and a standby parent.

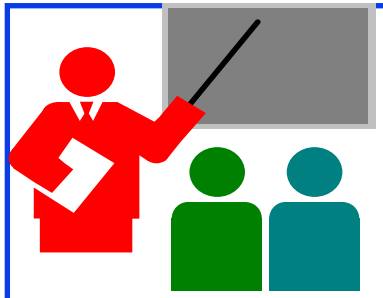


Ref: P. Beck, et al., "IBM and Cisco: Together for a World Class Data Center," IBM Red Book, 2013, 654 pp., ISBN: 0-7384-3842-1, http://www.redbooks.ibm.com/redbooks/pdfs/sg248105.pdf

Virtual Bridge Port Extension (VBE)

- IEEE 802.1BR-2012 standard for fabric extender functions
- Specifies how to form an extended bridge consisting of a controlling bridge and Bridge Port Extenders
- Extenders can be cascaded.
- Some extenders may be in a vSwitch in a server hypervisor.
- All traffic is relayed by the controlling bridge
  ⇒ Extended bridge is a bridge.

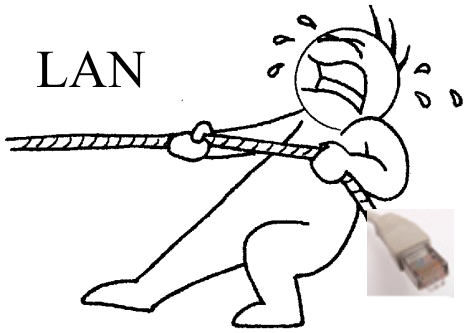




Summary of Part IV

1. Network virtualization includes virtualization of NICs, Bridges, Routers, and L2 networks.
2. Virtual Edge Bridge (VEB) vSwitches switch internally while Virtual Ethernet Port Aggregator (VEPA) vSwitches switch externally.
3. Fabric Extension and Virtual Bridge Extension (VBE) allows creating switches with a large number of ports using port extenders (which may be vSwitches)

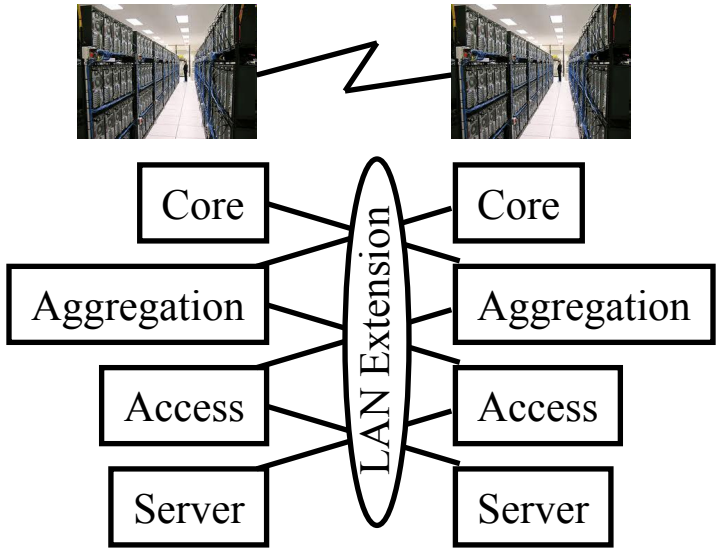
Part V: LAN Extension and Partitioning



1. Transparent Interconnection of Lots of Links (TRILL)
2. Network Virtualization using GRE (NVGRE)
3. Virtual eXtensible LANs (VXLAN)
4. Stateless Transport Tunneling Protocol (STT)

Challenges of LAN Extension

- **Broadcast storms**: Unknown and broadcast frames may create excessive flood
- **Loops**: Easy to form loops in a large network.
- **STP Issues**:
  - High spanning tree diameter: More than 7.
  - Root can become bottleneck and a single point of failure
  - Multiple paths remain unused
- **Tromboning**: Dual attached servers and switches generate excessive cross traffic
- **Security**: Data on LAN extension must be encrypted



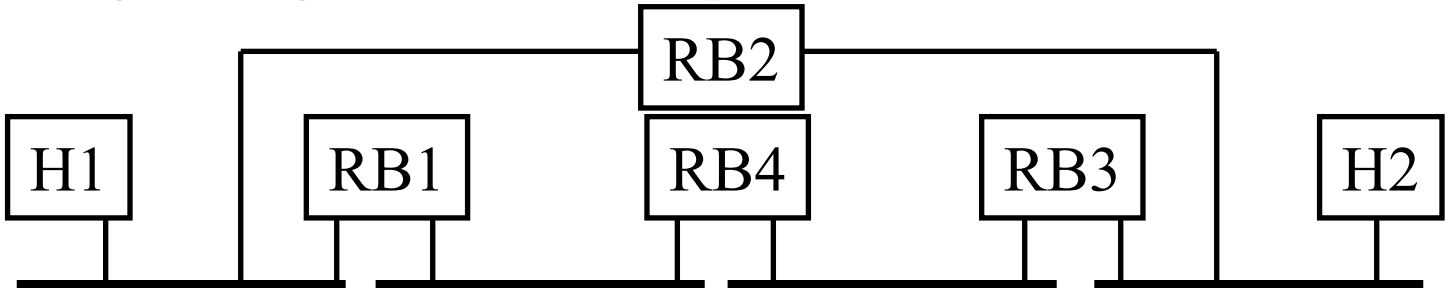
TRILL

- Transparent Interconnection of Lots of Links
- Allows a large campus to be a single extended LAN
- LANs allow free mobility inside the LAN but:
  - Inefficient paths using Spanning tree
  - Inefficient link utilization since many links are disabled
  - Inefficient link utilization since multipath is not allowed.
  - Unstable: small changes in network ⇒ large changes in spanning tree
- IP subnets are not good for mobility because IP addresses change as nodes move and break transport connections, but:
  - IP routing is efficient, optimal, and stable
- Solution: Take the best of both worlds
  ⇒ Use MAC addresses and IP routing

Ref: RFCs 5556, 6325, 6326, 6327, 6361, 6439

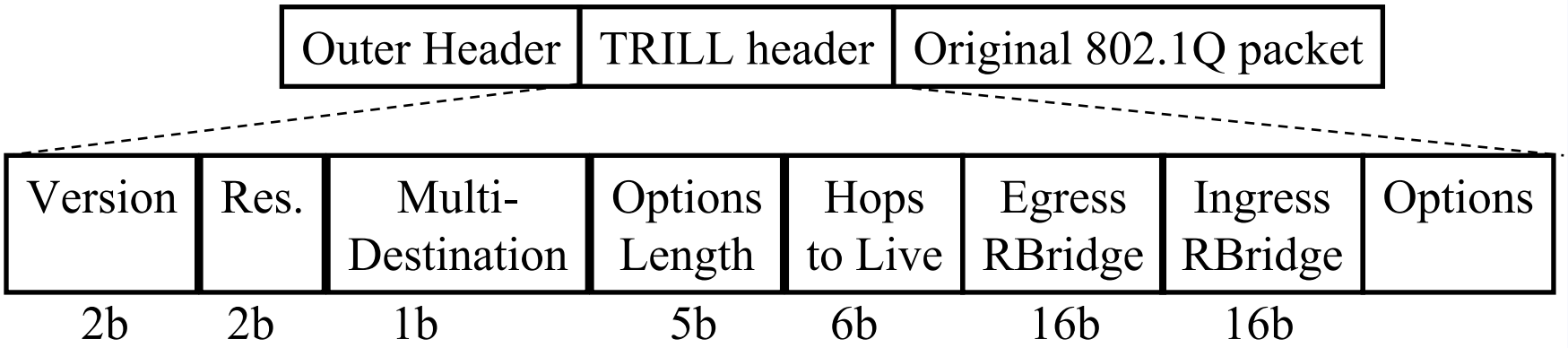
TRILL Architecture

- Routing Bridges (RBridges) encapsulate L2 frames and route them to destination RBridges which decapsulate and forward
- Header contains a hop-limit to avoid looping
- RBridges run IS-IS to compute pair-wise optimal paths for unicast and distribution trees for multicast
- RBridge learn MAC addresses by source learning and by exchanging their MAC tables with other RBridges
- Each VLAN on the link has one (and only one) designated RBridge using IS-IS election protocol



Ref: R. Perlman, "RBridges: Transparent Routing," Infocom 2004

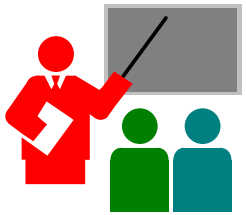
TRILL Encapsulation Format



- For outer headers both PPP and Ethernet headers are allowed. PPP for long haul.
- Outer Ethernet header can have a VLAN ID corresponding to the VLAN used for TRILL.
- Priority bits in outer headers are copied from inner VLAN

TRILL Features

- Transparent: No change to capabilities. Broadcast, Unknown, Multicast (**BUM**) support. Auto-learning.
- Zero Configuration: RBridges discover their connectivity and learn MAC addresses automatically
- Hosts can be multi-homed
- VLANs are supported
- Optimized route
- No loops
- Legacy bridges with spanning tree in the same extended LAN

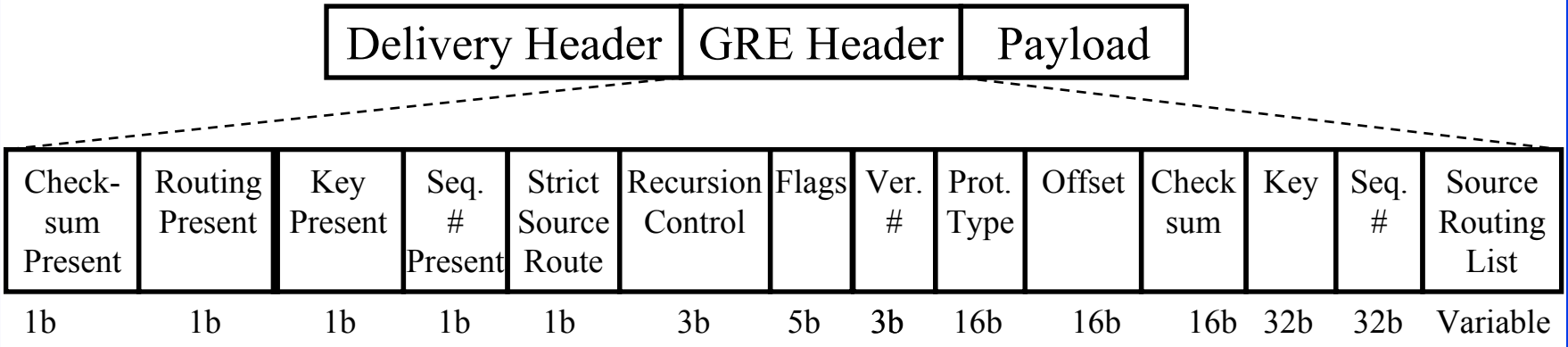


TRILL: Summary

- TRILL allows a large campus to be a single Extended LAN
- Packets are encapsulated and routed using IS-IS routing

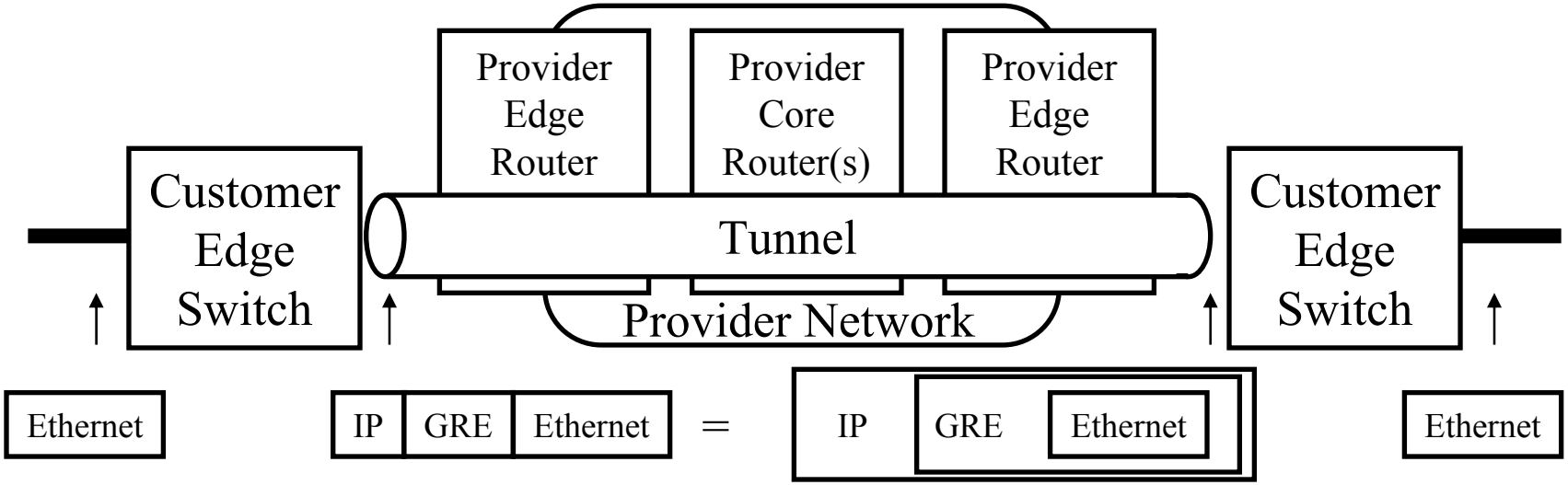
GRE

- Generic Routing Encaptulation (RFC 1701/1702)
- Generic ⇒ X over Y for any X or Y
- Over IPv4, GRE packets use a protocol type of 47
- Optional Checksum, Loose/strict Source Routing, Key
- Key is used to authenticate the source
- Recursion Control: # of additional encapsulations allowed.
  0 ⇒ Restricted to a single provider network ⇒ end-to-end
- Offset: Points to the next source route field to be used
- IP or IPSec are commonly used as delivery headers



NVGRE

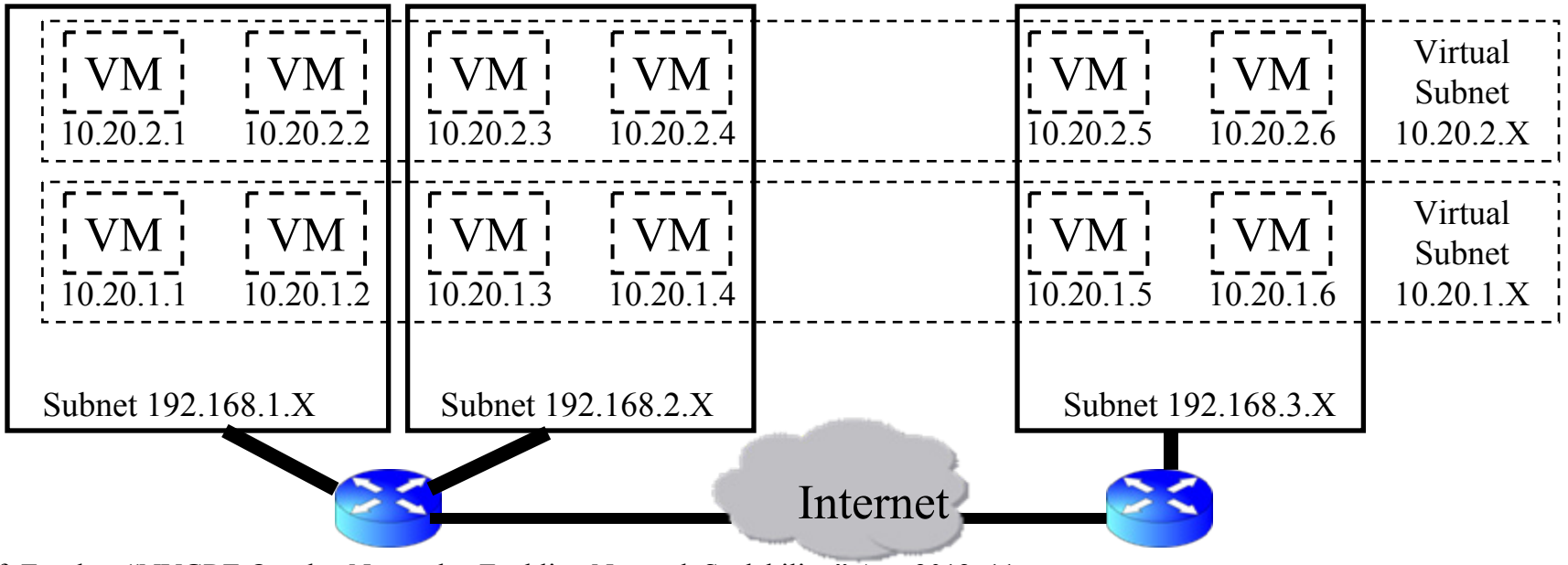
- Network Virtualization using GRE
  ⇒ Ethernet over GRE over IP (point-to-point)
- A unique 24-bit Virtual Subnet Identifier (VSID) is used as the lower 24-bits of GRE key field ⇒ 2^{24} tenants can share
- Unique IP multicast address is used for BUM (Broadcast, Unknown, Multicast) traffic on each VSID
- Equal Cost Multipath (ECMP) allowed on point-to-point tunnels



Ref: M. Sridharan, "MVGRE: Network Virtualization using GRE," Aug 2013, http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-03

NVGRE (Cont)

- In a cloud, a pSwitch or a vSwitch can serve as tunnel endpoint
- VMs need to be in the same VSID to communicate
- VMs in different VSIDs can have the same MAC address
- Inner IEEE 802.1Q tag, if present, is removed.



Ref: Emulex, "NVGRE Overlay Networks: Enabling Network Scalability," Aug 2012, 11pp., http://www.emulex.com/artifacts/074d492d-9dfa-42bd-9583-69ca9e264bd3/elx_wp_all_nvgre.pdf

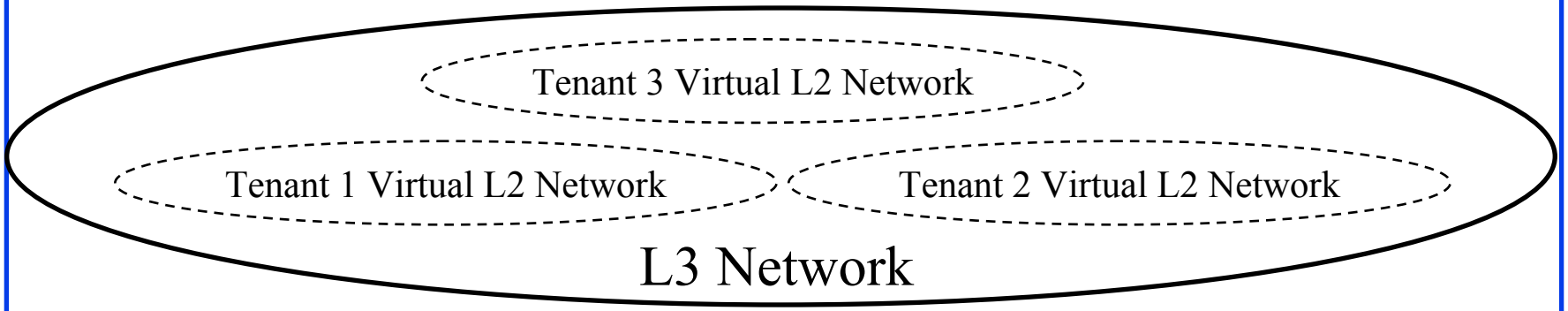
VXLAN

- Virtual eXtensible Local Area Networks (VXLAN)
- L3 solution to isolate multiple tenants in a data center (L2 solution is Q-in-Q and MAC-in-MAC)
- Developed by VMware. Supported by many companies in IETF NVO3 working group
- Problem:
  - 4096 VLANs are not sufficient in a multi-tenant data center
  - Tenants need to control their MAC, VLAN, and IP address assignments ⇒ Overlapping MAC, VLAN, and IP addresses
  - Spanning tree is inefficient with large number of switches ⇒ Too many links are disabled
  - Better throughput with IP equal cost multipath (ECMP)

Ref: M. Mahalingam, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks," draft-mahalingam-dutt-dcops-vxlan-04, May, 8, 2013, http://tools.ietf.org/html/draft-mahalingam-dutt-dcops-vxlan-04

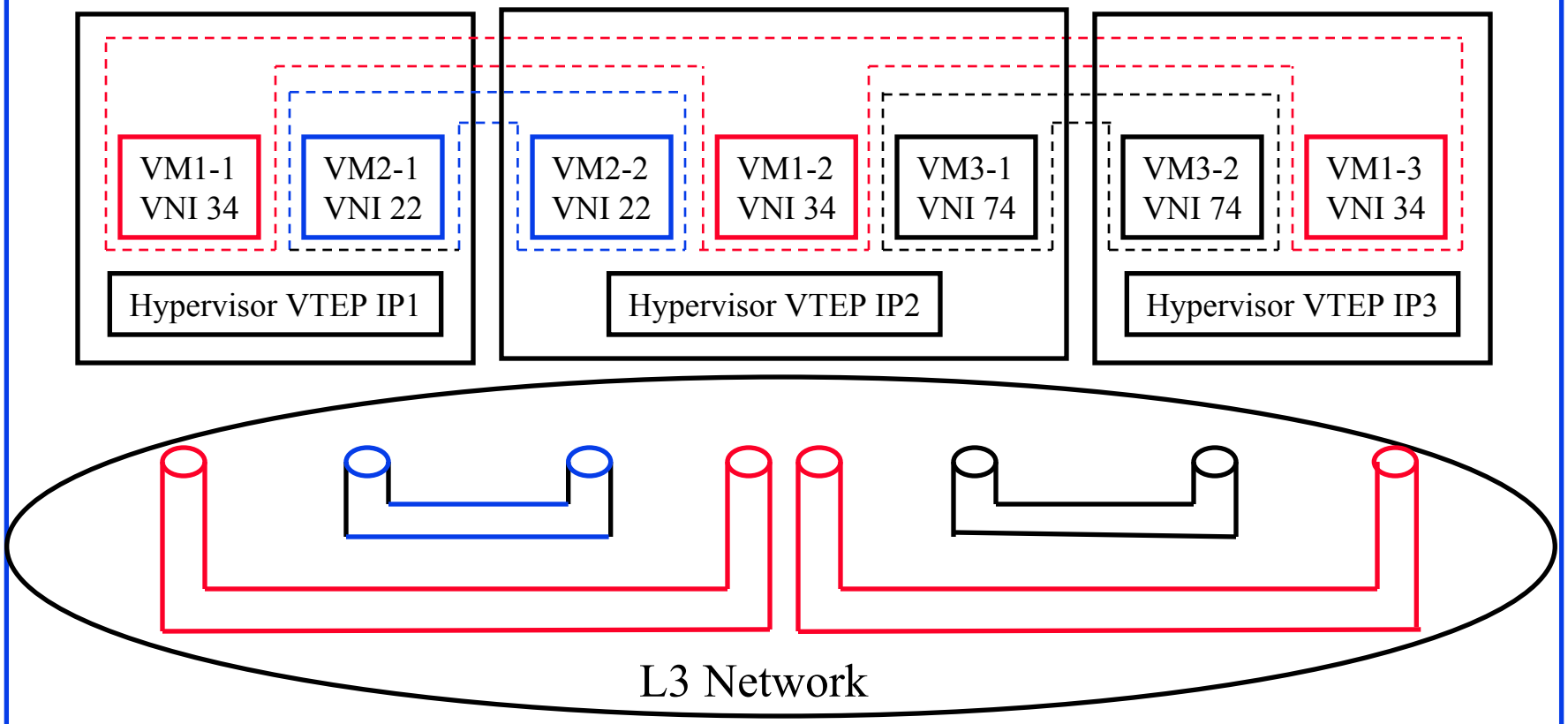
VXLAN Architecture

- Create a virtual L2 overlay (called VXLAN) over L3 networks
- 2^{24} VXLAN Network Identifiers (VNIs)
- Only VMs in the same VXLAN can communicate
- vSwitches serve as VTEP (VXLAN Tunnel End Point).
  ⇒ Encapsulate L2 frames in UDP over IP and send to the destination VTEP(s).
- Segments may have overlapping MAC addresses and VLANs but L2 traffic never crosses a VNI



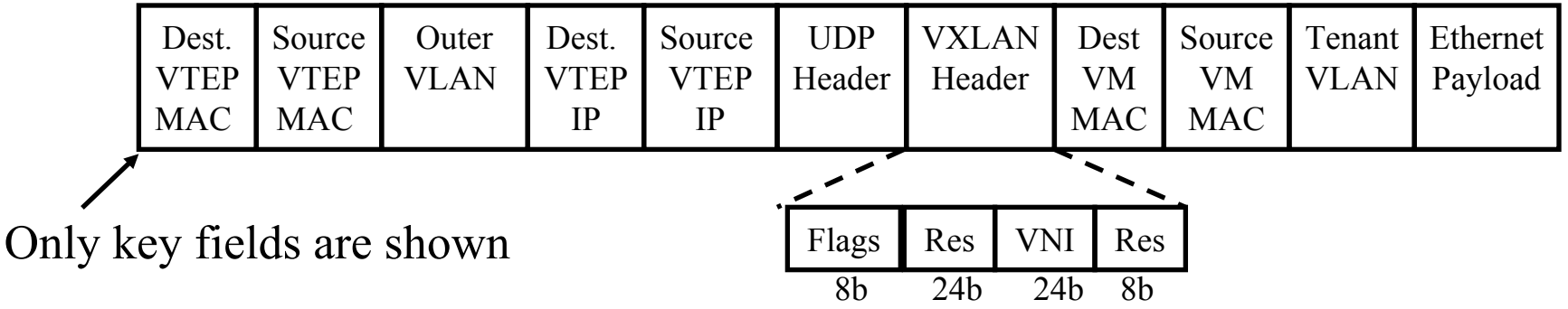
VXLAN Deployment Example

Example: Three tenants. 3 VNIs. 4 Tunnels for unicast.
+ 3 tunnels for multicast (not shown)



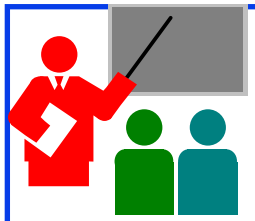
VXLAN Encapsulation Format

- Outer VLAN tag is optional.
  Used to isolate VXLAN traffic on the LAN
- Source VM ARPs to find Destination VM's MAC address.
  All L2 multicasts/unknown are sent via IP multicast.
  Destination VM sends a standard IP unicast ARP response.
- Destination VTEP learns inner-Src-MAC-to-outer-src-IP mapping
  ⇒ Avoids unknown destination flooding for returning responses



VXLAN Encapsulation Format (Cont)

- IGMP is used to prune multicast trees
- 7 of 8 bits in the flag field are reserved.
  I flag bit is set if VNI field is valid
- UDP source port is a hash of the inner MAC header
  $\Rightarrow$ Allows load balancing using Equal Cost Multi Path using L3-L4 header hashing
- VMs are unaware that they are operating on VLAN or VXLAN
- VTEPs need to learn MAC address of other VTEPs and of client VMs of VNIs they are handling.
- A VXLAN gateway switch can forward traffic to/from non-VXLAN networks. Encapsulates or decapsulates the packets.



VXLAN: Summary

- VXLAN solves the problem of multiple tenants with overlapping MAC addresses, VLANs, and IP addresses in a cloud environment.
- A server may have VMs belonging to different tenants
- No changes to VMs. Hypervisors responsible for all details.
- Uses UDP over IP encapsulation to isolate tenants

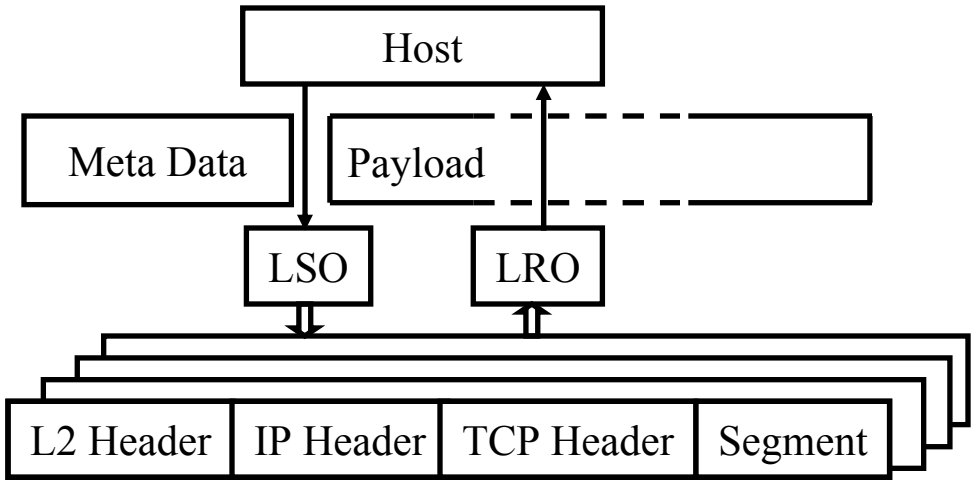
Stateless Transport Tunneling Protocol (STT)

- Ethernet over TCP-Like over IP tunnels.
  GRE, IPSec tunnels can also be used if required.
- Tunnel endpoints may be inside the end-systems (vSwitches)
- Designed for large storage blocks 64kB. Fragmentation allowed.
- Most other overlay protocols use UDP and disallow fragmentation ⇒ Maximum Transmission Unit (MTU) issues.
- TCP-Like: Stateless TCP ⇒ Header identical to TCP (same protocol number 6) but no 3-way handshake, no connections, no windows, no retransmissions, no congestion state ⇒ Stateless Transport (recognized by standard port number).
- Broadcast, Unknown, Multicast (BUM) handled by IP multicast tunnels

Ref: B. Davie and J. Gross, "A Stateless Transport Tunneling Protocol for Network Virtualization (STT)," Apr 2014, http://tools.ietf.org/html/draft-davie-stt-06

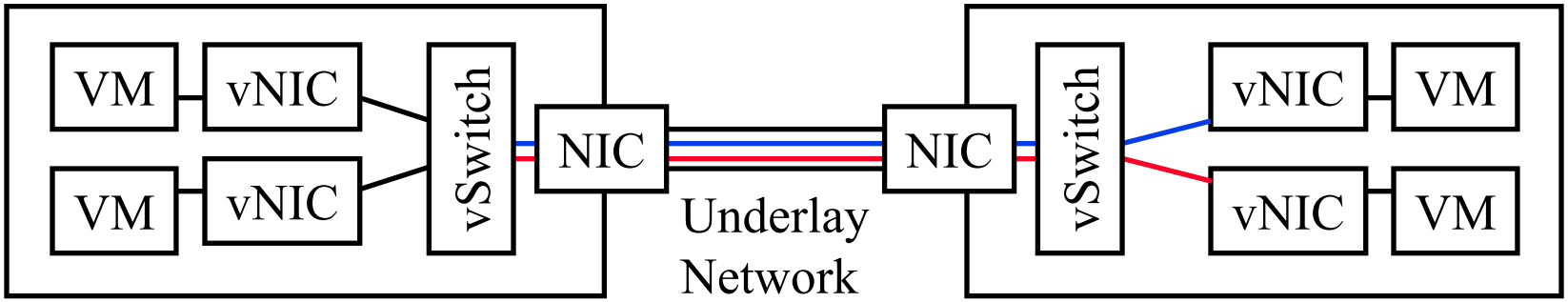
LSO and LRO

- Large Send Offload (LSO): Host hands a large chunk of data to NIC and meta data. NIC makes MSS size segments, adds checksum, TCP, IP, and MAC headers to each segment.
- Large Receive Offload (LRO): NICs attempt to reassemble multiple TCP segments and pass larger chunks to the host. Host does the final reassembly with fewer per packet operations.
- STT takes advantage of LSO and LRO features, if available.
- Using a protocol number other than 6 will not allow LSO/LRO to handle STT



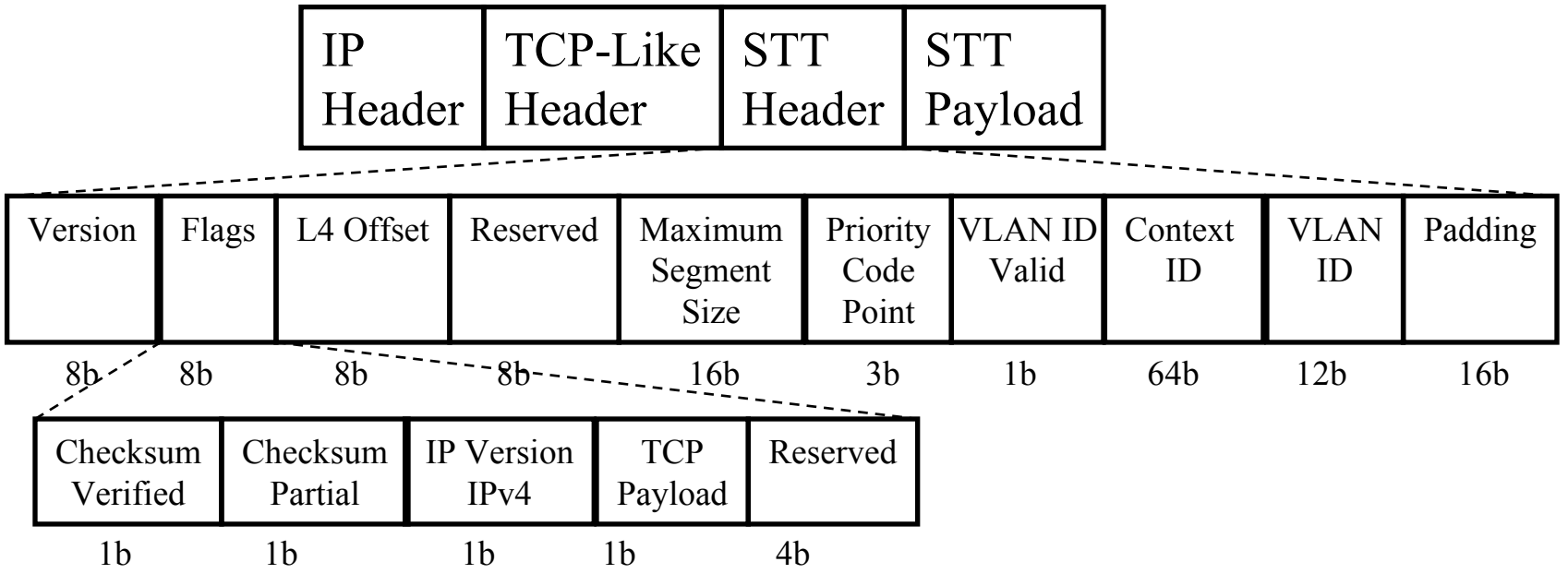
STT Optimizations

- Large data size: Less overhead per payload byte
- Context ID: 64-bit tunnel end-point identifier
- Optimizations:
  - 2-byte padding is added to Ethernet frames to make its size a multiple of 32-bits.
  - Source port is a hash of the inner header ⇒ ECMP with each flow taking different path and all packets of a flow taking one path
- No protocol type field ⇒ Payload assumed to be Ethernet, which can carry any payload identified by protocol type.



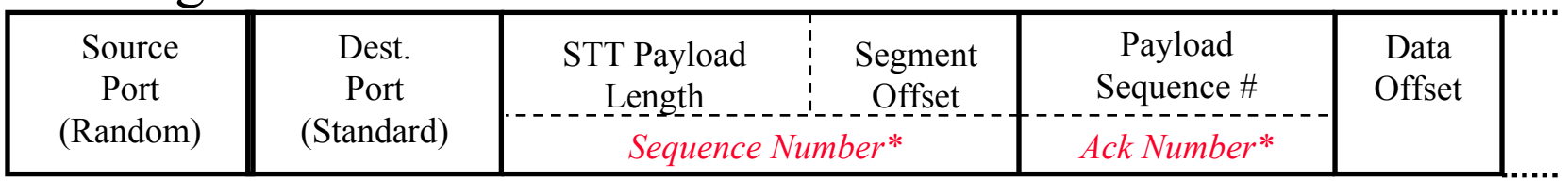
STT Frame Format

- 16-Bit MSS ⇒ 2^{16} B = 64K Byte maximum
- L4 Offset: From the of STT header to the start of encapsulated L4 (TCP/UDP) header ⇒ Helps locate payload quickly
- Checksum Verified: Checksum covers entire payload and valid
- Checksum Partial: Checksum only includes TCP/IP headers



TCP-Like Header in STT

- Destination Port: Standard to be requested from IANA
- Source Port: Selected for efficient ECMP
- Ack Number: STT payload sequence identifier. Same in all segments of a payload
- Sequence Number (32b): Length of STT Payload (16b) + offset of the current segment (16b) ⇒ Correctly handled by NICs with Large Receive Offload (LRO) feature
- No acks. STT delivers partial payload to higher layers.
- Higher layer TCP can handle retransmissions if required.
- Middle boxes will need to be programmed to allow STT pass through

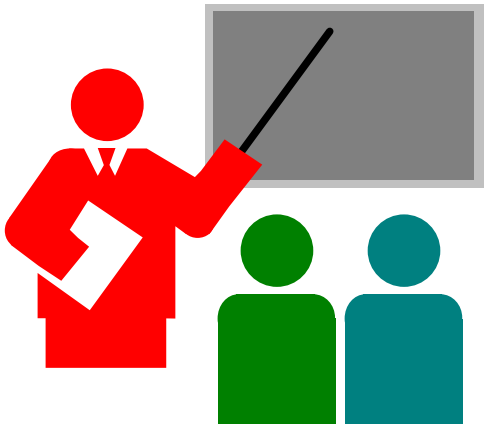


*Different meaning than TCP

STT Summary

- STT solves the problem of *efficient* transport of large 64 KB storage blocks
- Uses Ethernet over TCP-Like over IP tunnels
- Designed for software implementation in hypervisors

Summary of Part V



1. TRILL allows Ethernet to span a large campus using IS-IS encapsulation
2. NVGRE uses Ethernet over GRE for L2 connectivity.
3. VXLAN uses Ethernet over UDP over IP
4. STT uses Ethernet over TCP-like stateless protocol over IP.

Part VI: OpenFlow and Tools

- Planes of Networking
- OpenFlow
- OpenFlow Operation
- OpenFlow Evolution
- OpenFlow Configuration Protocol (OF-Config)
- OpenFlow Notification Framework
- OpenFlow Controllers

Planes of Networking

- **Data Plane**: All activities involving as well as resulting from data packets sent by the end user, e.g.,
  - Forwarding
  - Fragmentation and reassembly
  - Replication for multicasting
- **Control Plane**: All activities that are necessary to perform data plane activities but do not involve end-user data packets
  - Making routing tables
  - Setting packet handling policies (e.g., security)
  - Base station beacons announcing availability of services

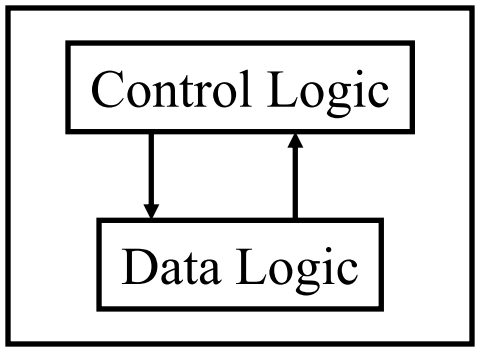
Ref: Open Data Center Alliance Usage Model: Software Defined Networking Rev 1.0,"
http://www.opendatacenteralliance.org/docs/Software_Defined_Networking_Master_Usage_Model_Rev1.0.pdf

Planes of Networking (Cont)

- **Management Plane**: All activities related to provisioning and monitoring of the networks
  - Fault, Configuration, Accounting, Performance and Security (**FCAPS**).
  - Instantiate new devices and protocols (Turn devices on/off)
  - Optional ⇒ May be handled manually for small networks.
- **Services Plane**: Middlebox services to improve performance or security, e.g.,
  - Load Balancers, Proxy Service, Intrusion Detection, Firewalls, SSL Off-loaders
  - Optional ⇒ Not required for small networks

Data vs. Control Logic

- Data plane runs at line rate,
  e.g., 100 Gbps for 100 Gbps Ethernet ⇒ Fast Path
  ⇒ Typically implemented using special hardware,
  e.g., Ternary Content Addressable Memories (TCAMs)
- Some exceptional data plane activities are handled by the CPU in the switch ⇒ Slow path
  e.g., Broadcast, Unknown, and Multicast (BUM) traffic
- All control activities are generally handled by CPU



OpenFlow: Key Ideas

1. Separation of control and data planes
2. Centralization of control
3. Flow based control

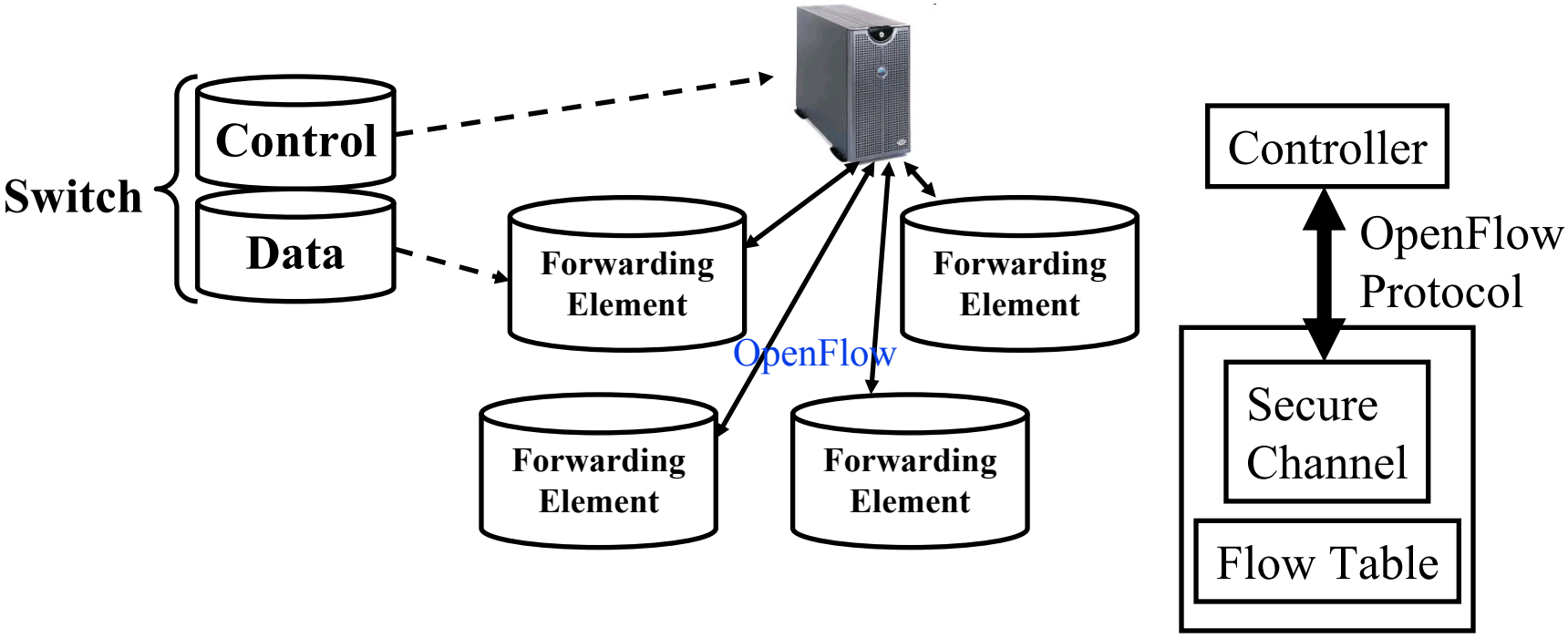
Ref: N. McKeown, et al., ``OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM CCR, Vol. 38, No. 2, April 2008, pp. 69-74.

History of OpenFlow

- 2006: Martin Casado, a PhD student at Stanford and team propose a clean-slate security architecture (SANE) which defines a centralized control of security (in stead of at the edge as normally done). Ethane generalizes it to all access policies.
- April 2008: OpenFlow paper in ACM SIGCOMM CCR
- 2009: Stanford publishes OpenFlow V1.0.0 specs
- June 2009: Martin Casado co-founds Nicira
- March 2010: Guido Appenzeller, head of clean slate lab at Stanford, co-founds Big Switch Networks
- March 2011: Open Networking Foundation is formed
- Oct 2011: First Open Networking Summit. Juniper, Cisco announce plans to incorporate.
- July 2012: VMware buys Nicira for $1.26B
- Nov 6, 2013: Cisco buys Insieme for $838M

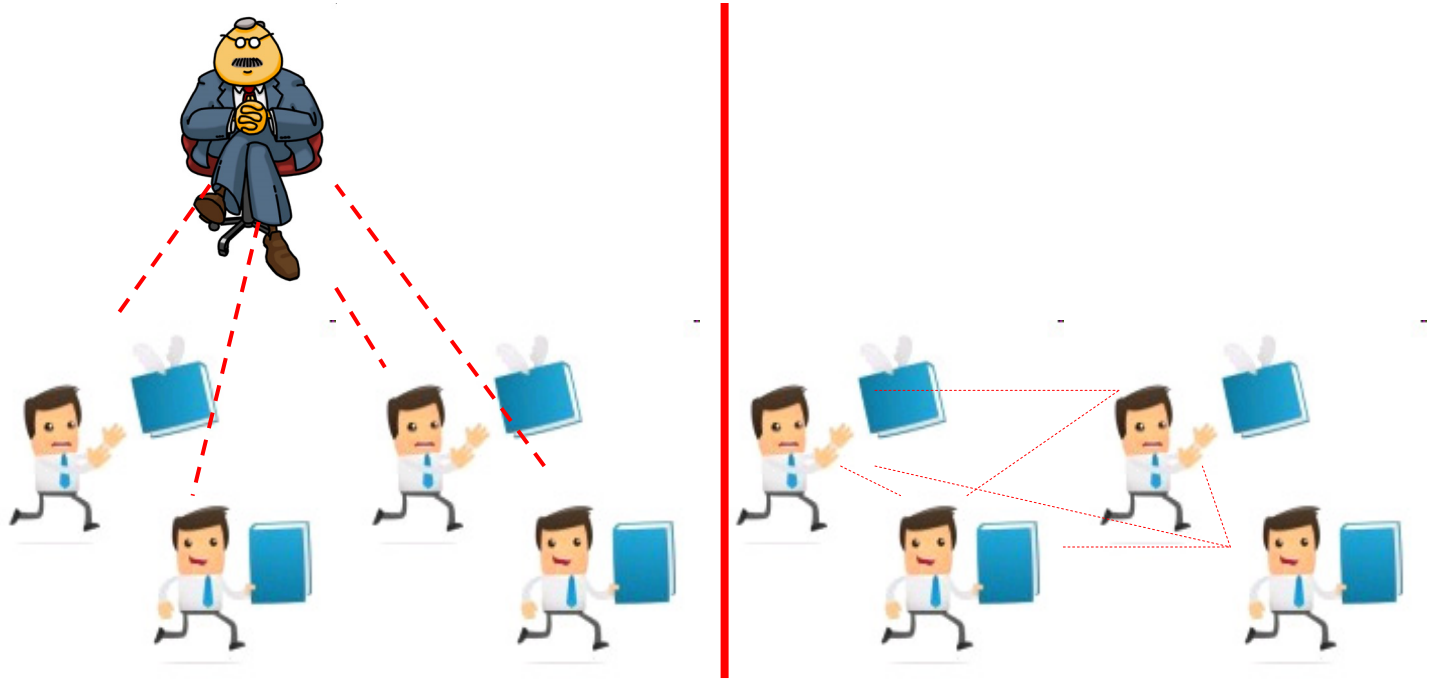
Ref: ONF, "The OpenFlow Timeline," http://openflownetworks.com/of_timeline.php
http://www.cse.wustl.edu/~jain/tutorials/secon14.htm

Separation of Control and Data Plane



- Control logic is moved to a controller
- Switches only have forwarding elements
- One expensive controller with a lot of cheap switches
- OpenFlow is the protocol to send/receive forwarding rules from controller to switches

Centralization of Control Plane



Centralized vs. **Distributed**

- Consistency
- Fast Response to changes
- Easy management of lots of devices

OpenFlow V1.0

- On packet arrival, match the header fields with flow entries in a table, if any entry matches, update the counters indicated in that entry and perform indicated actions

Flow Table:

Header Fields	Counters	Actions
Header Fields	Counters	Actions
…	…	…
Header Fields	Counters	Actions

Ingress Port	Ether Source	Ether Dest	VLAN ID	VLAN Priority	IP Src	IP Dst	IP Proto	IP ToS	Src L4 Port	Dst L4 Port

Ref: http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf

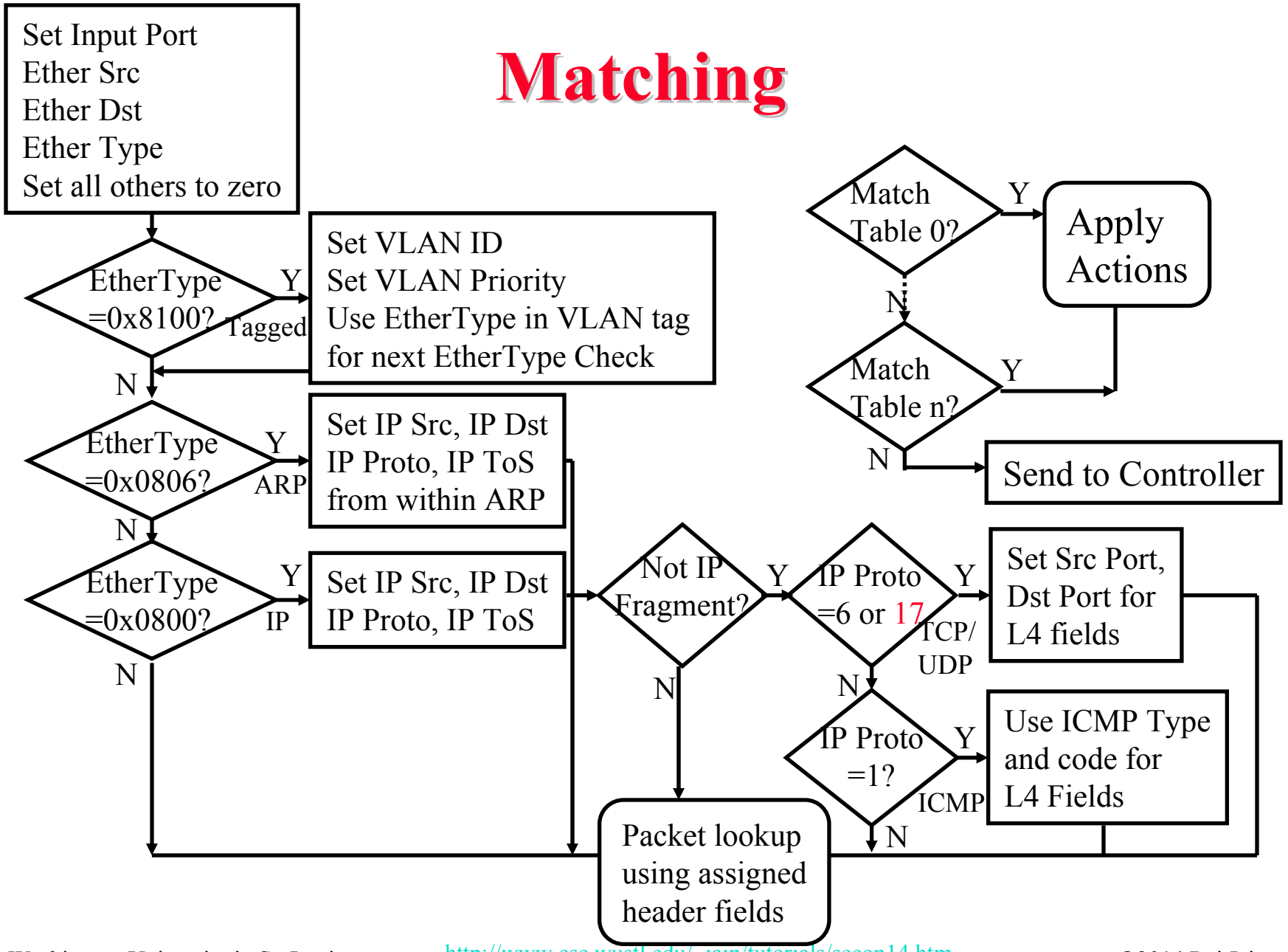
Flow Table Example

Port	Src MAC	Dst MAC	VLAN ID	Priority	EtherType	Src IP	Dst IP	IP Proto	IP ToS	Src L4 Port ICMP Type	Dst L4 Port ICMP Code	Action	Counter
*	*	0A:C8:*	*	*	*	*	*	*	*	*	*	Port 1	102
*	*	*	*	*	*	*	192.168.*.*	*	*	*	*	Port 2	202
*	*	*	*	*	*	*	*	*	*	21	21	Drop	420
*	*	*	*	*	*	*	*	0x806	*	*	*	Local	444
*	*	*	*	*	*	*	*	0x1*	*	*	*	Controller	1

- Idle timeout: Remove entry if no packets received for this time
- Hard timeout: Remove entry after this time
- If both are set, the entry is removed if either one expires.

Ref: S. Azodolmolky, "Software Defined Networking with OpenFlow," Packt Publishing, October 2013, 152 pp., ISBN:978-1-84969-872-6 (Safari Book)

Matching



Counters

Per Table	Per Flow	Per Port	Per Queue
Active Entries	Received Packets	Received Packets	Transmit Packets
Packet Lookups	Received Bytes	Transmitted Packets	Transmit Bytes
Packet Matches	Duration (Secs)	Received Bytes	Transmit overrun errors
	Duration (nanosecs)	Transmitted Bytes	
		Receive Drops	
		Transmit Drops	
		Receive Errors	
		Transmit Errors	
		Receive Frame Alignment Errors	
		Receive Overrun erorrs	
		Receive CRC Errors	
		Collisions	

Actions

- Forward to Physical Port *i* or to *Virtual Port*:
  - **All**: to all interfaces except incoming interface
  - **Controller**: encapsulate and send to controller
  - **Local**: send to its local networking stack
  - **Table**: Perform actions in the flow table
  - **In_port**: Send back to input port
  - **Normal**: Forward using traditional Ethernet
  - **Flood**: Send along minimum spanning tree except the incoming interface
- Enqueue: To a particular queue in the port ⇒ QoS
- Drop
- Modify Field: E.g., add/remove VLAN tags, ToS bits, Change TTL

Actions (Cont)

- Masking allows matching only selected fields, e.g., Dest. IP, Dest. MAC, etc.
- If header matches an entry, corresponding actions are performed and counters are updated
- If no header match, the packet is queued and the header is sent to the controller, which sends a new rule. Subsequent packets of the flow are handled by this rule.
- Secure Channel: Between controller and the switch using TLS
- Modern switches already implement flow tables, typically using Ternary Content Addressable Memories (TCAMs)
- Controller can change the forwarding rules if a client moves
  $\Rightarrow$ Packets for mobile clients are forwarded correctly
- Controller can send flow table entries beforehand (**Proactive**) or Send on demand (**Reactive**). OpenFlow allows both models.

Hardware OpenFlow Switches

- Arista 7050
- Brocade MLXe, Brocade CER, Brocade CES
- Extreme Summit x440, x460, x670
- Huawei openflow-capable router platforms
- HP 3500, 3500yl, 5400zl, 6200yl, 6600, and 8200zl (the old-style L3 hardware match platform)
- HP V2 line cards in the 5400zl and 8200zl (the newer L2 hardware match platform)
- IBM 8264
- Juniper (MX, EX)
- NEC IP8800, NEC PF5240, NEC PF5820
- NetGear 7328SO, NetGear 7352SO
- Pronto (3290, 3295, 3780) - runs the shipping pica8 software
- Switch Light platform

Software OpenFlow Switches

- **Indigo**: Open source implementation that runs on physical switches and uses features of the ASICs to run OpenFlow
- **LINC**: Open source implementation that runs on Linux, Solaris, Windows, MacOS, and FreeBSD
- **Pantou**: Turns a commercial wireless router/access point to an OpenFlow enabled switch. OpenFlow runs on OpenWRT. Supports generic Broadcom and some models of LinkSys and TP-Link access points with Broadcom and Atheros chipsets.
- **Of13softswitch**: User-space software switch based on Ericsson TrafficLab 1.1 softswitch
- **XORPlus**: Open source switching software to drive high-performance ASICs. Supports STP/RSTP/MSTP, LCAP, QoS, VLAN, LLDP, ACL, OSPF/ECMP, RIP, IGMP, IPv6, PIM-SM
- **Open vSwitch**

Ref: http://www.openvswitch.org/, http://www.projectfloodlight.org/indigo/, http://flowforwarding.github.io/LINC-Switch/, http://github.com/CPqD/openflow-openwrt, http://cpqd.github.io/ofsoftswitch13/, http://sourceforge.net/projects/xorplus

Open vSwitch

- Open Source Virtual Switch
- Nicira Concept
- Can Run as a stand alone hypervisor switch or as a distributed switch across multiple physical servers
- Default switch in XenServer 6.0, Xen Cloud Platform and supports Proxmox VE, VirtualBox, Xen KVM
- Integrated into many cloud management systems including OpenStack, openQRM, OpenNebula, and oVirt
- Distributed with Ubuntu, Debian, Fedora Linux. Also FreeBSD
- Intel has an accelerated version of Open vSwitch in its own Data Plane Development Kit (DPDK)

Ref: http://openvswitch.org/

Open vSwitch Features

- Inter-VM communication monitoring via:
  - **NetFlow**: Cisco protocol for sampling and collecting traffic statistics (RFC 3954)
  - **sFlow**: Similar to NetFlow by sflow.org (RFC 3176)
  - **Jflow**: Juniper's version of NetFlow
  - **NetStream**: Huawei's version of NetFlow
  - **IPFIX**: IP Flow Information Export Protocol (RFC 7011) - IETF standard for NetFlow
  - SPAN, RSPAN: Remote Switch Port Analyzer – port mirroring by sending a copy of all packets to a monitor port
  - **GRE-tunneled mirrors**: Monitoring device is remotely connected to the switch via a GRE tunnel

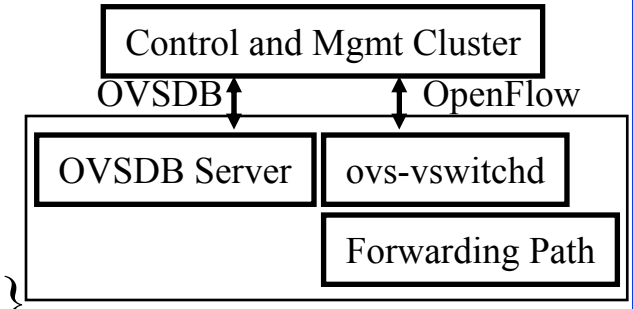
Open vSwitch Features (Cont)

- Link Aggregation Control Protocol (LACP)
- IEEE 802.1Q VLAN
- IEEE 802.1ag Connectivity Fault Management (CFM)
- Bidirectional Forwarding Detection (BFD) to detect link faults (RFC 5880)
- IEEE 802.1D-1998 Spanning Tree Protocol (STP)
- Per-VM traffic policing
- OpenFlow
- Multi-table forwarding pipeline
- IPv6
- GRE, VXLAN, IPSec tunneling
- Kernel and user-space forwarding engine options

OVSDB

- Open vSwitch Database Management Protocol (OVSDB)
- Monitoring capability using publish-subscribe mechanisms
- Stores both provisioning and operational state
- Java Script Object Notation (JSON) used for schema format and for JSON-RPC over TCP for wire protocol (RFC 4627)

<database-schema>
"name": <id>
"version": <version>
"tables": {<id>: <table-schema>,…}

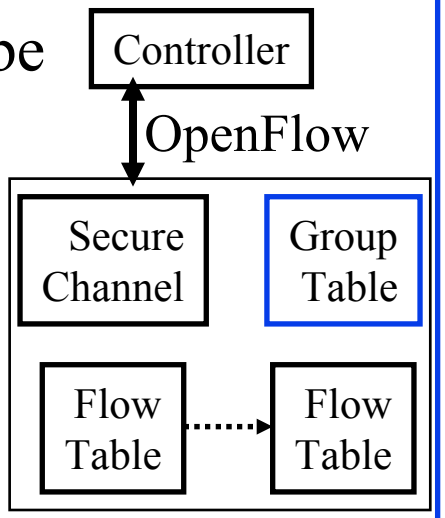


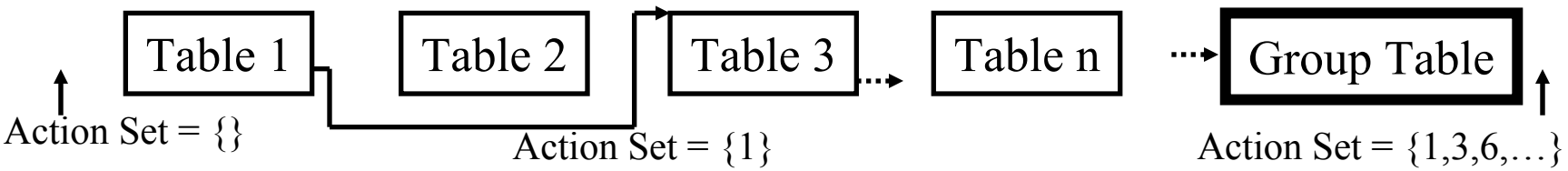
- RPC Methods: List databases, Get Schema, Update, Lock, …
- Open vSwitch project includes open source OVSDB client and server implementations

Ref: B. Pfaff and B. Davie, "The Open vSwitch Database Management Protocol," IETF draft, Oct 2013, http://tools.ietf.org/html/draft-pfaff-ovsdb-proto-04

OpenFlow V1.1

- V1: Perform action on a match. Ethernet/IP only. Single Path Did not cover MPLS, Q-in-Q, ECMP, and efficient Multicast
- V1.1 Introduced *Table chaining, Group Tables, and added MPLS Label and MPLS traffic class to match fields.*
- **Table Chaining**: On a match, instruction may be
  - Immediate actions: modify packet, update match fields and/or
  - Update action set, and/or
  - Send match data and action set to *Table n*,
  - Go to *Group Table* entry *n*





OpenFlow V1.1 (Cont)

- On a miss, the instruction may be to send packet to controller or continue processing with the sequentially next table
- Group Tables: each entry has a variable number of buckets
  - **All**: Execute each bucket. Used for Broadcast, Multicast.
  - **Select**: Execute one *switch selected* bucket. Used for port mirroring. Selection may be done by hashing some fields.
  - **Indirect**: Execute one *predefined* bucket.
  - **Fast Failover**: Execute the first live bucket ⇒ Live port
- New Features supported:
  - **Multipath**: A flow can be sent over one of several paths
  - **MPLS**: multiple labels, traffic class, TTL, push/pop labels
  - **Q-in-Q**: Multiple VLAN tags, push/pop VLAN headers
  - **Tunnels**: via virtual ports

Ref: http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf

OpenFlow V1.2

1. **IPv6 Support**: Matching fields include IPv6 source address, destination address, protocol number, traffic class. ICMPv6 type, ICMPv6 code, IPv6 neighbor discovery header fields, and IPv6 flow labels.
2. **Extensible Matches**: Type-Length-Value (TLV) structure. Previously the order and length of match fields was fixed.
3. **Experimenter extensions** through dedicated fields and code points assigned by ONF

Ref: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.2.pdf

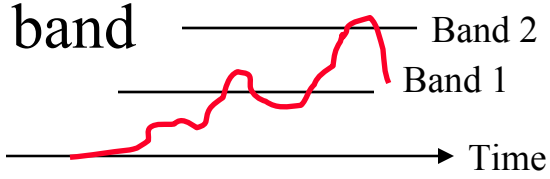
OpenFlow 1.3

- **IPv6 extension headers**: Can check if Hop-by-hop, Router, Fragmentation, Destination options, Authentication, Encrypted Security Payload (ESP), unknown extension headers are present
- **MPLS Bottom-of-Stack bit** matching
- **MAC-in-MAC** encapsulation
- **Tunnel ID meta data**: Support for tunnels (VxLAN, …)
- **Per-Connection Event Filtering**: Better filtering of connections to multiple controllers
- Many **auxiliary connections** to the controller allow to exploit parallelism
- Better **capability negotiation**: Requests can span multiple messages
- More general **experimenter capabilities** allowed
- A separate flow entry for **table miss actions**

Ref: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf

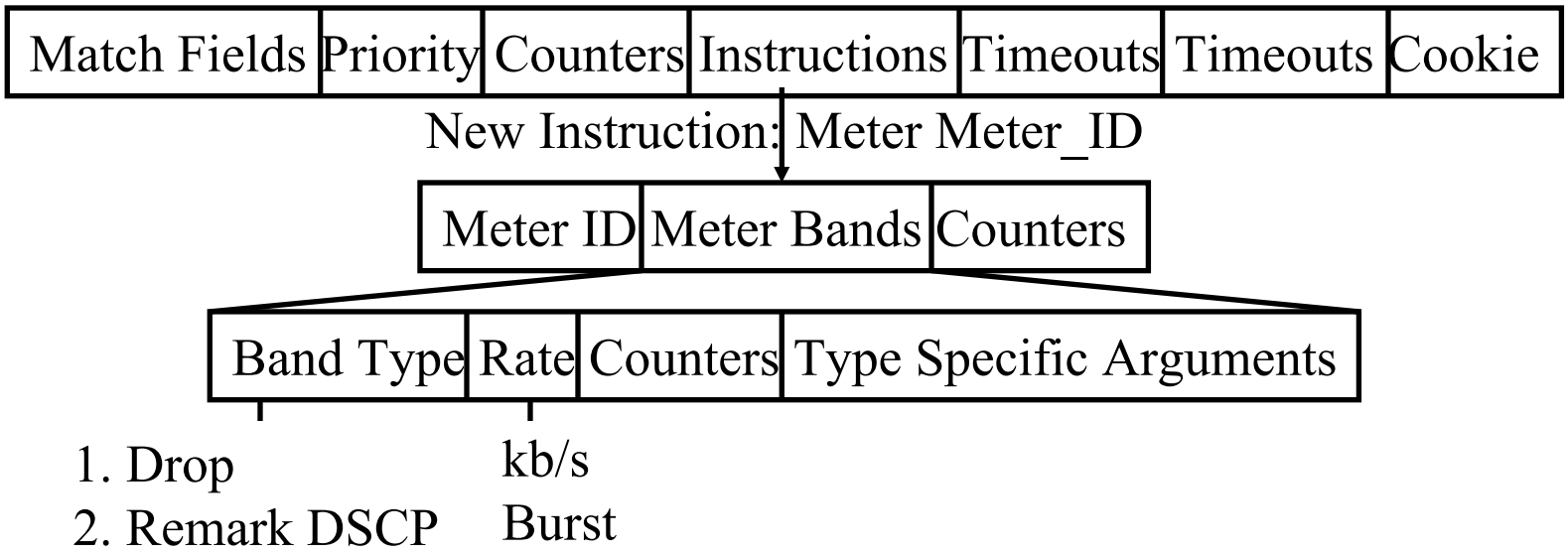
OpenFlow V1.3 (Cont)

- **Cookies**: A cookie field is added to messages containing new packets sent to the controller. This helps controller process the messages faster than if it had to search its entire database.
- **Duration**: Duration field has been added to most stats. Helps compute rates.
- Per-flow counters can be disabled to improve performance
- Per Flow Meters and meter bands
- **Meter**: Switch element that can measure and control the rate of packets/bytes.
  - **Meter Band**: If the packet/byte rate exceeds a pre-defined threshold ⇒ the meter has triggered the band
  - A meter may have multiple bands



OpenFlow V1.3 (Cont)

- If on triggering a band the meter drops the packet, it is called rate limiter.
- Other QoS and policing mechanisms can be designed using these meters
- Meters are attached to a flow entry not to a queue or a port.
- Multiple flow entries can all point to the same meter.

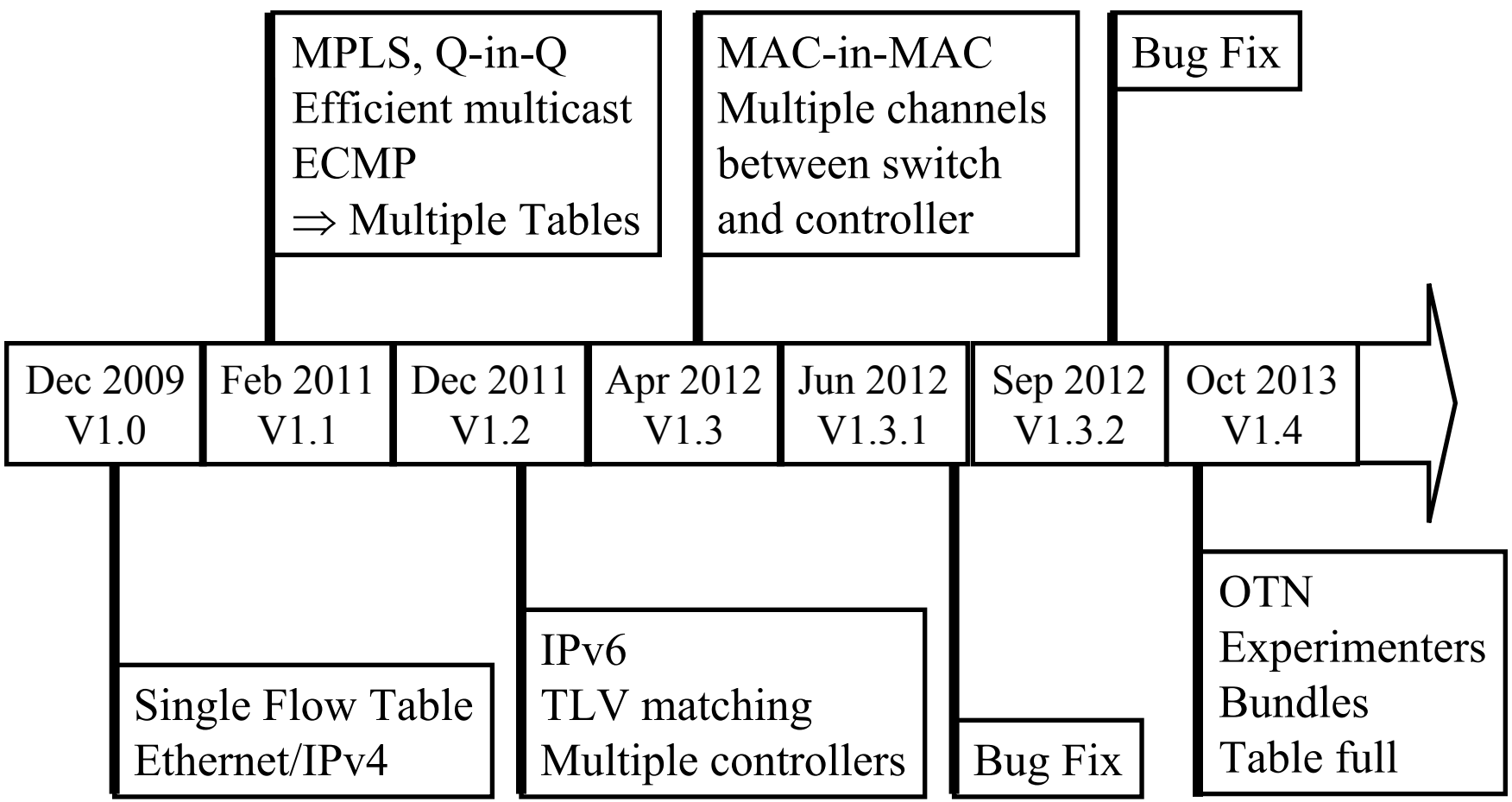


OpenFlow V1.4

- **Optical ports**: Configure and monitor transmit and receive frequencies of lasers and their power
- **Improved Extensibility**: Type-Length-Value (TLV) encodings at most places ⇒ Easy to add new features in future
- **Extended Experimenter Extension API**: Can easily add ports, tables, queues, instructions, actions, etc.
- More information when a packet is sent to controller, e.g., no match, invalid TTL, matching group bucket, matching action, ..
- Controllers can select a subset of flow tables for monitoring
- Switches can **evict** entries of lower importance if table full
- Switches can notify controller if table is getting full
- Atomic execution of a **bundle** of instructions

Ref: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

OpenFlow Evolution Summary



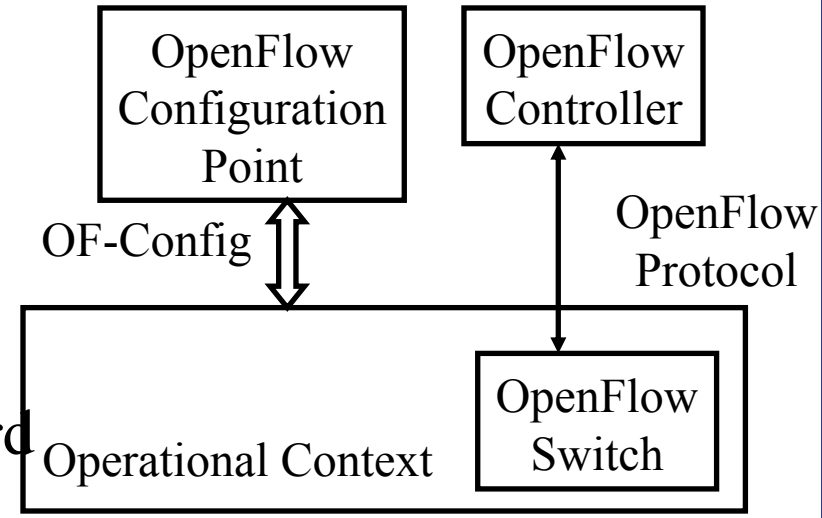
Bootstrapping

- Switches require initial configuration: Switch IP address, Controller IP address, Default gateway
- Switches connect to the controller
- Switch provides configuration information about ports
- Controller installs a rule to forward LLDP packets to controller and then sends, one by one, LLDP packets to be sent out to port i (i=1, 2, …, n) which are forwarded to respective neighbors. The neighbors send the packets back to controller.
- Controller determines the topology from LLDP packets
- LLDP is a one-way protocol to advertise the capabilities at fixed intervals.

Ref: S. Sharma, et al., "Automatic Bootstrapping of OpenFlow Networks," 19th IEEE Workshop on LANMAN, 2013, pp. 1-6, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6528283 (Available to subscribers only)

OpenFlow Configuration Protocol (OF-Config)

- **OpenFlow Control Point**: Entity that configures OpenFlow switches
- **OF-Config**: Protocol used for configuration and management of OpenFlow Switches.
  Assignment of OF controllers so that switches can initiate connections to them:
  - IP address of controller
  - Port number at the controller
  - Transport protocol: TLS or TCP
  - Configuration of queues (min/max rates) and ports
  - Enable/disable receive/forward speed, media on ports



Ref: Cisco, "An Introduction to OpenFlow," Feb 2013,
http://www.cisco.com/web/solutions/trends/open_network_environment/docs/cisco_one_webcastan_introduction_to_openflowfebruary142013.pdf

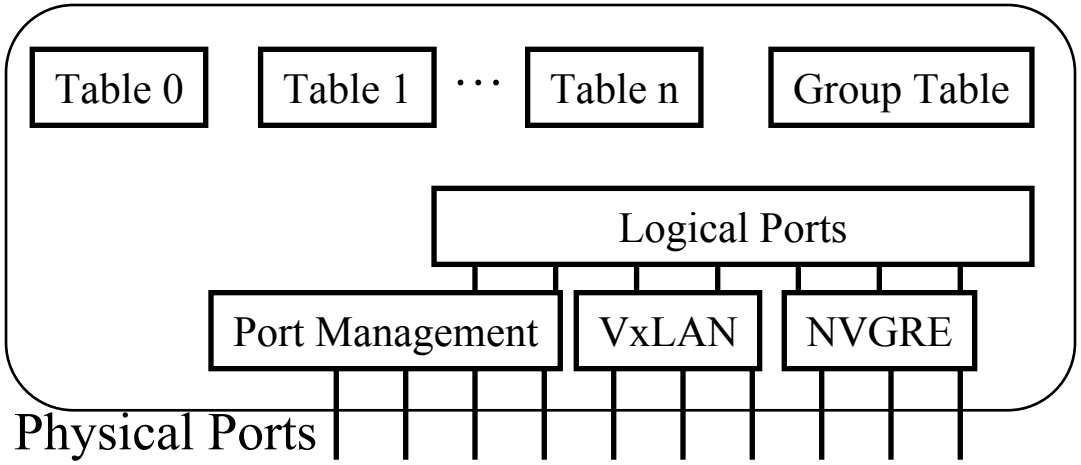
OpenFlow Notification Framework

- **Notification**: Event triggered messages, e.g., link down
- **Publish/subscribe model**: Switch = publisher. OpenFlow controller or OpenFlow config points, and others can subscribe. They will be notified about the events they subscribe.
- Use **ITU-T M.3702** Notifications: Attribute value change, Communication alarm, Environmental alarm, Equipment alarm, QoS alarm, Processing error alarm, Security alarm, State change, Object creation and deletion
- **Pre-existing Notifications**: Do not fit in the framework but will be recognized.
  - OpenFlow: Packet-in, Flow removed, Port Status, Error, Hello, Echo request, Echo reply, Experimenter
  - OpenFlow Config: OpenFlow logical switch instantiation, OpenFlow capability switch capability change, Successful OpenFLow session establishment, Failed OpenFlow session establishment, Port failure or recovery

Ref: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-notifications-framework-1.0.pdf

Implementation Issues

- 40+ matching fields in a flow
- Multiple tables, each with a large number of flow entries
- Instructions and actions for each table
- Need VXLAN, NVGRE, etc. support
- For a large network, flow level programming can take a long time



Ref: R. Oshana and S. Addepalli, "Networking Trends- Software Defined Networking, Network Virtualization and Cloud Orchestration," Asia Power Arch. Conf, Oct 2012, https://www.power.org/wp-content/uploads/2012/10/13.-FSL-SDN-Openflow-and-Cloud-computing-UPD_Rob-Oshana.pdf

OpenFlow: Future Work Items

- Each controller has its own way to program.
  Need a common standard "Northbound API" (ONF NBI group)
- No standard API for communication between controllers of overlapping domain ⇒ Need an East-West API
- Ability to continue operation when the controller is down
- Many other packet formats (non-IP, non-Ethernet, …)
- Flow ⇒ Decide once, use many times ⇒ Performance
  - But does not help non-flow based request/response apps
- Need API to encrypt data plane packets, to inject packets, to instantiate a service, such as a firewall, IDS, on the switch
- Need to program an abstract view, e.g., source to destination, without knowing the physical network

Ref: http://onrc.stanford.edu/research_modern_sdn_stack.html

Ref: T. Nadeau and K. Gray, "SDN," O'Reilly, 2013, 384 pp, ISBN:978-1-449-34230-2

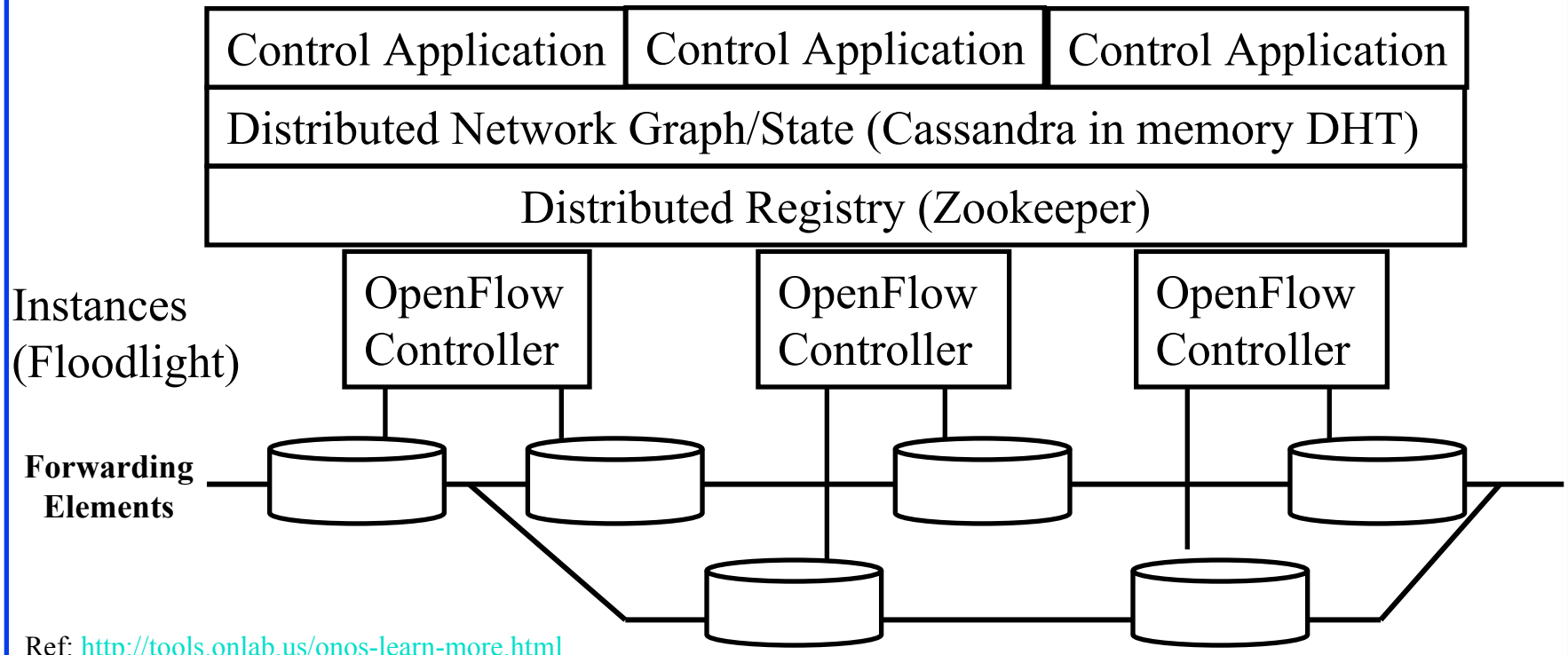
OpenFlow Controllers

1. NOX
2. POX
3. SNAC
4. Beacon
5. Trema
6. Maestro
7. Floodlight
8. ONIX
9. **ONOS**

Many more…This is not a complete list.

ONOS

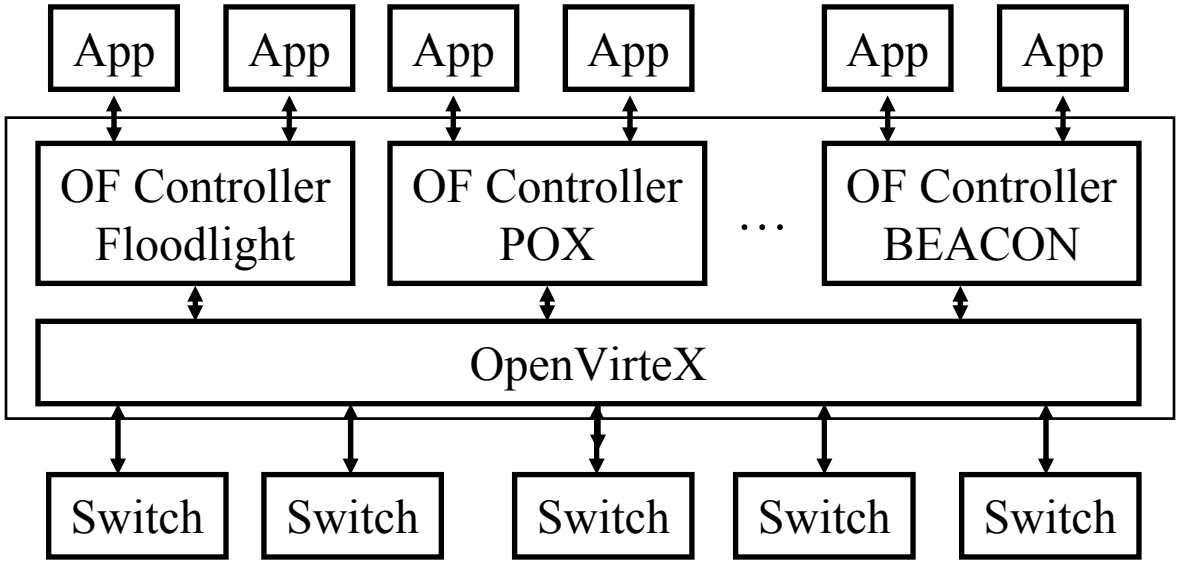
- Open Network Operating System:
  Distributed OpenFlow OS for a large WAN
- 8-10 instances in a cluster.
  Each Instance responsible for a part of a network



Ref: http://tools.onlab.us/onos-learn-more.html

OpenVirteX (OVX)

- Transparent Proxy between OpenFlow switches and multiple OpenFlow Controllers. Slices defined by header fields.
- Creates network slices that can be managed by different controllers ⇒ Isolates slices from each other
- All control traffic goes through OVX ⇒ Slight latency



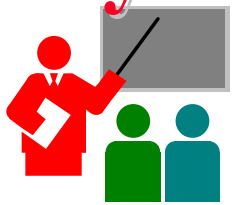
Ref: http://tools.onlab.us/ovx.html

Mininet

- Widely used open source network emulation environment.
- Can simulate a number of end-hosts, switches, routers, links on a Linux
- Used for rapid prototyping of software define networks
- Built-in Open vSwitch, and a OpenFlow capable switch
- Command line launcher and Python API for creating networks of varying sizes, e.g., *mn –topo tree,depth=2,fanout=3*
- Useful diagnositc commands like iperf, ping, and other commands in a host, e.g., *mininet> h11 ifconfig –a*
- Mininet code for several popular commercial switches are available.

Ref: https://github.com/mininet/mininet

Summary of Part VI



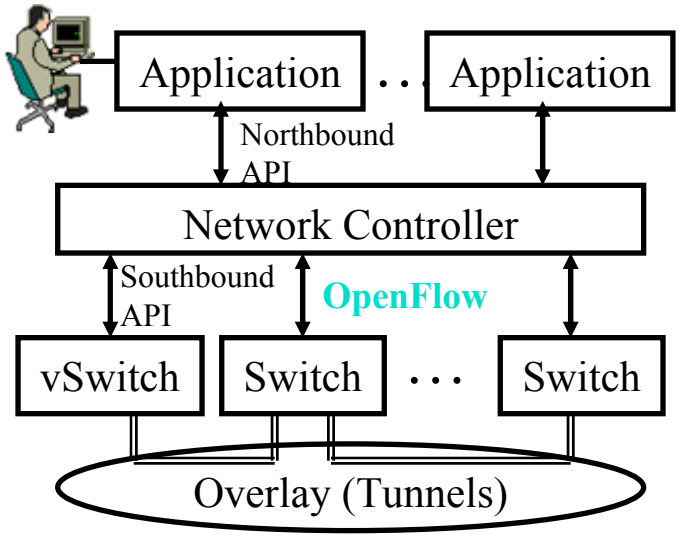
1. Four planes of Networking: Data, Control, Management, Service
2. OpenFlow separates control plane and moves it to a central controller ⇒ Simplifies the forwarding element
3. Switches match incoming packets with flow entries in a table and handle it as instructed. The controller supplies the flow tables and other instructions.
4. OpenFlow has been extended to IPv4, MPLS, IPv6, and Optical Network. But more work ahead.
5. ONOS controller, OVX virtualization, Mininet for emulation

Part VII: Software Defined Networking (SDN)

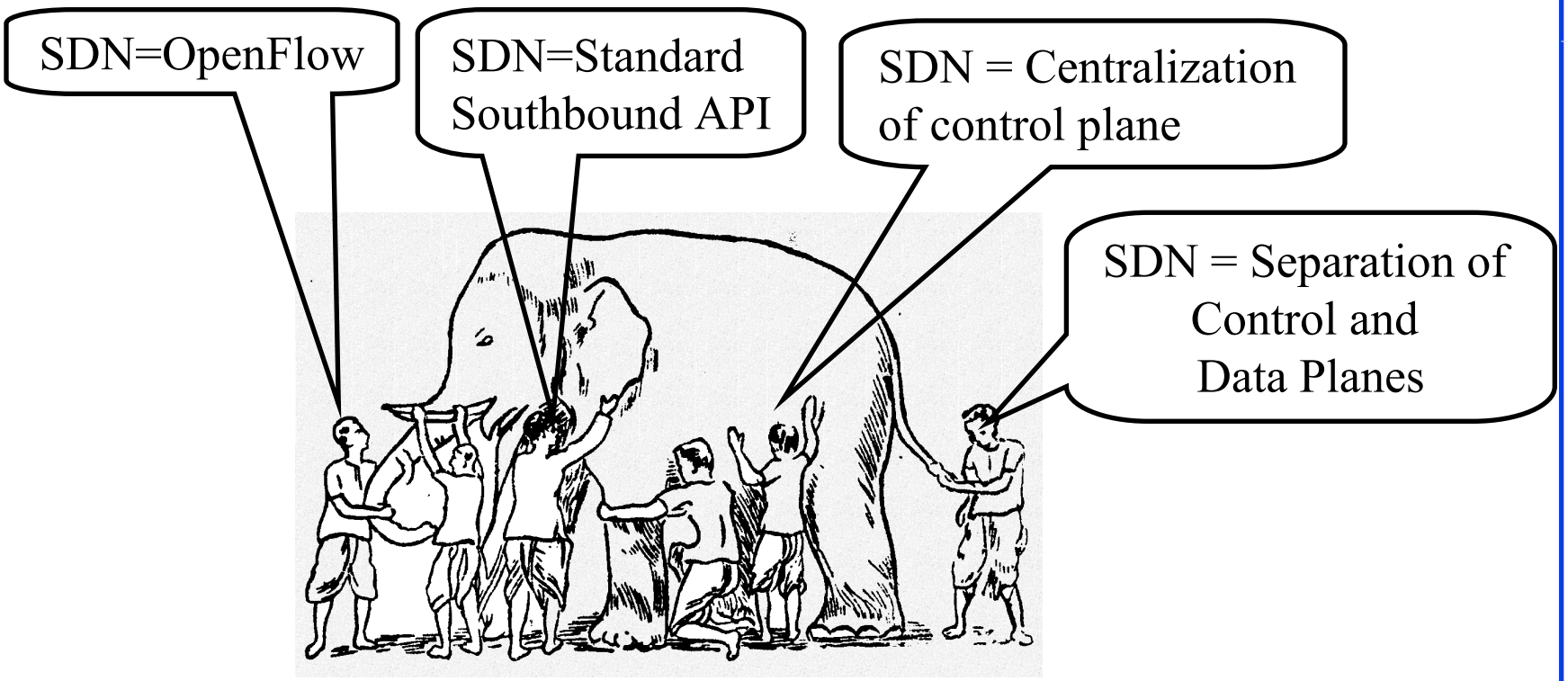
- What is SDN?
- Alternative APIs: XMPP, PCE, ForCES, ALTO
- OpenDaylight SDN Controller Platform and Tools

SDN 1.0: SDN Based on OpenFlow

- SDN originated from OpenFlow
- Centralized Controller
  ⇒ Easy to program
  ⇒ Change routing policies on the fly
  ⇒ Software Defined Network (SDN)
- Initially, SDN = OpenFlow



What is SDN?



- All of these are mechanisms.
- SDN is *not* a mechanism.
- It is a framework to solve a set of problems ⇒ Many solutions

ONF Definition of SDN

"***What is SDN?***

The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices."

1. Directly programmable
2. Agile: *Abstracting control from forwarding*
3. Centrally managed
4. Programmatically configured
5. Open standards-based vendor neutral

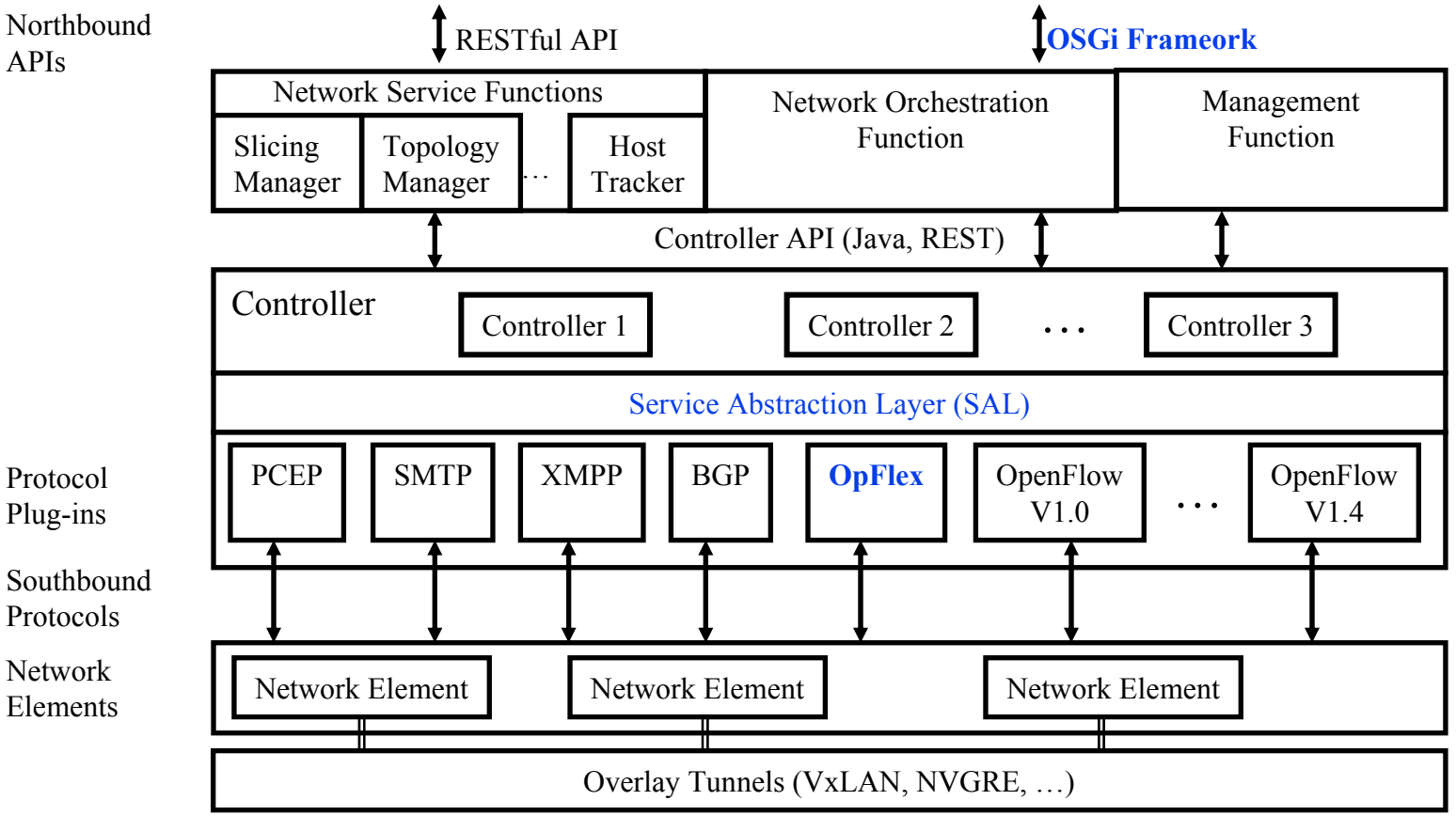
The above definition includes *How*.
Now many different opinions about ***How***.
⇒SDN has become more general.
Need to define by ***What***?

Ref: https://www.opennetworking.org/index.php?option=com_content&view=article&id=686&Itemid=272&lang=en
http://www.cse.wustl.edu/~jain/tutorials/secon14.htm

What do We need SDN for?

1. **Virtualization**: Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.
2. **Orchestration**: Manage thousands of devices
3. **Programmable**: Should be able to change behavior on the fly.
4. **Dynamic Scaling**: Should be able to change size, quantity
5. **Automation**: Lower OpEx
6. **Visibility**: Monitor resources, connectivity
7. **Performance**: Optimize network device utilization
8. **Multi-tenancy**: Sharing expensive infrastructure
9. **Service Integration**
10. **Openness**: Full choice of Modular plug-ins
11. **Unified management** of computing, networking, and storage

SDN 2.0: OpenDaylight Style SDN



- **NO-OpenFlow** (**Not Only** OpenFlow) Multi-Protocol
- New work in **IETF** XMPP, ALTO, I2RS, PCEP, ….
- Linux Foundation

Open Everything

- Open Networking Foundation
- OpenFlow
- OpenStack
- OpenDaylight
- Open Access
- Open Source

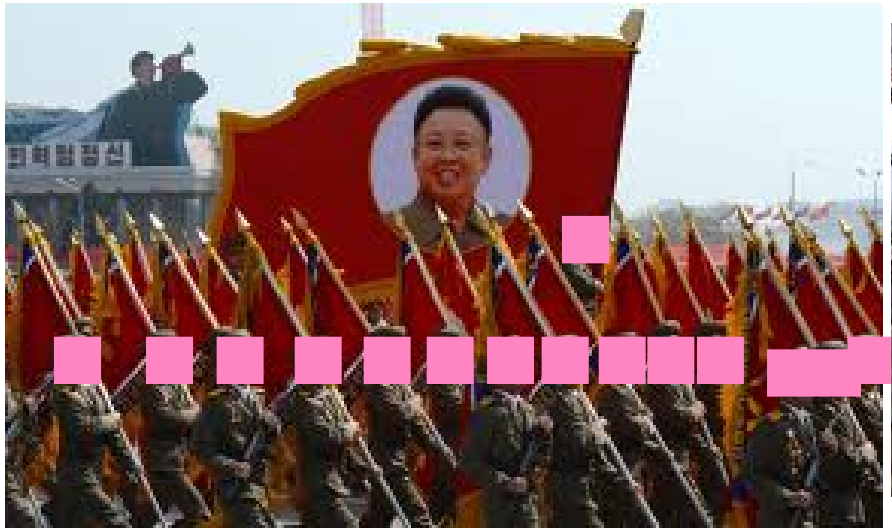
Current SDN Debate: What vs. How?

- SDN is easy if control plane is centralized but not necessary.
  Distributed solutions may be required for legacy equipment and for fail-safe operation.
- Complete removal of control plane may be harmful.
  Exact division of control plane between centralized controller and distributed forwarders is yet to be worked out
- SDN is easy with a standard southbound protocol like OpenFlow but one protocol may not work/scale in all cases
  - Diversity of protocols is a fact of life.
  - There are no standard operating systems, processors, routers, or Ethernet switches.
- If industry finds an easier way to solve the same problems by another method, that method may win. E.g., ATM vs. MPLS.

Separation vs. Centralization

Separation of Control Plane

Centralization of Control Plane



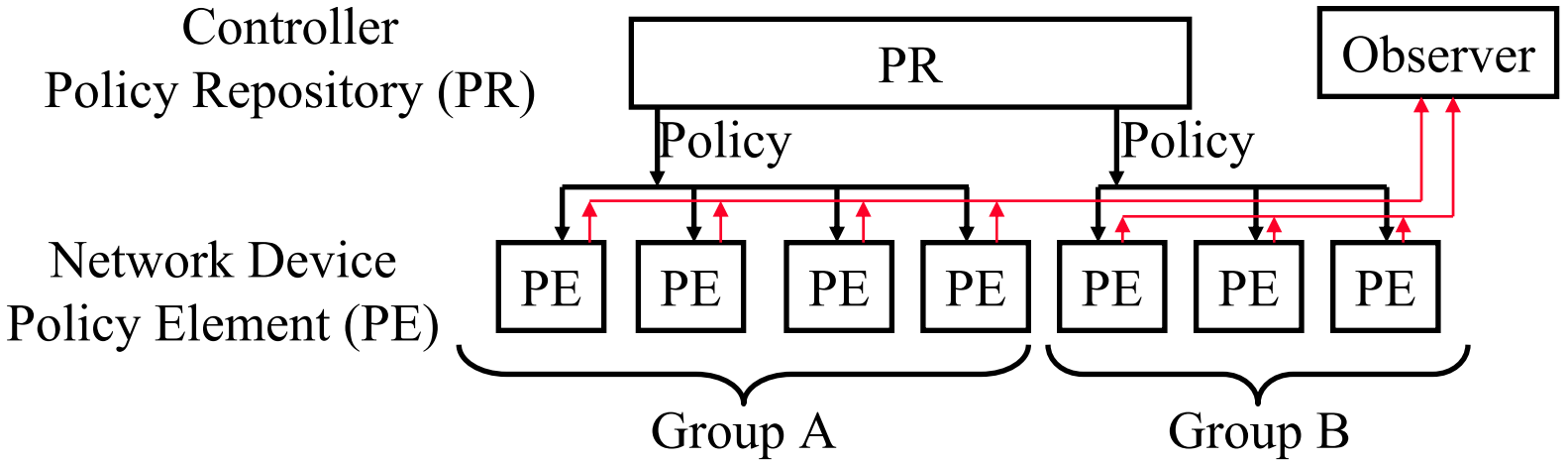
Micromanagement is not scalable

SDN Everywhere

- Software Defined Switches
- Software Defined Routers
- Software Defined Data Center
- Software Defined Storage
- Software Defined Base Stations
- Software Defined GPS
- Software Defined Radio
- Software Defined Infrastructure
- Software Defined Optical Switches

OpFlex

- Open Policy Protocol
- Declarative Control: Controller tells the policy to the network device, which implements it in its own way.

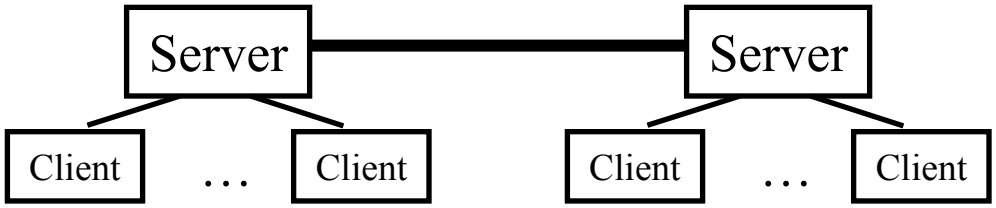


- Policies are communicated using XML, JSON, binary, …
- Observer collects management info, faults, events, …

Ref: http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731302.html

XMPP

- Extensible Messaging and Presence Protocol
- **Extensible** ⇒ Using XML
- Similar to SMTP email protocol but for near real-time communication
- Each client has an ID, e.g., john@wustl.edu/mobile (John's mobile phone)
- Client sets up a connection with the server ⇒ Client is online
- **Presence**: Server maintains contact addresses and may let other contacts know that this client is now on-line
- **Messaging**: When a client sends a "chat" message to another clients, it is forwarded to these other clients
- Messages are "*pushed*" (⇒ real-time) as opposed to "*polled*" as in SMTP/POP emails.



Ref: P. Saint-Andre, et al., "XMPP: The Definitive Guide," O'Reilly, 2009, 320 pp., ISBN:9780596521264 (Safari Book)

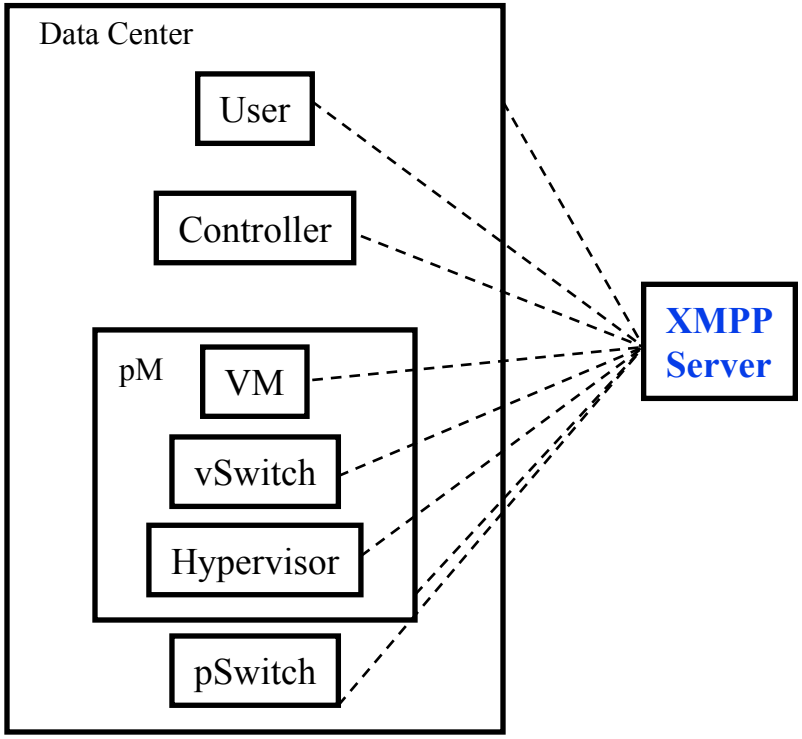
XMPP (Cont)

- XMPP is IETF standardization of Jabber protocol
- RFC 6121 defines XMPP using TCP connections.
  But HTTP is often used as transport to navigate firewalls
- All messages are XML encoded
  ⇒ Not efficient for binary file transfers
  ⇒ Out-of-band binary channels are often used with XMPP.
- A number of open-source implementations are available
- Variations of it are widely used in most instant messaging programs including Google, Skype, Facebook, …, many games
- Used in IoT and data centers for management. Network devices have XMPP clients that respond to XMPP messages containing CLI management requests ⇒ You can manage your network using any other XMPP client, e.g., your mobile phone
- Arista switches can be managed by XMPP, Juniper uses XMPP as a southbound protocol for SDN

Ref: http://en.wikipedia.org/wiki/XMPP

XMPP in Data Centers

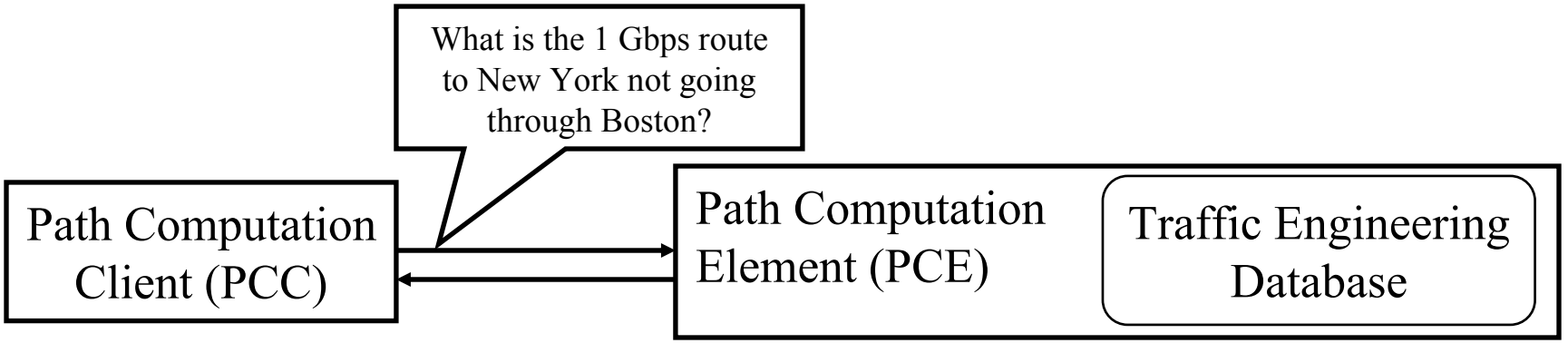
- Everything is an XMPP entity.
  It has its own contact list and authorizations.



Ref: https://github.com/ArchipelProject/Archipel/wiki/Architecture-%26-Concepts
http://www.cse.wustl.edu/~jain/tutorials/secon14.htm

Path Computation Element (PCE)

- MPLS and GMPLS require originating routers to find paths that satisfy multiple constraints including not using any backup routers and having a given bandwidth etc.
- This may require more computer power or network knowledge than a router may have.
- IETF PCE working group has developed a set of protocols that allow a Path computation client (PCC), i.e., router to get the path from path computation element (PCE)
- PCE may be centralized or may be distributed in many or every router.



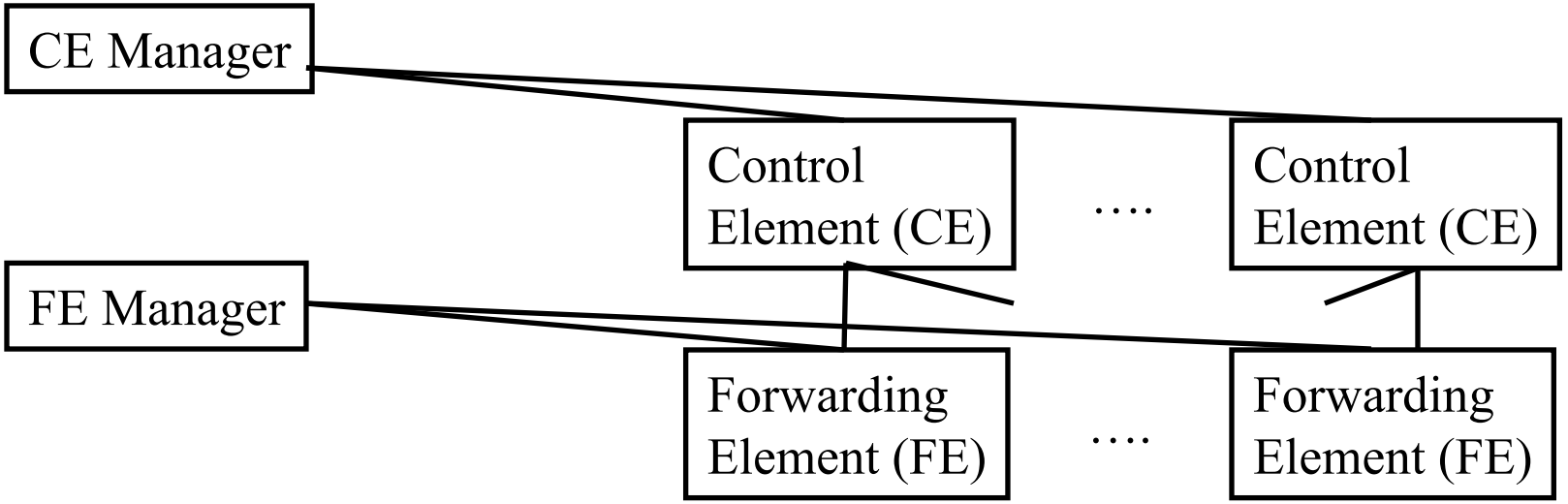
PCE (Cont)

- PCE separates the route computation function from the forwarding function.
- Both functions may be resident in the same box or different boxes.
- 25+ RFCs documenting protocols for:
  - PCE-to-PCC communication
  - PCE-to-PCE communication (Multiple PCEs)
  - PCE discovery

Ref: http://datatracker.ietf.org/wg/pce/
Ref: http://en.wikipedia.org/wiki/Path_computation_element

Forwarding and Control Element Separation (ForCES)

- IETF working group since July 2001
- Control Elements (CEs) prepare the routing table for use by forwarding elements (FEs).
- Each CE may interact with one or more FEs
- There may be many CEs and FEs managed by a CE manager and a FE manager

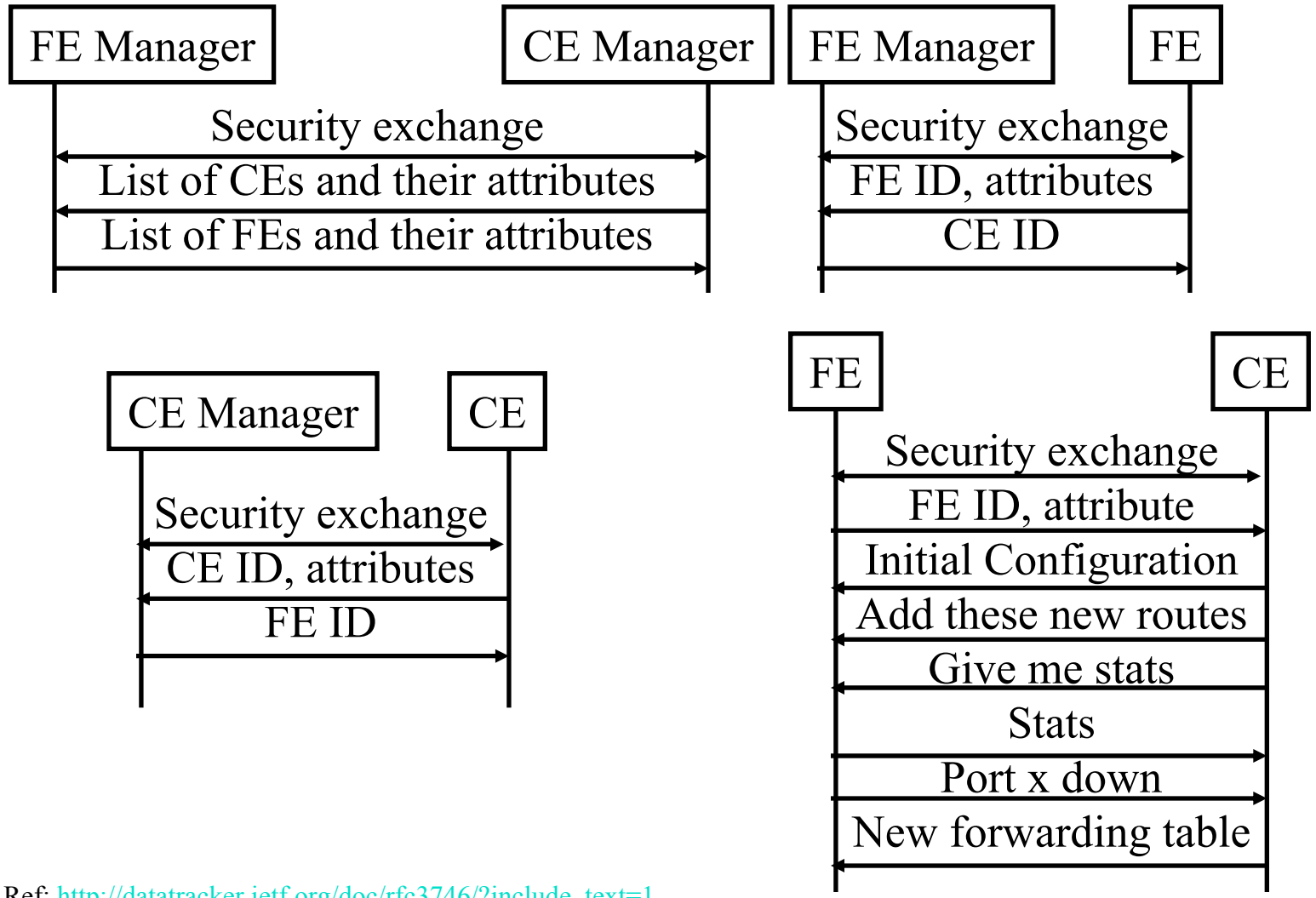


ForCES (Cont)

- Idea of control and data plane separation was used in BSD 4.4 *routing sockets* in early 1990s. It allowed routing tables to be controlled by a simple command line or by a route daemon.
- ForCES protocol supports exchange of:
  - Port type, link speed, IP address
  - IPv4/IPv6 unicast/multicast forwarding
  - QoS including metering, policing, shaping, and queueing
  - Packet classification
  - High-touch functions, e.g., Network Address Translation (NAT), Application-level Gateways (ALG)
  - Encryptions to be applied to packets
  - Measurement and reporting of per-flow traffic information

Ref: http://datatracker.ietf.org/doc/rfc3654/?include_text=1

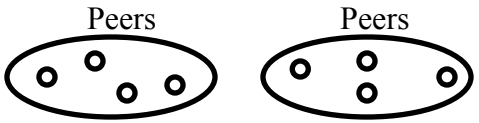
Sample ForCES Exchanges



Ref: http://datatracker.ietf.org/doc/rfc3746/?include_text=1

Application Layer Traffic Optimization (ALTO)

- IETF working group to optimize P2P traffic
  $\Rightarrow$ Better to get files from nearby peers

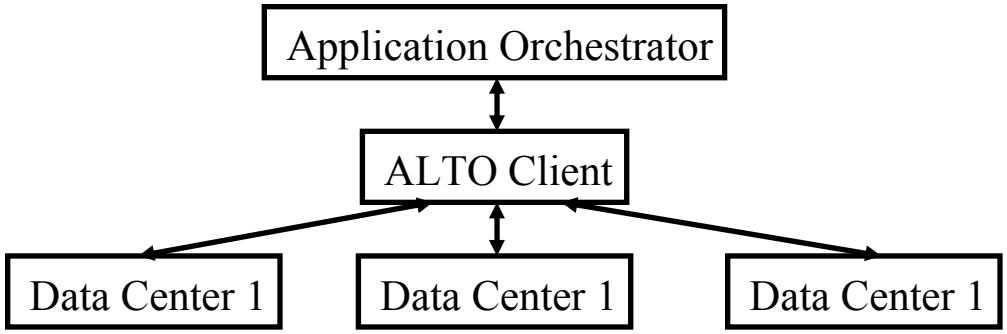


- Provide guidance in peer selection
- ALTO Server: Has knowledge of distributed resources
- ALTO Client: Requests information from servers about the appropriate peers
- Ratio Criteria: Topological distance, traffic charges, …
- ALTO Server could get information from providers or from nodes about their characteristics, e.g., flat-rate or volume based charging
- A client may get the list of potential peers and send it to the server, which can return a ordered list
- Also need a protocol for ALTO server discovery

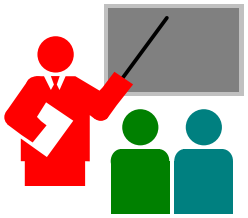
Ref: J. Seedorf and E. Berger, "ALTO Problem Statement," http://datatracker.ietf.org/doc/rfc5693/?include_text=1
Ref: Y. Lee, et al., "ALTO Extensions for collecting Data Center Resource Information," http://datatracker.ietf.org/doc/draft-lee-alto-ext-dc-resource/?include_text=1

ALTO Extension

- Now being extended to locate resources in data centers
- Need to be able to express
  - resource (memory, storage, CPU, network) availability
  - Cost of these resources
  - Constraints on resources, e.g., bandwidth
  - Constraints on structure, e.g., Power consumption
- ALTO client gets the info from various providers
- Issue of privacy of resource and cost info for the provider



Summary of Part VII



1. SDN is the framework to automatically manage and control a large number of network devices and services in a multi-tenant environment
2. OpenFlow originated SDN but now many different southbound and northbound APIs, intermediate services and tools are being discussed and implemented by the industry, e.g., XMPP, ForCES, PCE, ALTO
3. OpenDaylight SDN Controller platform is the leading open source SDN controller project under Linux Foundation
4. Its modular implementation allows many southbound protocols

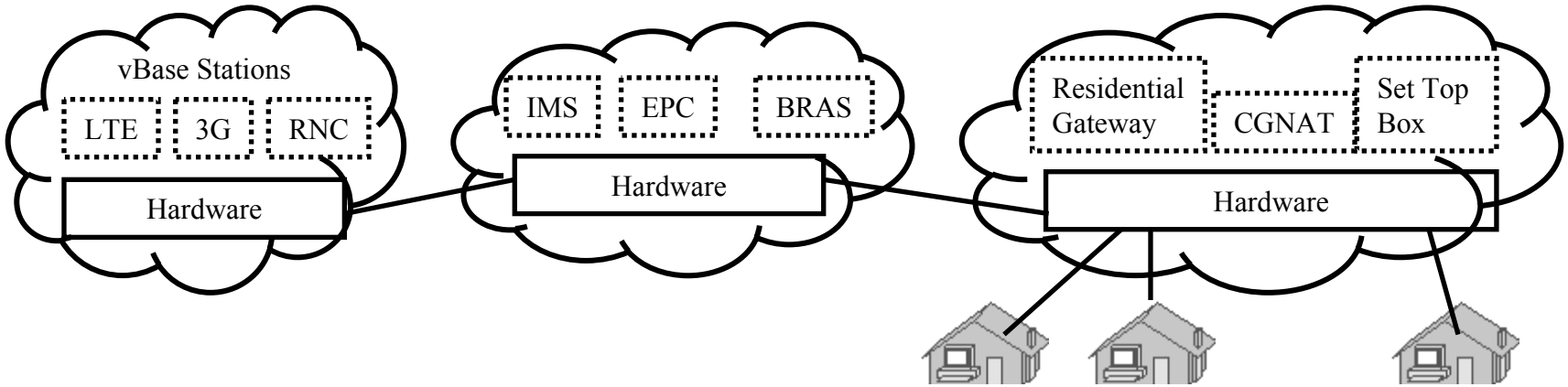
Part VIII: Network Function Virtualization (NFV)

- What is NFV?
- NFV and SDN Relationship
- ETSI NFV ISG Specifications
- Concepts, Architecture, Requirements, Use cases
- Proof-of-Concepts and Timeline

Network Function Virtualization (NFV)

1. Fast standard hardware ⇒ **Software based Devices**
   Routers, Firewalls, Broadband Remote Access Server (BRAS) ⇒ A.k.a. *white box* implementation

2. **Virtual Machine implementation**
   ⇒ Virtual appliances
   ⇒ All advantages of virtualization (quick provisioning, scalability, mobility, Reduced CapEx, Reduced OpEx, …)



Ref: ETSI, "NFV – Update White Paper," Oct 2013, http://www.tid.es/es/Documents/NFV_White_PaperV2.pdf (Must read)

Why We need NFV?

1. **Virtualization**: Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.
2. **Orchestration**: Manage thousands of devices
3. **Programmable**: Should be able to change behavior on the fly.
4. **Dynamic Scaling**: Should be able to change size, quantity
5. **Automation**
6. **Visibility**: Monitor resources, connectivity
7. **Performance**: Optimize network device utilization
8. **Multi-tenancy**
9. **Service Integration**
10. **Openness**: Full choice of Modular plug-ins

Note: These are exactly the same reasons why we need SDN.

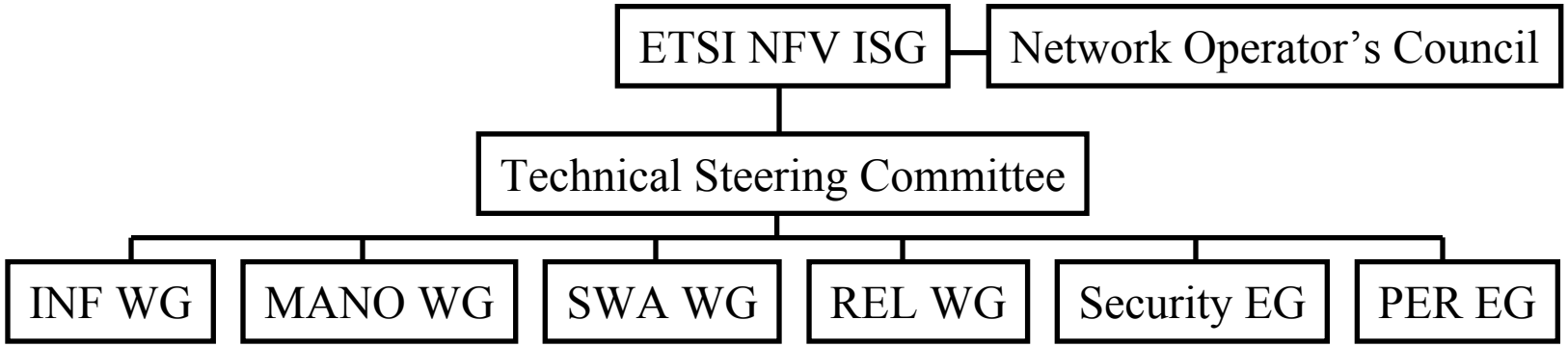
NFV and SDN Relationship

- Concept of NFV originated from SDN
  ⇒ First ETSI white paper showed overlapping Venn diagram
  ⇒ It was removed in the second version of the white paper
- NFV and SDN are complementary.
  One does not depend upon the other.
  You can do SDN only, NFV only, or SDN and NFV.
- Both have similar goals but approaches are very different.
- SDN needs new interfaces, control modules, applications.
  NFV requires moving network applications from dedicated hardware to virtual containers on commercial-off-the-shelf (COTS) hardware
- NFV is present. SDN is the future.
- Virtualization alone provides many of the required features
- Not much debate about NFV.

Mobile Network Functions

- Switches, e.g., Open vSwitch
- Routers, e.g., Click
- Home Location Register (HLR),
- Serving GPRS Support Node (SGSN),
- Gateway GPRS Support Node (GGSN),
- Combined GPRS Support Node (CGSN),
- Radio Network Controller (RNC),
- Serving Gateway (SGW),
- Packet Data Network Gateway (PGW),
- Residential Gateway (RGW),
- Broadband Remote Access Server (BRAS),
- Carrier Grade Network Address Translator (CGNAT),
- Deep Packet Inspection (DPI),
- Provider Edge (PE) Router,
- Mobility Management Entity (MME),
- Element Management System (EMS)

ETSI NFV ISG



- Industry Specification Group (ISG)'s goal is to define the requirements.
- Four Working Groups:
  - **INF**: Architecture for the virtualization Infrastructure
  - **MANO**: Management and orchestration
  - **SWA**: Software architecture
  - **REL**: Reliability and Availability, resilience and fault tolerance

Ref: M. Cohn, "NFV, An Insider's Perspective: Part 1: Goals, History, and Promise," Sep 2013, http://www.sdncentral.com/education/nfv-insiders-perspective-part-1-goals-history-promise/2013/09/

ETSI NFV ISG (Cont)

- Two Expert Groups:
  - **Security** Expert Group: Security
  - **Performance and Portability** Expert Group: Scalability, efficiency, and performance VNFs relative to current dedicated hardware

NFV Specifications

1. NFV Use cases (GS NFV 001)
2. NFV Architectural Framework (GS NFV 002)
3. Terminology for Main Concepts in NFV (GS NFV 003)
4. NFV Virtualization Requirements (GS NFV 004)
5. NFV Proof of Concepts Framework (GS NFV-PER 002)

Ref: ETSI, "Network Function Virtualization," http://www.etsi.org/technologies-clusters/technologies/nfv

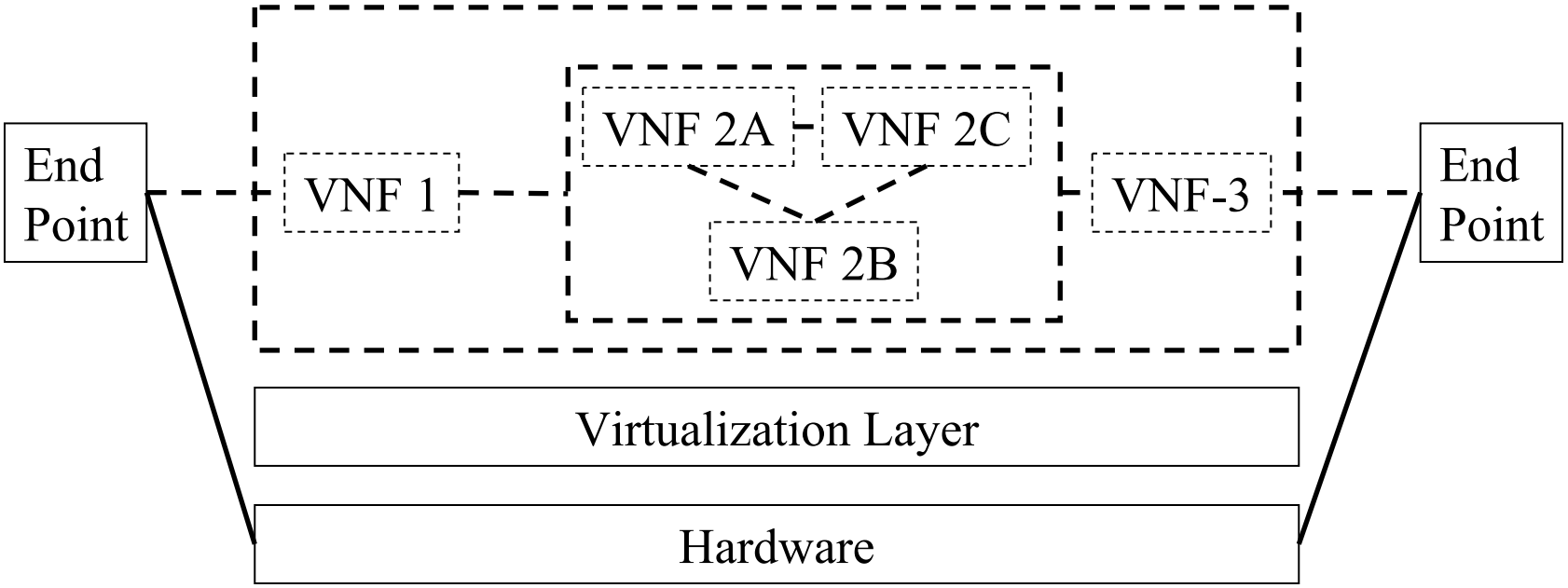
NFV Concepts

- **Network Function (NF):** Functional building block with a well defined interfaces and well defined functional behavior
- **Virtualized Network Function (VNF)**: Software implementation of NF that can be deployed in a virtualized infrastructure
- **VNF Set**: Connectivity between VNFs is not specified, e.g., residential gateways
- **VNF Forwarding Graph**: Service chain when network connectivity order is important, e.g., firewall, NAT, load balancer
- **NFV Infrastructure (NFVI)**: Hardware and software required to deploy, mange and execute VNFs including computation, networking, and storage.

Ref: ETSI, "Architectural Framework," Oct 2013, http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf
Ref: ETSI, "NFV Terminology for Main Concepts in NFV," Oct 2013, http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.01.01_60/gs_NFV003v010101p.pdf
Ref: W. Xu, et al., "Data Models for NFV," IETF Draft, Sep 2013, http://tools.ietf.org/html/draft-xjz-nfv-model-datamodel-00

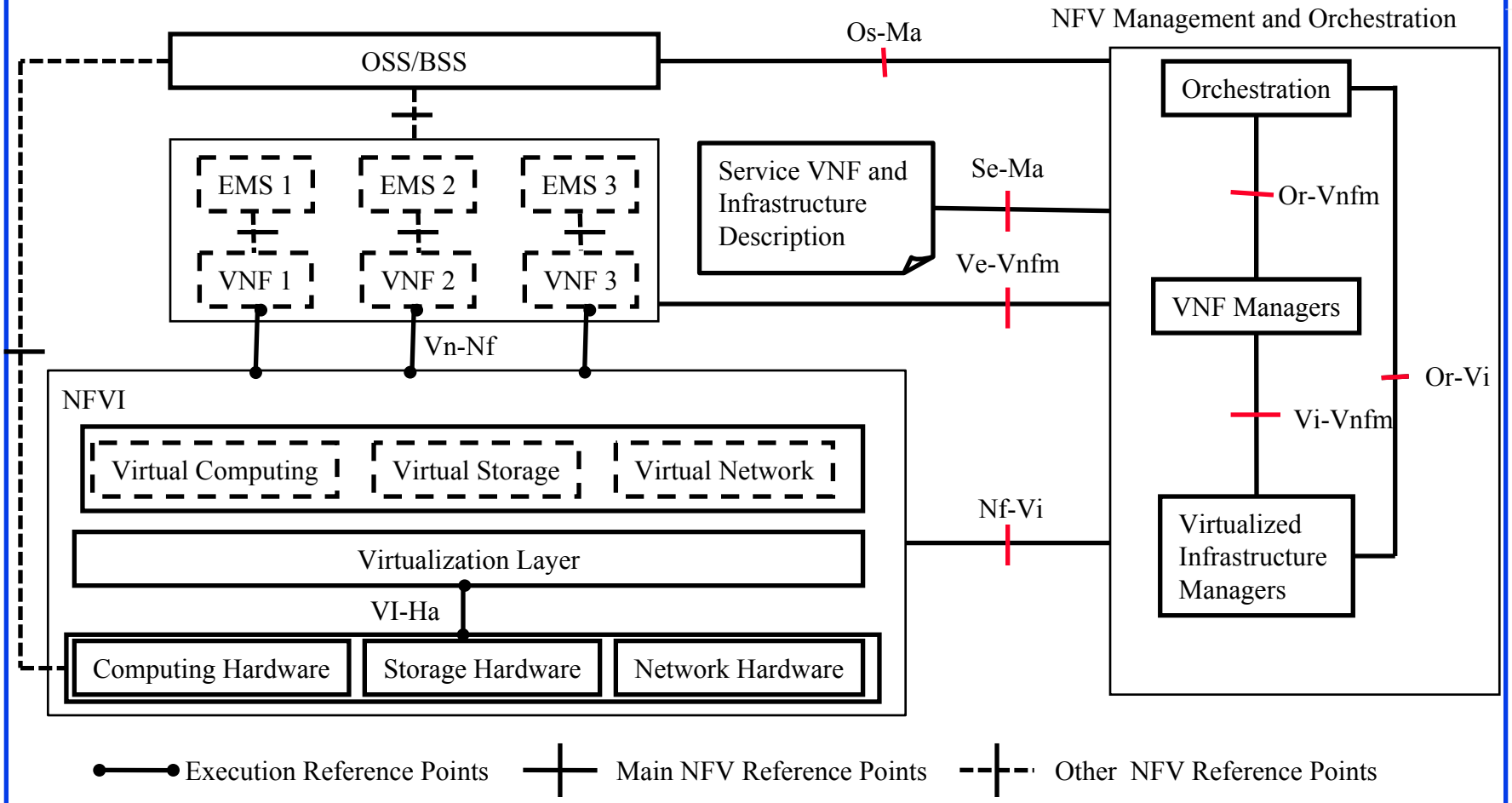
Network Forwarding Graph

- An end-to-end service may include nested forwarding graphs



Ref: ETSI, "Architectural Framework," Oct 2013, http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf

NFV Architecture



Ref: ETSI, "Architectural Framework," Oct 2013, http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf

NFV Reference Points

Reference Point: Points for inter-module specification

1. Virtualization Layer-Hardware Resources (**VI-Ha**)
2. VNF – NFVI (**Vn-Nf**)
3. Orchestrator – VNF Manager (**Or-Vnfm**)
4. Virtualized Infrastructure Manager – VNF Manager (**Vi-Vnfm**)
5. Orchestrator – Virtualized Infrastructure Manager (**Or-Vi**)
6. NFVI-Virtualized Infrastructure Manager (**Nf-Vi**)
7. Operation Support System (OSS)/Business Support Systems (BSS) – NFV Management and Orchestration (**Os-Ma**)
8. VNF/ Element Management System (EMS) – VNF Manager (**Ve-Vnfm**)
9. Service, VNF and Infrastructure Description – NFV Management and Orchestration (**Se-Ma**): VNF Deployment template, VNF Forwarding Graph, service-related information, NFV infrastructure information

Ref: ETSI, "Architectural Framework," Oct 2013, http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf

NFV Use Cases

- **Cloud**:
  1. NFV infrastructure as a service (NFVIaaS) like IaaS
  2. Virtual Network Functions (VNFs) as a service (VNFaaS) like SaaS
  3. VNF forwarding graphs (Service Chains)
  4. Virtual Network Platform as a Service (VNPaaS) like PaaS
- **Mobile**:
  5. Virtualization of the Mobile Core Network and IMS
  6. Virtualization of Mobile Base Station
- **Data Center**:
  7. Virtualization of CDNs
- **Access/Residential**:
  8. Virtualization of the Home environment
  9. Fixed Access NFV

Ref: ETSI, "NFV Use Cases," http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf
Ref: M. Cohn, "NFV Insider's Perspective, Part 2: There's a Network in NFV – The Business Case for SDN," Sep 2013, http://www.sdncentral.com/education/nfv-insiders-perspective-part-2-theres-network-nfv-business-case-sdn/2013/09/

NFV Proof of Concepts (PoCs)

ETSI has formed and NFV ISG PoC Forum.
Following modules have been demoed:

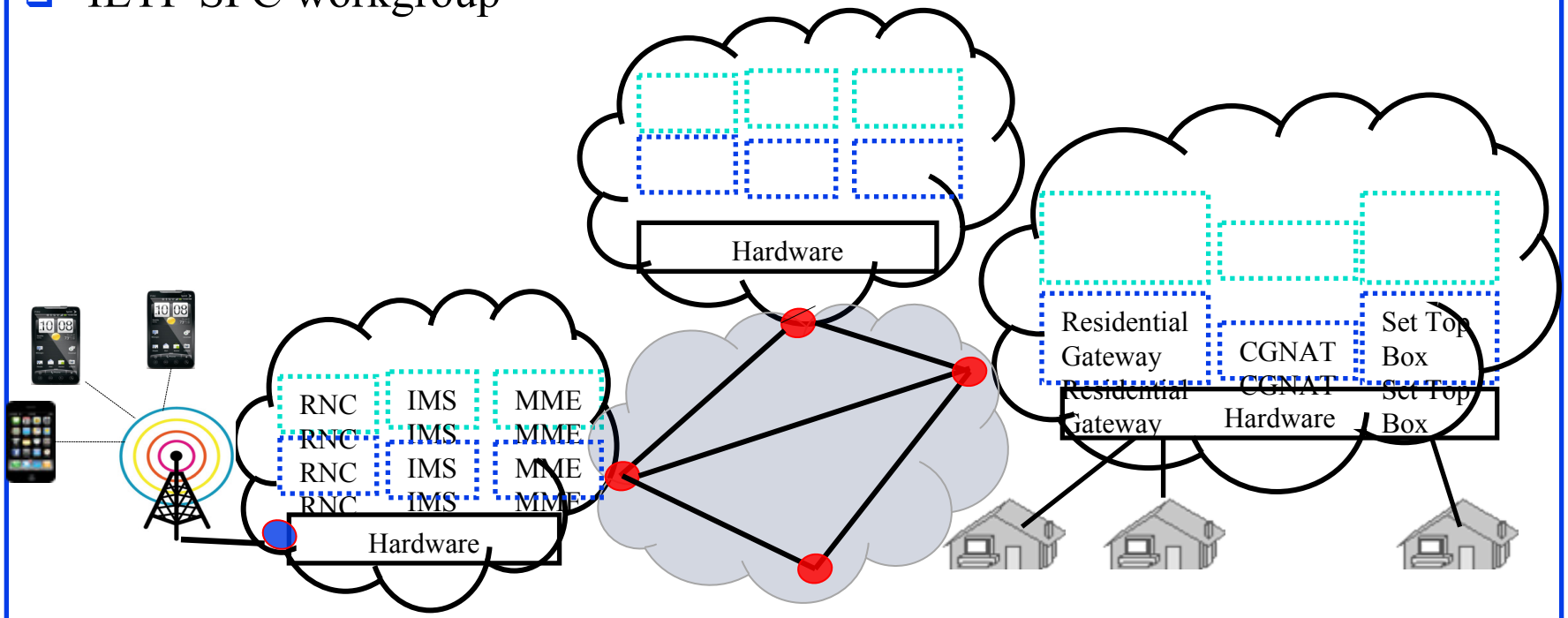
1. Virtual Broadband Remote Access Server (BRAS) by British Telecom
2. Virtual IP Multimedia System (IMS) by Deutsche Telekom
3. Virtual Evolved Packet Core (vEPC) by Orange Silicon Valley
4. Carrier-Grade Network Address Translator (CGNAT) and Deep Packet Inspection (DPI), Home Gateway by Telefonica
5. Perimeta Session Border Controller (SBC) from Metaswitch
6. Deep packet inspection from Procera

Most of these are based on Cloud technologies, e.g., OpenStack

Ref: M. Cohn, "NFV Group Flocks to Proof-of-Concept Demos," Aug 2013, http://www.sdncentral.com/technology/nfv-group-flocks-to-proof-of-concept-models/2013/08/

Service Chaining in a Multi-Cloud Multi-Tenant Environment

- VNFs (Virtual network fns) belong to tenants. Multiple tenants.
- Each Cloud belongs to a different Cloud Service Provider (CSP)
- Internet infrastructure belongs to an NFVI service provider (NSP)
- Service chain = Workflow
- IETF SFC workgroup



Any Function Virtualization (FV)

- Network function virtualization of interest to Network service providers
- But the same concept can be used by any other industry, e.g., financial industry, banks, stock brokers, retailers, mobile games, …
- Everyone can benefit from:
  - Functional decomposition of there industry
  - Virtualization of those functions
  - Service chaining those virtual functions (VFs)
    ⇒ A service provided by the next gen ISPs

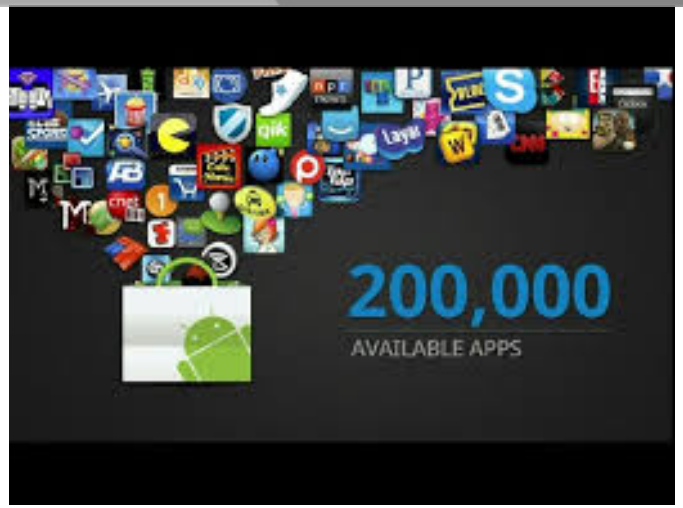
Enterprise App Market: Lower CapEx

Virtual IP Multimedia System

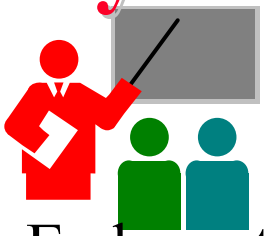
Available on the App Store

amazon.com® and you're done.™ App Store

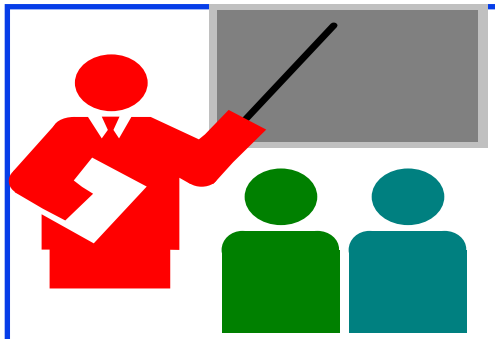
ANDROID



Summary of Part VIII



1. NFV aims to reduce OpEx by automation and scalability provided by implementing network functions as virtual appliances
2. NFV allows all benefits of virtualization and cloud computing including orchestration, scaling, automation, hardware independence, pay-per-use, fault-tolerance, …
3. NFV and SDN are independent and complementary. You can do either or both.
4. NFV requires standardization of reference points and interfaces to be able to mix and match VNFs from different sources
5. NFV can be done now. Several of virtual functions have already been demonstrated by carriers.



Overall Summary

1. Cloud computing requires Ethernet to be extended globally and partitioned for sharing by a very large number of customers who have complete control over their address assignment and connectivity and requires rapid provisioning of a large number of virtual NICs and switches
2. Data center bridging extensions reduce the packet loss by enhanced transmission selection and Priority-based flow control. Make Ethernet suitable for storage traffic.
3. Ethernet carrier extensions Q-in-Q (PB), MAC-in-MAC (PBB), and PBB-TE extensions allow Internet/Cloud service providers to allow customers to have their own VLAN IDs, MAC addresses, and QoS guarantees

Overall Summary (Cont)

4. Virtual Edge Bridge (VEB) vSwitches switch internally while Virtual Ethernet Port Aggregator (VEPA) vSwitches switch externally.
5. TRILL, NVGRE, VXLAN, and STT allow Ethernet to span a large campus or data center
6. OpenFlow separates control plane and moves it to a central controller ⇒ Simplifies the forwarding element
7. SDN is the framework to automatically manage and control a large number of multi-tenant network devices and services
8. NFV reduces OpEx by automation and scalability provided by implementing network functions as virtual appliances

Acronyms

- ACI Application Policy Infrastructure
- ACL Access Control List
- ADC Application Delivery Controller
- AEX Application Information Exposure
- ALG Application Level Gateway
- ALTO Application Layer Traffic Optimization
- ANDSF Access Network Discovery and Selection Function
- API Application Programming Interface
- APIC Application Policy Infrastructure Controller
- ARP Address Resolution Protocol
- ASICs Application Specific Integrated Circuit
- ATIS Association for Telecom Industry Solutions
- ATM Asynchronous Transfer Mode
- AVNP Active Virtual Network Management Protocol
- BER Bit Error Rate
- BFD Bidirectional Forwarding Detection

Acronyms (Cont)

- BGP Border Gateway Protocol
- BIRD Bird Internet Routing Daemon
- BNC Big Switch Network Controller
- BRAS Broadband Remote Access Server
- BSD Berkeley Software Distribution
- BSS Business Support Systems
- BUM Broadcast, Unknown, and Multicast
- CapEx Capital Expenditure
- CD Compact Disk
- CDN Content Distribution Network
- CDNI Content Distribution Network Interconnection
- CE Control Element
- CFI Canonical Format Indicator
- CFM Connectivity Fault Management
- CGNAT Carrier-Grade Network Address Translator
- CGSN Combined GPRS Support Node

Acronyms (Cont)

- CLI Command Line Interface
- CMS Content Management System
- COTS Commercial-off-the-shelf
- CPU Central Processing Unit
- CRC Cyclic Redundancy Check
- CRUD Create, Read, Update, Delete
- CSMA/CD Carrier Sense Multiple Access with Collision Detection
- CSP Cloud Service Provider
- DA Destination Address
- DCB Data Center Bridging
- DCBX Data Center Bridging Exchange
- DDIO Data Direct I/O Technology
- DEI Drop Eligibility Indicator
- DFCA Dynamic Frequency Channel Allocation
- DHCP Dynamic Host control Protocol
- DNS Domain Name Service

Acronyms (Cont)

- DOVE Distributed Overlay Virtual Ethernet
- DPI Deep Packet Inspection
- DSCP Differentiated Service Control Point
- DVS Distributed Virtual Switch
- ECMP Equal-cost multi-path
- EID Endpoint Identifier
- EMS Element Management System
- ENNI Ethernet Network to Network Interface
- EPL Ethernet Private Line
- ESP Encrytec Security Payload
- ETS Enhanced Transmission Service
- ETSI European Telecom Standards Institute
- EVC Ethernet Virtual Channel
- EVP-Tree Ethernet Virtual Private Tree
- EVPL Ethernet Virtual Private Line
- EVPLAN Ethernet Virtual Private LAN

Acronyms (Cont)

- EVPN Ethernet Virtual Private Network
- FCAPS Faults, configuration, accounting, performance, and security
- FCoE Fibre Channel over Ethernet
- FE Forwarding Element
- FEX Fabric Extension
- FIB Forwarding information base
- ForCES Forwarding and Control Element Separation
- GB Giga Byte
- GGSN Gateway GPRS Support Node
- GMPLS Generalized Multi-Protocol Label Switching
- GRE Generic Routing Encapsulation
- GUI Graphical User Interface
- HLR Home Location Register
- HSRP Hot Standby Router Protocol
- HTML Hypertext Markup Language
- HTTP Hypertext Transfer Protocol

Acronyms (Cont)

- I2AEX Infrastructure to Application Information Exposure
- IaaS Infrastructure as a Service
- IANA Internet Addressing and Naming Authority
- ICMP Internet Control Message Protocol
- ICSI International Computer Science Institute
- ID Identifier
- IDS Intrusion Detection System
- IEEE Institution of Electrical and Electronic Engineers
- IETF Internet Engineering Task Force
- IGMP Internet Group Management Protocol
- IGP Interior Gateway Protocol
- IMS IP Multimedia System
- INF Architecture for the virtualization Infrastructure
- IO Input/Output
- IoT Internet of Things
- IP Internet Protocol

Acronyms (Cont)

- IPFIX IP Flow Information Export Protocol
- IPSec IP Security
- IPv4 Internet Protcol version 4
- IPv6 Internet Protocol version 6
- IRTF Internet Research Taskforce
- IS-IS Intermediate System to Intermediate System
- iSCSI Internet Small Computer Storage Interconnect
- ISG Industry Specification Group
- ISO International Standards Organization
- JSON Java Script Object Notation
- JVM Java Virtual Machine
- kB Kilo Byte
- KVM Kernel-based Virtual Machine
- LACP Link Aggregation Control Protocol
- LAN Local Area Network
- LISP Locator-ID Separation Protocol

Acronyms (Cont)

- LLDP Link Layer Discovery Protocol
- LRO Large Receive Offload
- LS Link State
- LSO Large Send Offload
- LSP Label Switched Path
- MAC Media Access Control
- MAN Metropolitan Area Network
- MANO Management and orchestration
- MDI Media Dependent Interface
- MME Mobility Management Entity
- MPLS Multi-protocol Label Switching
- MR-IOV Multi-Root I/O Virtualization
- MSB Most Significant Byte
- MSS Maximum Segment Size
- MST Multiple spanning tree
- MSTP Multiple Spanning Tree Protocol

Acronyms (Cont)

- MTU Maximum Transmission Unit
- MVGRE Network Virtualization Using GRE
- NAT Network Address Translation
- NF Network Function
- NFV Network Function Virtualization
- NFVI Network Function Virtualization Infrastructure
- NFVIaaS NFVI as a Service
- NIB Network Information Base
- NIC Network Interface Card
- NNI Network-to-Network Interface
- NSF National Science Foundation
- NTP Network Time Protocol
- NTT Nippon Telegraph and Telephone
- NVGRE Network Virtualization using Generic Routing Encapsulation
- NVO3 Network Virtualization over L3
- NVP Network Virtualization Platform

Acronyms (Cont)

- OAM Operation, Administration, and Management
- OF OpenFlow
- OFlops OpenFlow Operations Per Second
- OLSR Optimized Link State Routing
- ON.LAB Open Networking Lab at Stanford
- OnePK Open Network Environment Platform Kit
- ONF Open Networking Foundation
- ONV OpenDaylight Network Virtualization
- openQRM Open Clusters Resource Manager
- OpenWRT Open WRT54G (Linksys product name) software
- OpEx Operation Expenses
- OS Operating System
- OSCP OpenDaylight SDN Controller Platform
- OSGi Open Services Gateway Initiative
- OSPF Open Shortest Path First
- OSS Operation Support System

Acronyms (Cont)

- OTN Optical Transport Network
- OTV Overlay Transport Virtualization
- OVS Open Virtual Switch
- OVSDB Open Virtual Switch Database
- PaaS Platform as a Service
- PB Provider Bridge
- PBB-TE Provider Backbone Bridge with Traffic Engineering
- PBB Provider Backbone Bridge
- PBEB Provider Backbone Edge Bridge
- PCC Path Computation Client
- PCE Path Computation Element
- PCEP Path Computation Element Protocol
- PCI-SIG PCI Special Interest Group
- PCI Peripheral Component Interconnect
- PCIe PCI Express
- PCP Priority Code Point

Acronyms (Cont)

- PE Provider Edge
- PF Physical Function
- PFC Priority-based Flow Control
- PGW Packet Data Network Gateway
- PHY Physical Layer
- PIM-SM Protocol Independent Multicast - Sparse Mode
- PIM Protocol Independent Multicast
- pM Physical Machine
- pNIC Physical Network Interface Card
- PoC Proof-of-Concept
- PoP Point of Presence
- PPP Point-to-Point Protocol
- PSTN Public Switched Telephone Network
- pSwitch Physical Switch
- PW Pseudo wire
- PWE3 Pseudo wire Emulation Edge to Edge

Acronyms (Cont)

- PWoGRE Pseudo wire over Generic Routing Encapsulation
- PWoMPLS Pseudo wire over Multi Protocol Label Switching
- QCN Quantized Congestion Notification
- QoS Quality of Service
- RAID Redundant Array of Independent Disks
- RAN Radio area networks
- RBridge Routing Bridge
- REL Reliability, Availability, resilience and fault tolerance group
- REST Representational State Transfer
- RFC Request for Comments
- RGW Residential Gateway
- RIB Routing Information Base
- RIP Routing Information Protocol
- RLOC Routing Locator
- RNC Radio Network Controller
- RPC Remote Procedure Call

Acronyms (Cont)

- RS Routing System
- RSPAN Remote Switch Port Analyzer
- RSTP Rapid Spanning Tree Protocol
- SA Source Address
- SaaS Software as a Service
- SAL Service Abstraction Layer
- SBC Session Border Controller
- SDH Synchronous Digital Hierarchy
- SDN Software Defined Networking
- SGSN Serving GPRS Support Node
- SGW Serving Gateway
- SID Service Identifier
- SIP Session Initiation Protocol
- SLA Service Level Aggrement
- SMTP Simple Mail Transfer Protocol
- SNAC Name of an OpenFlow controller

Acronyms (Cont)

- SNIA Storage Network Industry Association
- SNMP Simple Network Management Protocol
- SONET Synchronous Optical Network
- SPAN Switch Port Analyzer
- SPB Shortest Path Bridging
- SR-IOV Single Root I/O Virtualization
- SSH Secure Socket Host
- SSL Secure Socket Layer
- STP Spanning Tree Protocol
- STT Stateless TCP-like Transport
- SWA Software architecture
- TAS Telephony Application Server
- TCAM Ternary Content Addressable Memory
- TCL Tool Command Language
- TCP Transmission Control Protocol
- TE Traffic Engineering

Acronyms (Cont)

- TIA Telecom Industry Association
- TLS Transport Level Security
- TLV Type-Length-Value
- TMF TM Forum
- ToS Type of Service
- TP Transport Protocol
- TPI Tag Protocol Identifier
- TRILL Transparent Interconnection of Lots of Links
- TTL Time to Live
- TTP Table Typing Patterns
- TV Television
- UC University of California
- UCA Use Customer Address
- UDP User Datagram Protocol
- UNI User Network Interface
- URI Uniform Resource Identifier

Acronyms (Cont)

- VBE Virtual Bridge Port Extension
- vBridge Virtual Bridge
- VDC Virtual Device Contexts
- VEB Virtual Edge Bridge
- VEM Virtual Ethernet Module
- VEPA Virtual Ethernet Port Aggregator
- vEPC Virtual Evolved Packet Core
- VF Virtual Function
- VID VLAN ID
- VIRL Virtual Internet Routing Lab
- VLAN Virtual LAN
- VM Virtual Machine
- VNF Virtual Network Function
- VNFaaS VNF as a Service
- VNI Virtual Network ID
- vNIC Virtual Network Interface Card

Acronyms (Cont)

- VNS Virtual Network Segement
- VoD Video on Demand
- VOIP Voice over IP
- vPC Virtual Port Channels
- VPLS Virtual Private LAN Service
- VPN Virtual Private Network
- VRF Virtual Routing and Forwarding
- VRRP Virtual Router Redundancy Protocol
- VSID Virtual Subnet Identifier
- VSM Virtual Switch Module
- VSS Virtual Switch System
- vSwitch Virtual Switch
- VT-d Virtualization Technology for Direct IO
- VT-x Virtualization Technology
- VTEP Virtual Tunnel End Point
- VTN Virtual Tenant Network

Acronyms (Cont)

- VXLAN Virtual Extensible LAN
- WAN Wide Area Network
- WG Working Group
- XML Extensible Markup Language
- XMPP Extensible Messaging and Presence Protocol
- XORP eXensible Open Router Platform