NYU TANDON SCHOOL OF ENGINEERING

Data Center Networks II

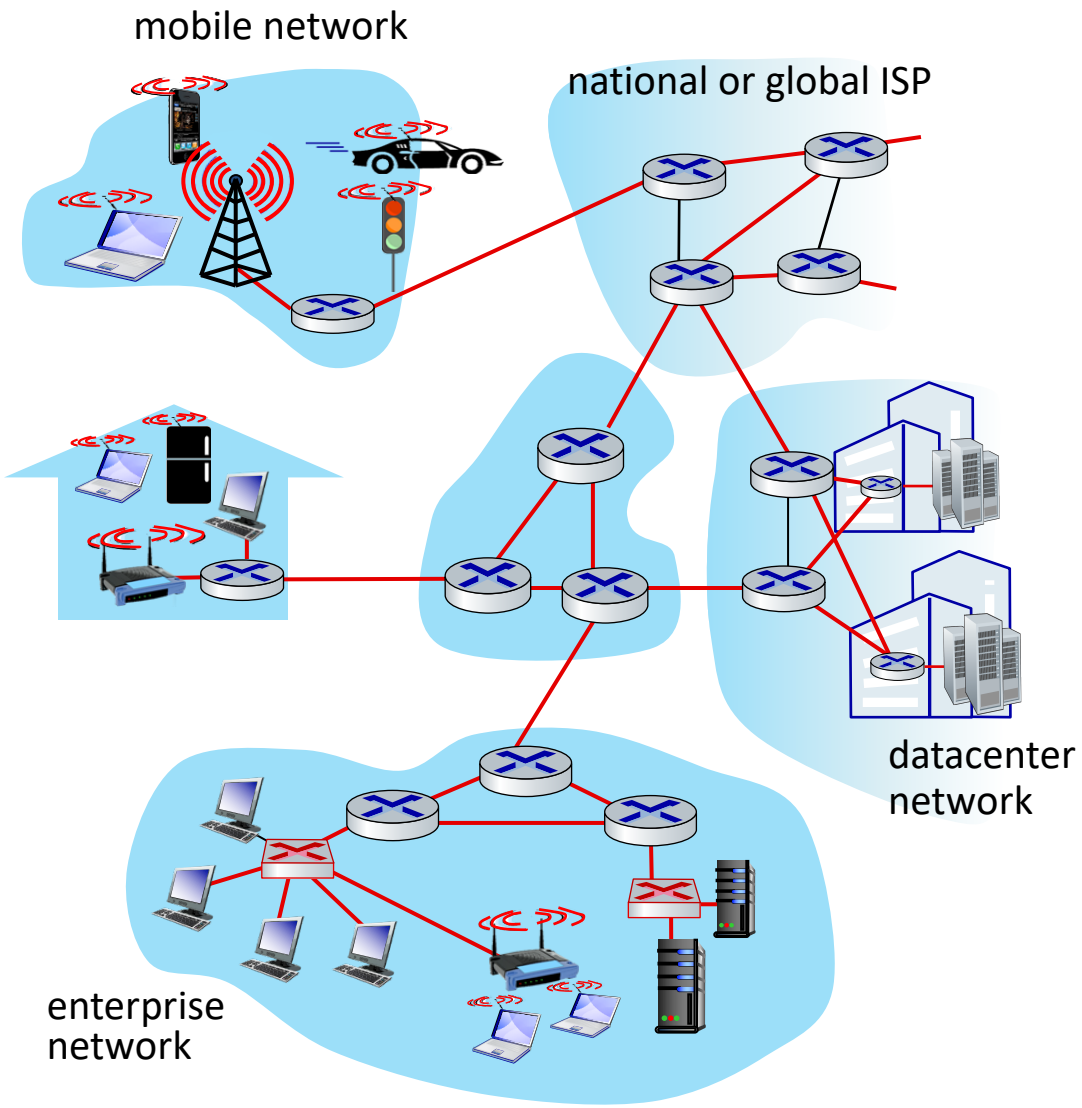
H. Jonathan Chao
ECE Department
chao@nyu.edu

Outline

1. Review of Layer2 and Layer 3
2. Cisco Data Center Network 3.0
3. Portland Data Center Network (DCN)
4. VL2 (Virtual Layer 2) DCN

1. Review of Layer2 and Layer 3

Naming & Addressing in Network



Layer2/Layer3 Overview

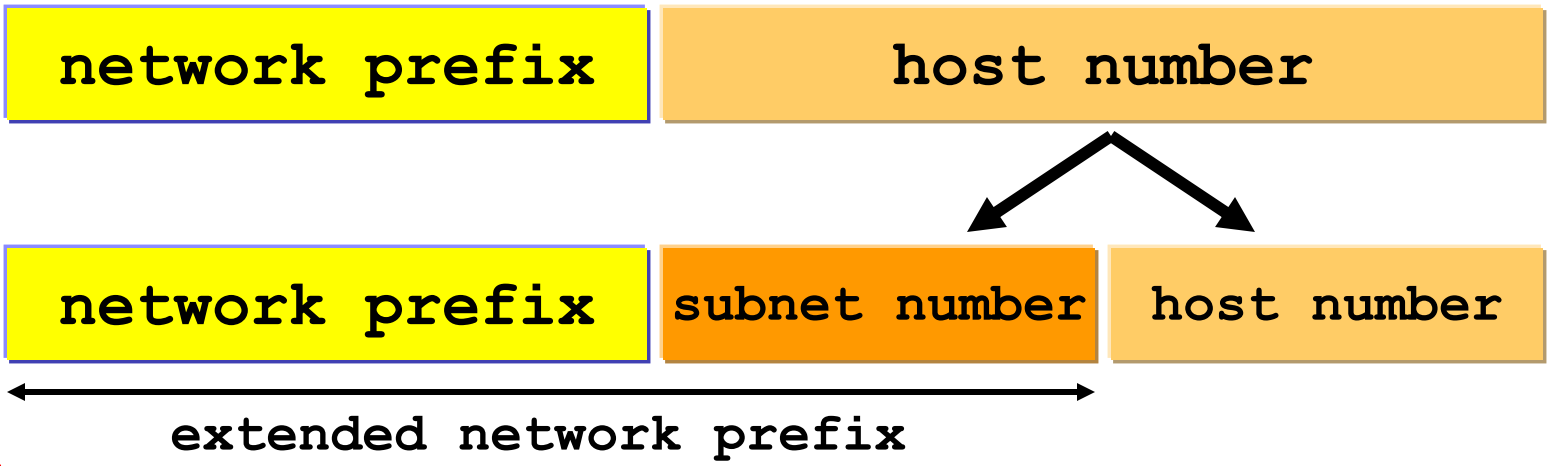
- Layer 2 (Data link layer)
  - Addressing: MAC address
  - Learning: Flooding
  - Switched
  - Minimum Spanning Tree
  - Semantics:
    - Unicast
    - Multicast
    - Broadcast

- Layer 3 (Network layer)
  - Addressing: IP Address
  - Learning: Routing protocol
    - Dynamic
      - BGP
      - OSPF
    - Static
  - Routed

- ARP
  - Discovery protocol
  - Ties IP back to MAC
  - Utilizes layer 2 broadcast semantics

Subnetting in Enterprise

- Split the host number portion of an IP address into a **subnet number** and a (smaller) **host number**.
- Result is a 3-layer hierarchy

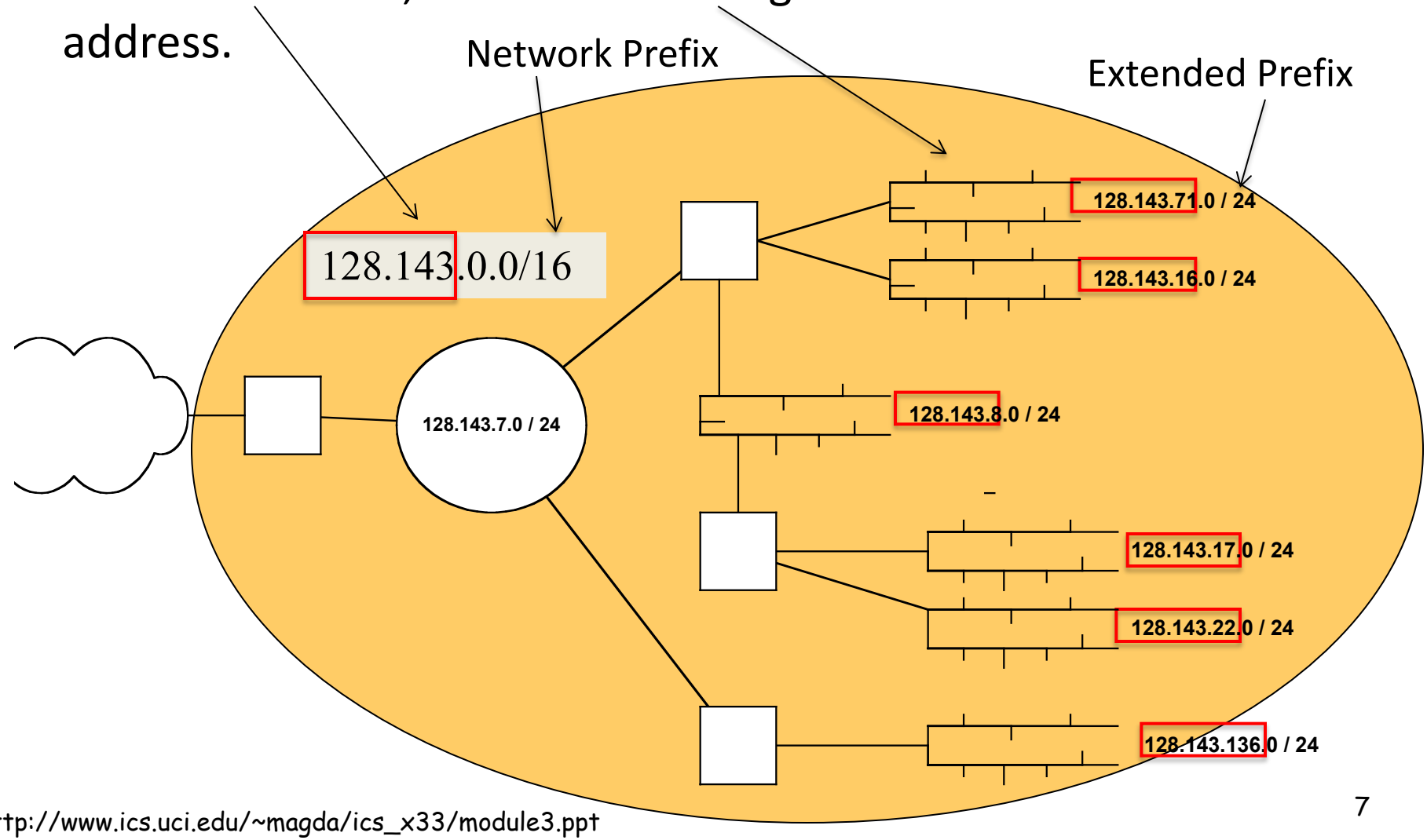


- Then:
  - Subnets can be freely assigned within the organization
  - Internally, subnets are treated as separate networks
  - Subnet structure is not visible outside the organization

http://www.ics.uci.edu/~magda/ics_x33/module3.ppt

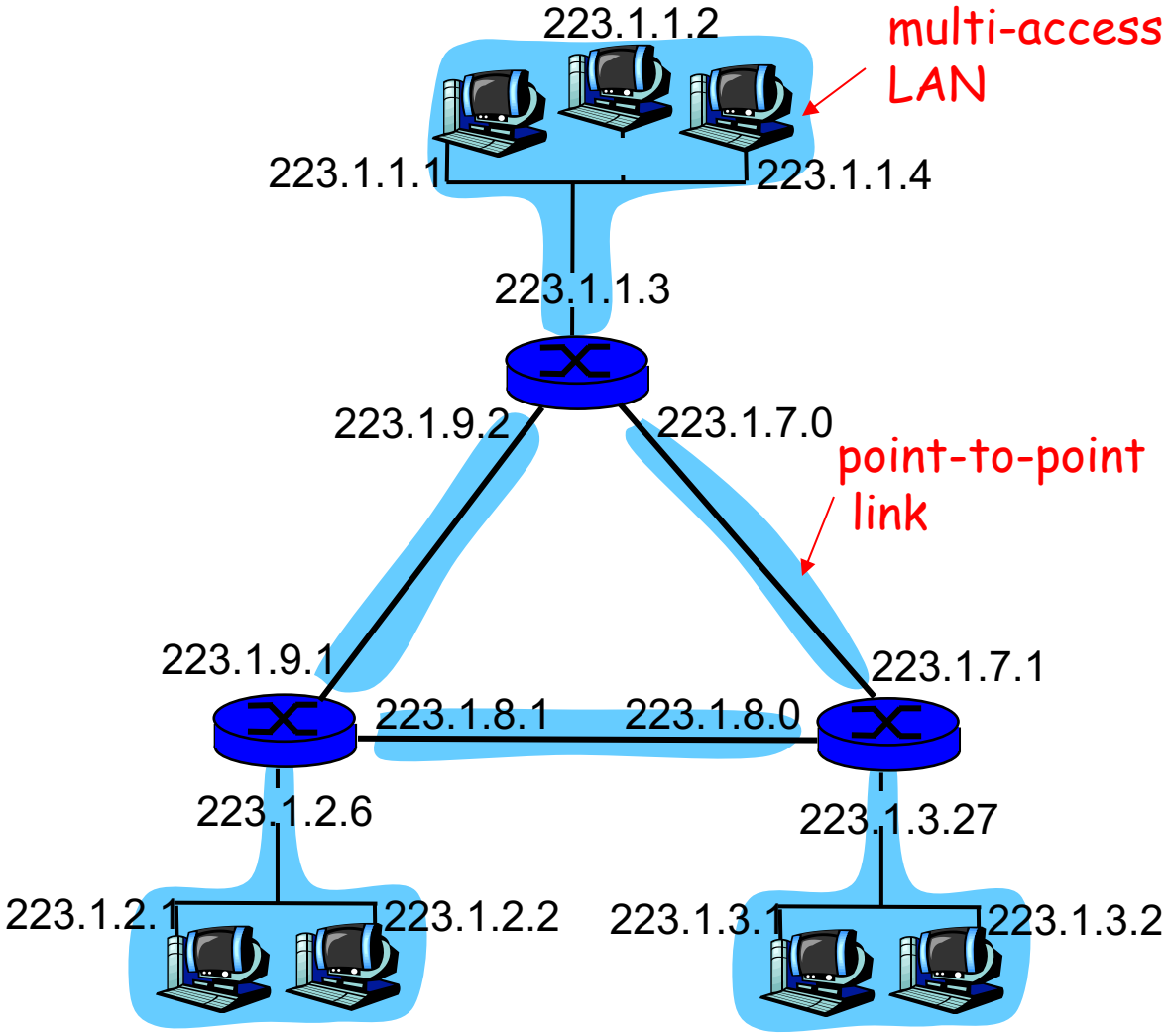
Typical addressing plan for an organization that uses subnetting

- In this IP domain, each Ethernet segment is allocated a subnet address.



IP Addressing: Subnet vs. Host

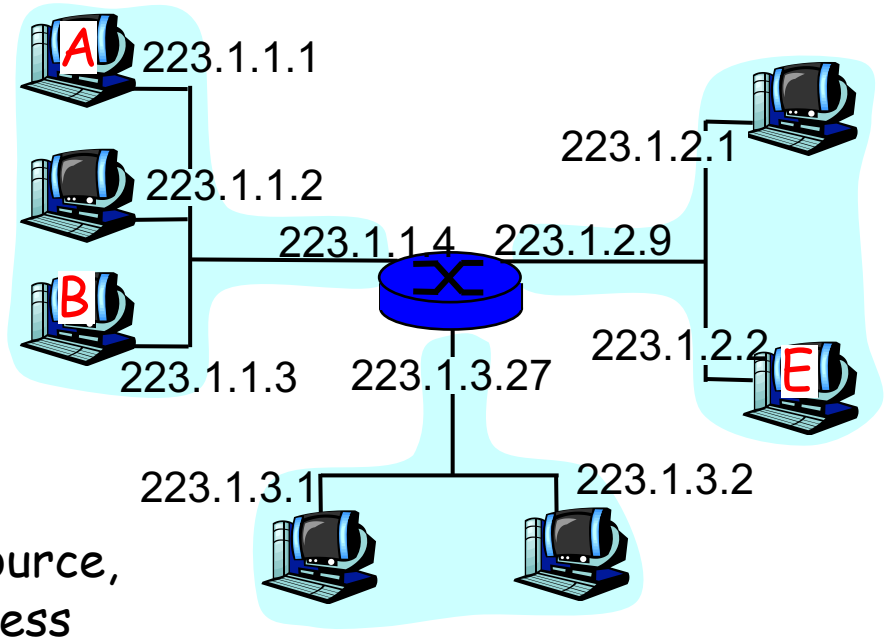
- Two-level hierarchy
  - Subnet (high order bits), reached by routers
  - Host part (low order bits), reached by Ethernet switches using hosts' MAC addresses
- *What's a subnet ?*
  - device interfaces with the same network part of IP address
  - can physically reach each other without intervening a router
  - devices on the same subnet can communicate through an Ethernet switch

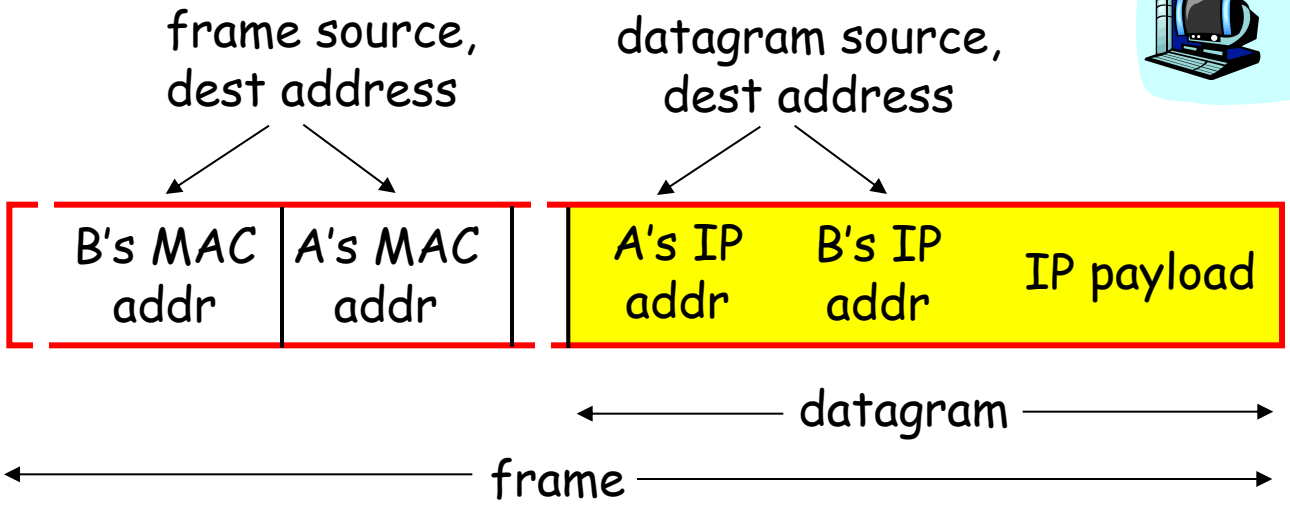


IP Datagram Forwarding on Same LAN: Interaction of IP and data link layers

Starting at A, given IP datagram addressed to B:

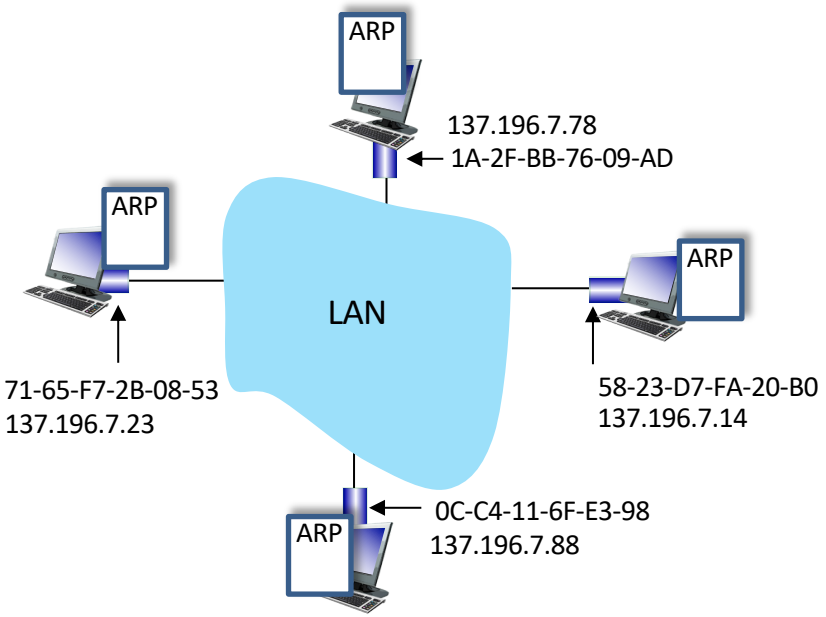
- look up net. address of B, find B on same net. as A
- link layer send datagram to B inside link-layer frame





ARP: address resolution protocol

Question: how to determine interface's MAC address, knowing its IP address?



ARP table: each IP node (host, router) on LAN has table

- IP/MAC address mappings for some LAN nodes:

  < IP address; MAC address; TTL>

- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

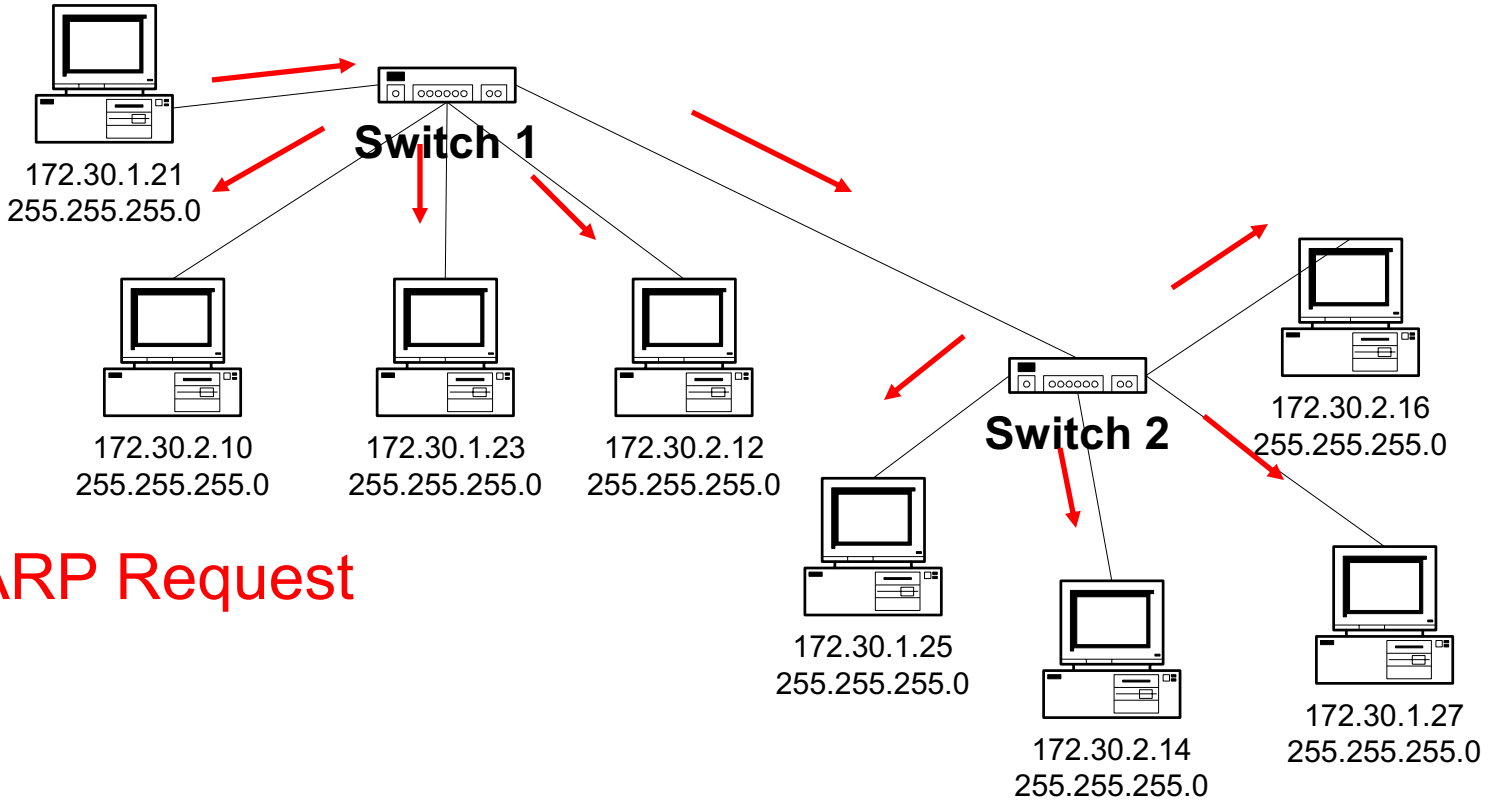
ARP Protocol

- A wants to send datagram to B, and A knows B's IP address.
- A looks up B's MAC address in its ARP table
- Suppose B's MAC address is not in A's ARP table.
- A broadcasts (why?) ARP query packet, containing B's IP address
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from net administrator

ARP Request & Response Processing

- The requester *broadcasts* ARP request
- The target node *unicasts* (why?) ARP reply to requester
  - With its physical address
  - Adds the requester into its ARP table (why?)
- On receiving the response, requester
  - updates its table, sets timer
- Other nodes upon receiving the ARP request
  - Refresh the requester entry if already there
  - No action otherwise (why?)
- Some questions to think about:
  - Shall requester buffer IP datagram while performing ARP?
  - What shall requester do if never receive any ARP response?

Broadcast domains



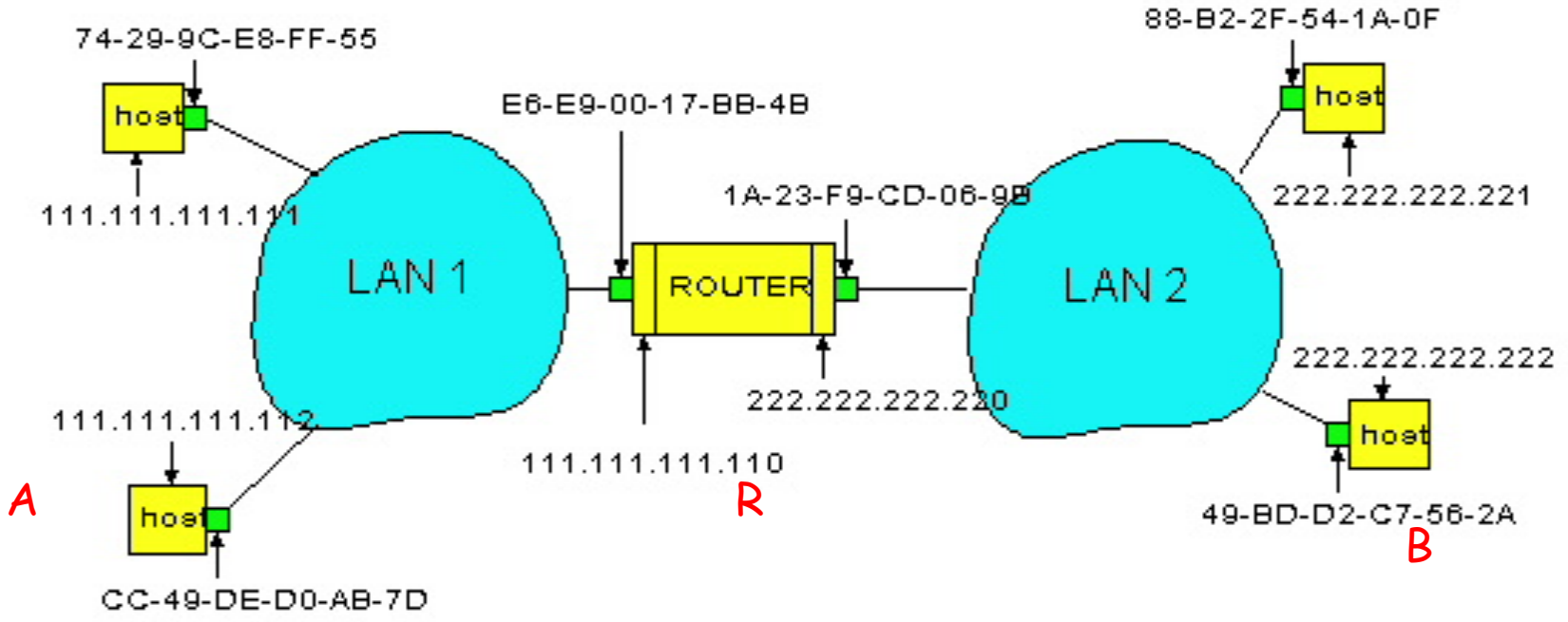
- ARP Request

- Even though the LAN switch reduces the size of collision domains, all hosts connected to the switch are still in the same broadcast domain.
- Therefore, a broadcast from one node will still be seen by all the other nodes connected through the LAN switch.

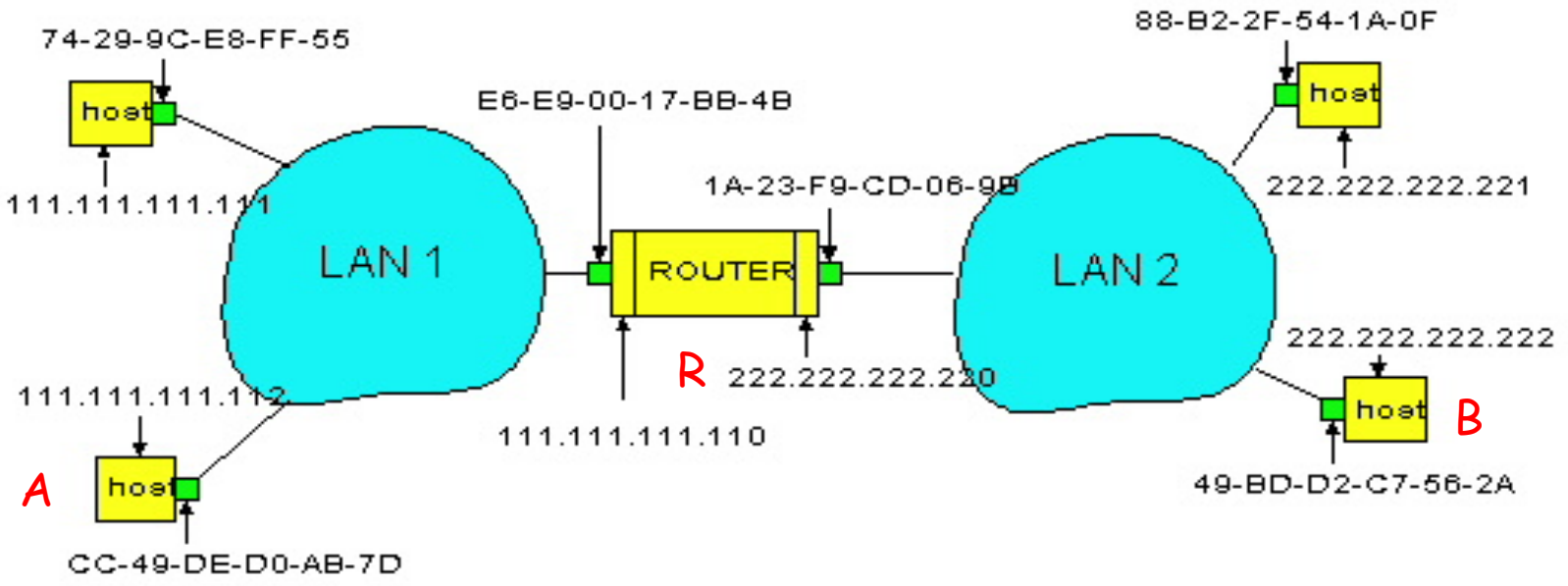
Forwarding to Another LAN:
Interaction of IP and Data Link Layer

walkthrough: send datagram from A to B via R

assume A knows B IP address



- Two ARP tables in router R, one for each IP network (LAN)
- In routing table at source host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B, etc



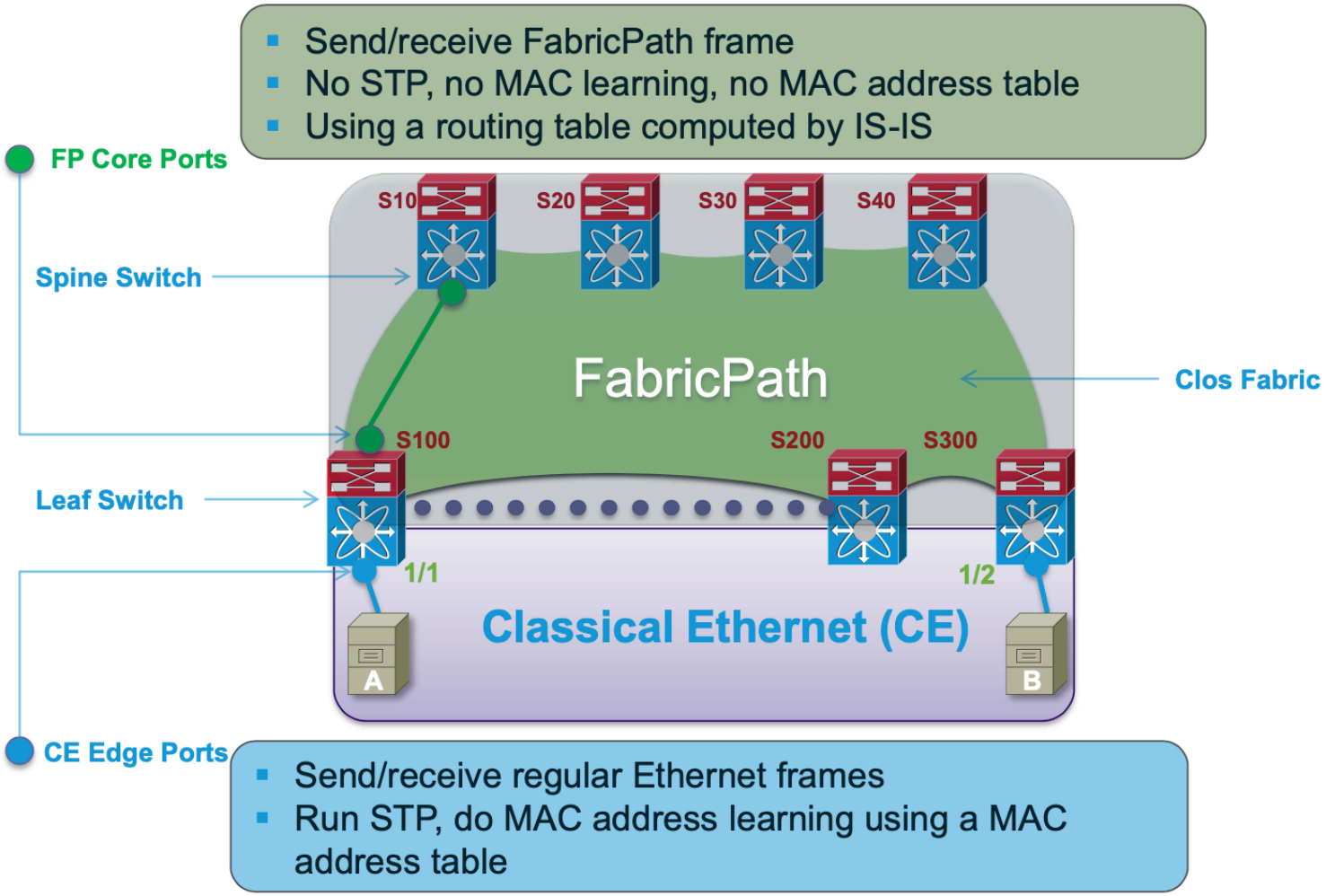
- Detailed Procedure
  - A creates datagram with source A, destination B
  - A uses ARP to get R's MAC address for 111.111.111.110
  - A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
  - A's data link layer sends frame
  - R's data link layer receives frame
  - R removes IP datagram from Ethernet frame, sees its destined to B
  - R uses ARP to get B's physical layer address
  - R creates frame containing A-to-B IP datagram sends to B

2. Cisco Data Center Network 3.0

Cisco FabricPath

- 2-tier DCN (Data Center Network) architecture
- Layer-2 forwarding
- Flat address space
- Support multipath routing and load balancing

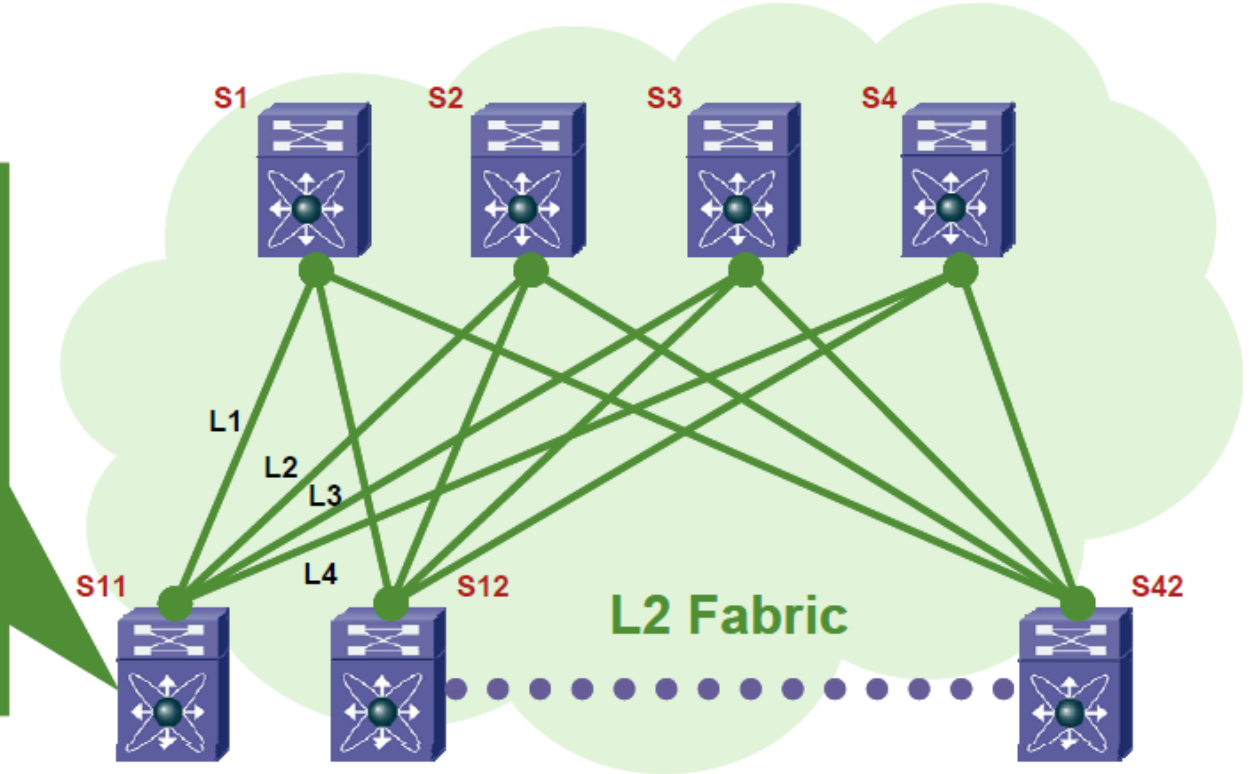
FabricPath and Classical Ethernet Plans



Routing Table in FabricPath Plan

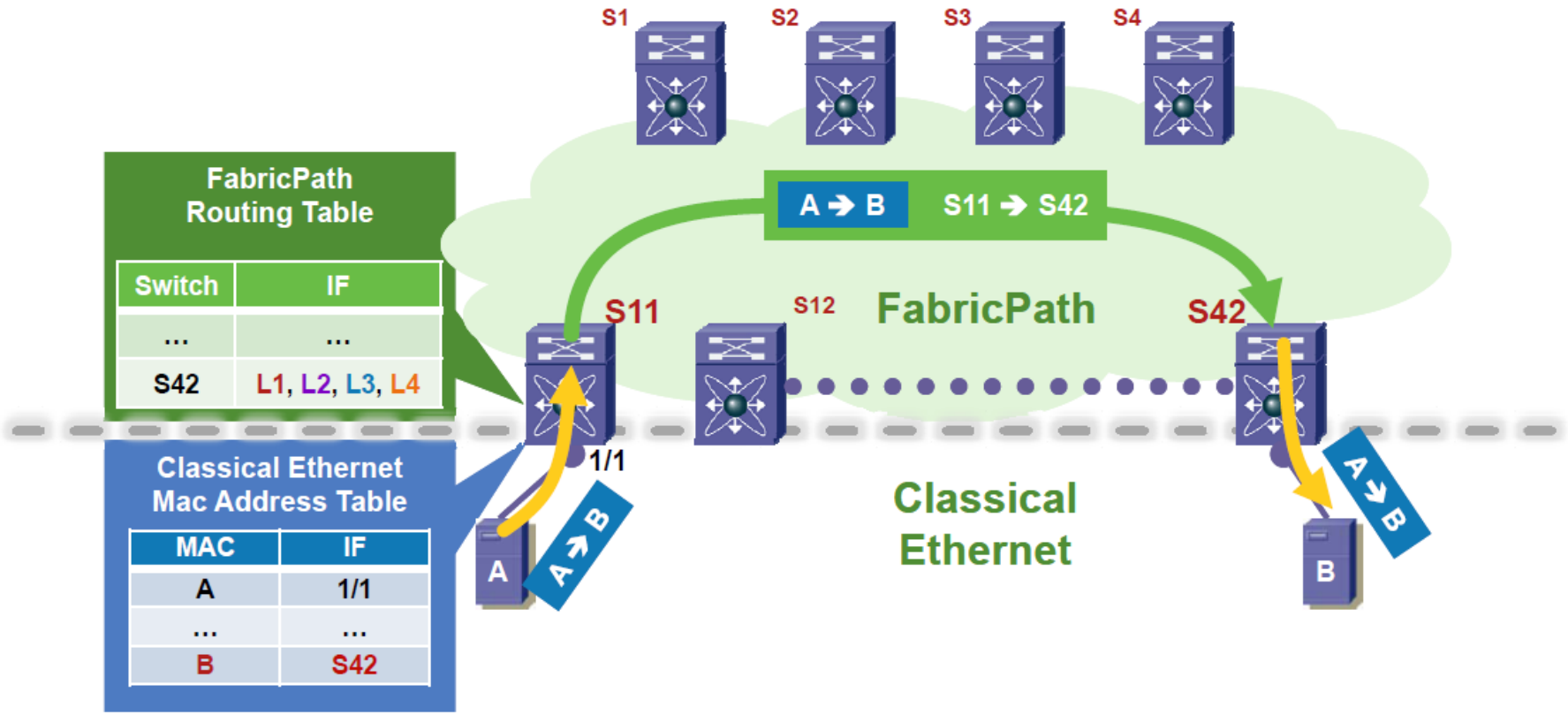
FabricPath Routing Table

Switch	IF
S1	L1
S2	L2
S3	L3
S4	L4
S12	L1, L2, L3, L4
…	…
S42	L1, L2, L3, L4

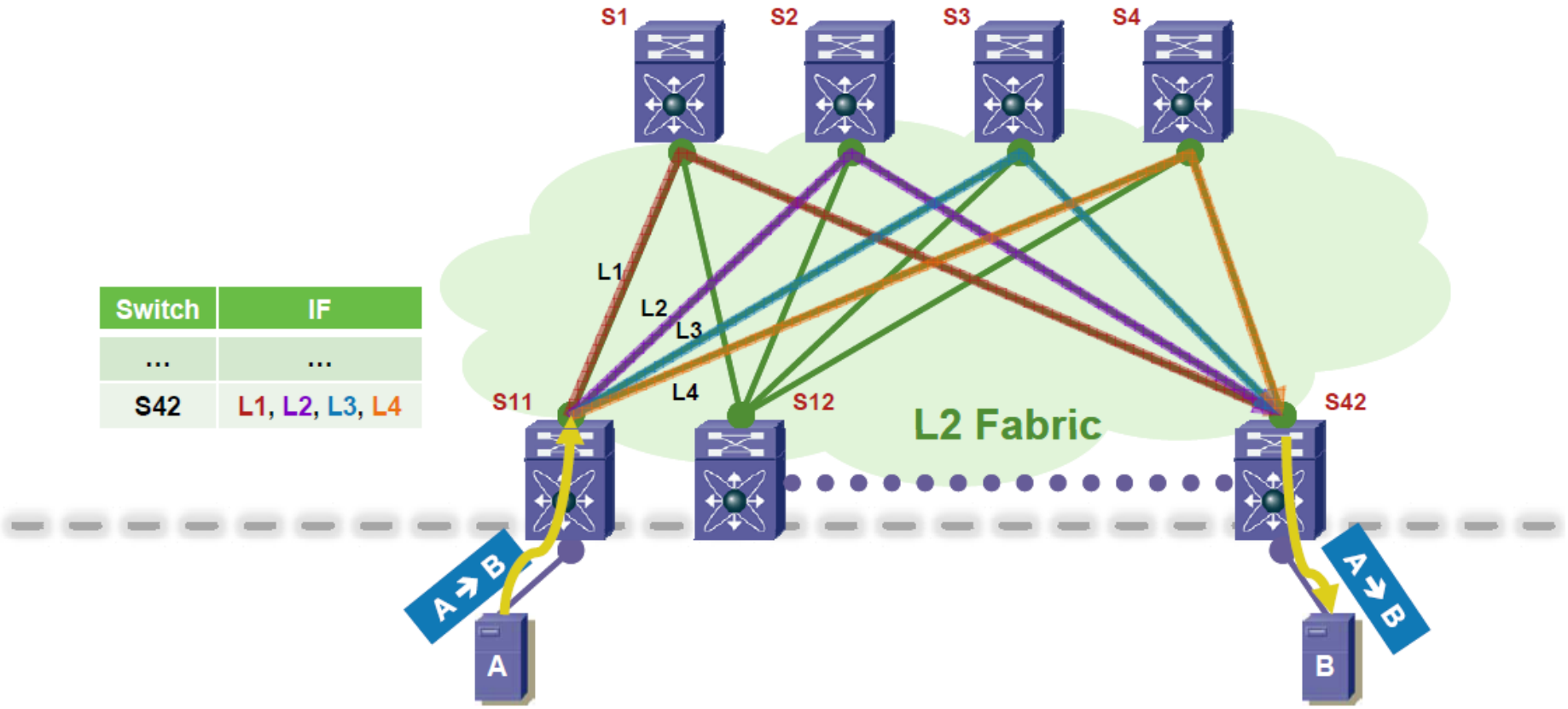


1. System assigns addresses to all FabricPath switches automatically
2. Compute shortest, pair-wise paths
3. Support equal-cost paths between any FabricPath switch pairs

Encapsulation & Forwarding

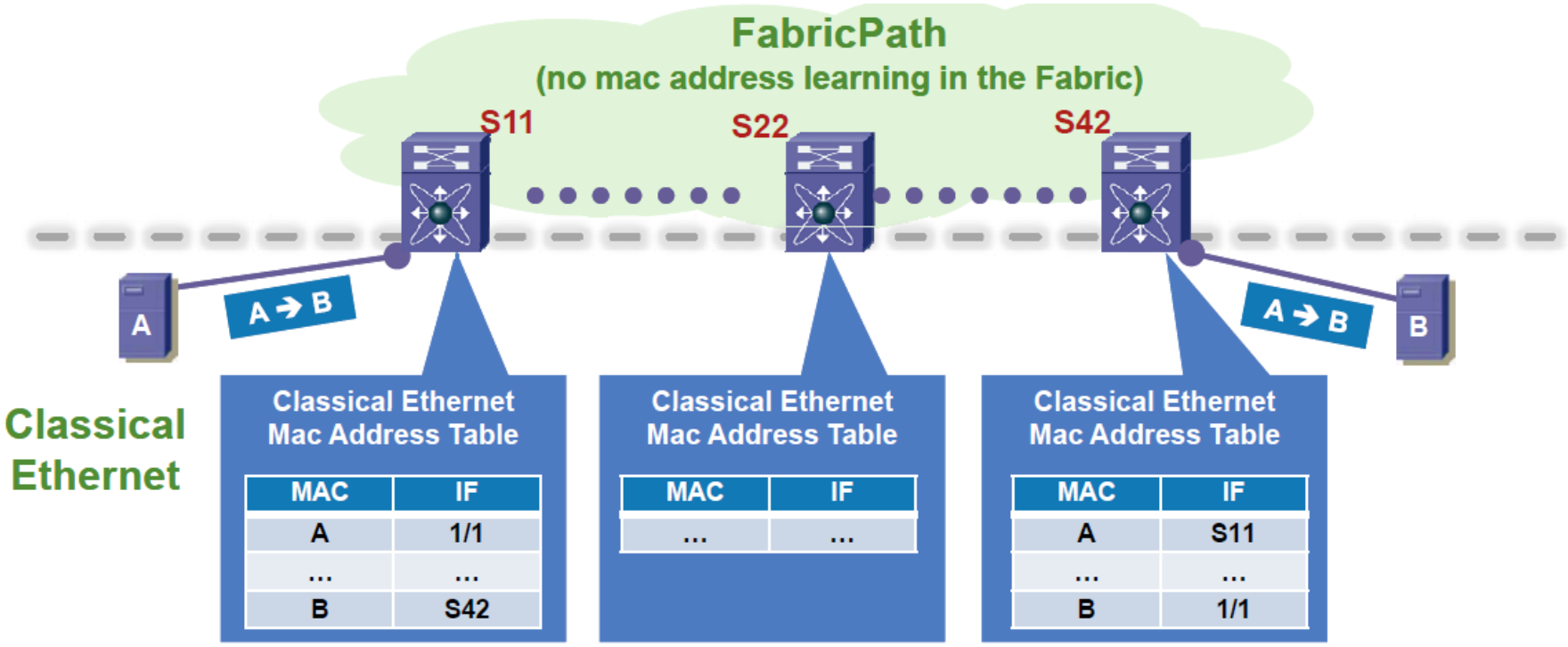


Multipath Fowarding



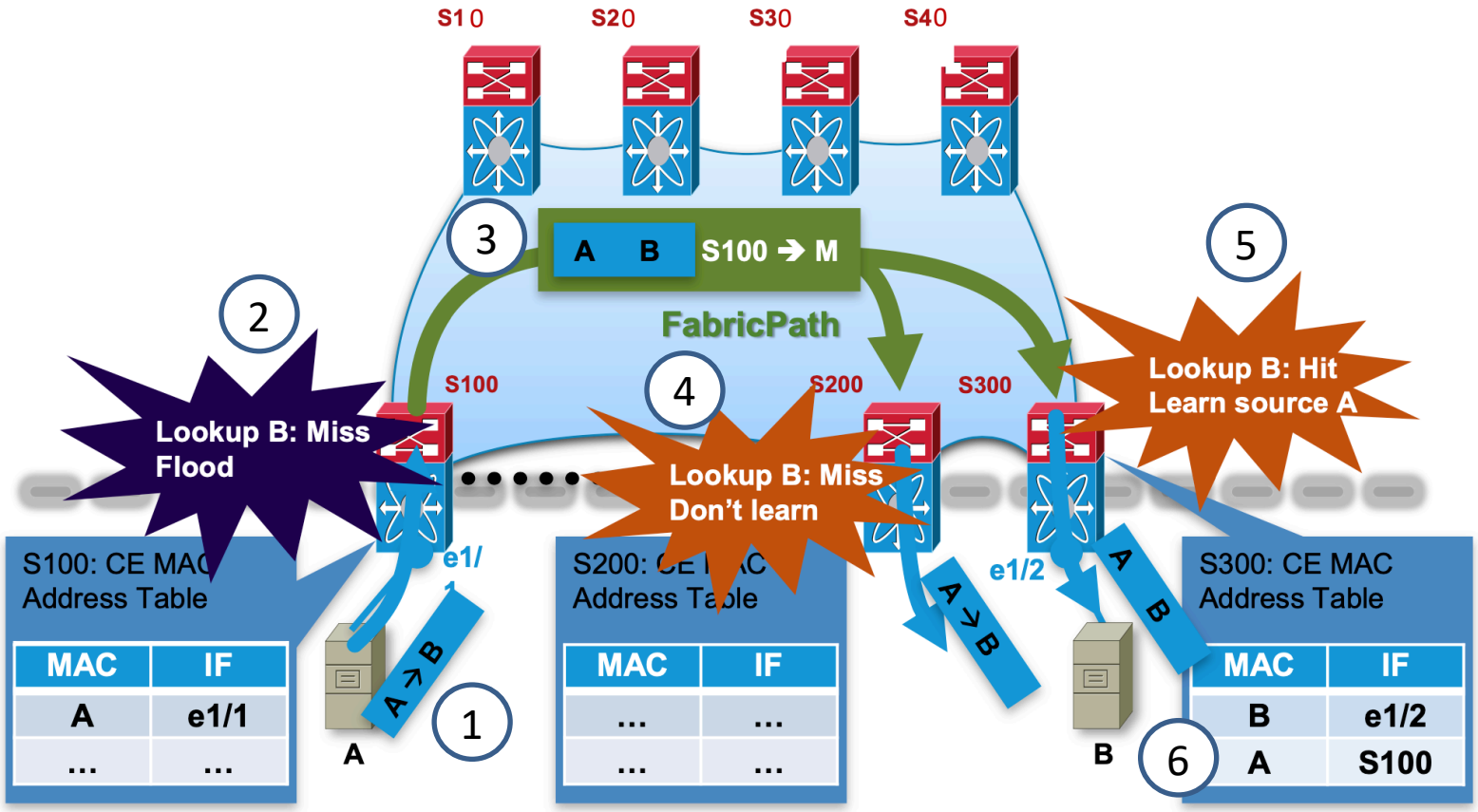
Switch	IF
…	…
S42	L1, L2, L3, L4

Access Forwarding Tables

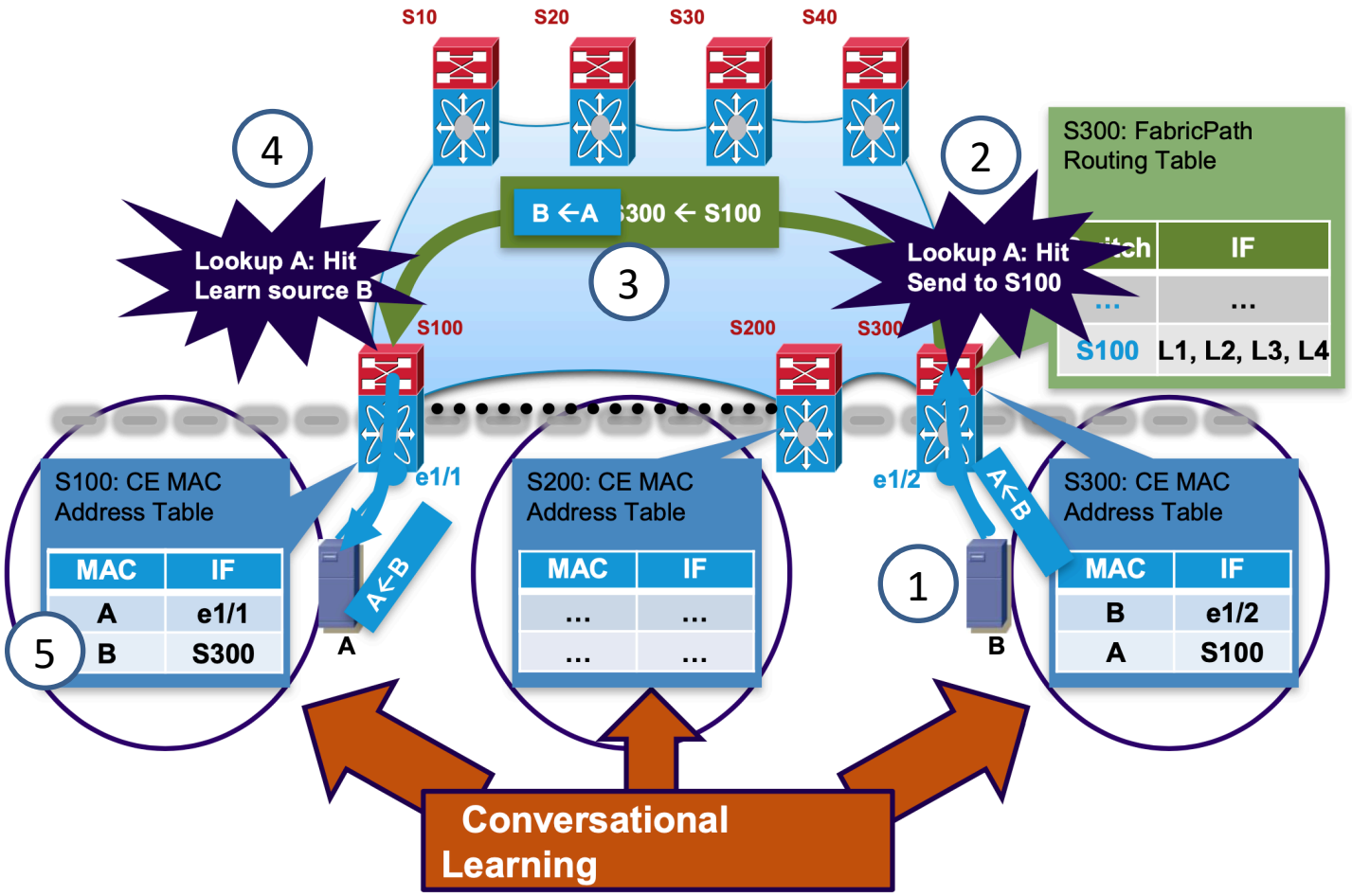


- Address learning by an edge switch is to bind a dest. MAC address to a dest. edge switch (using broadcast, like today's switches)
- No address learning is required in the FabricPath due to a Clos network connection

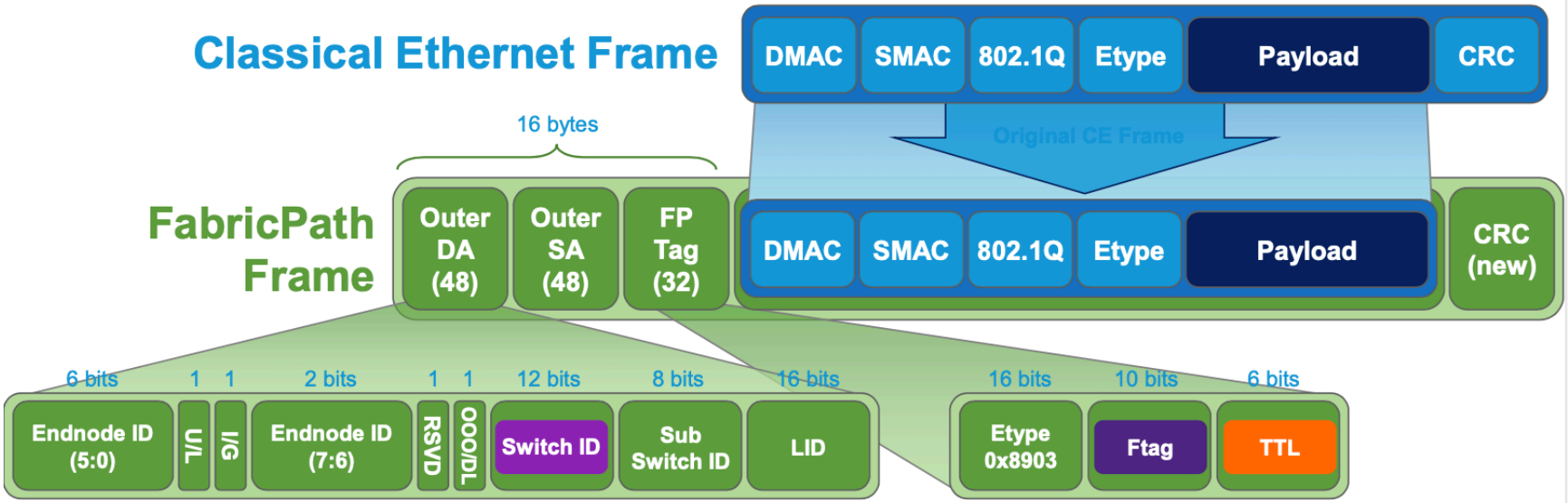
Unknown Unicast Flooding



Conversational MAC Learning



FabricPath Encapsulation



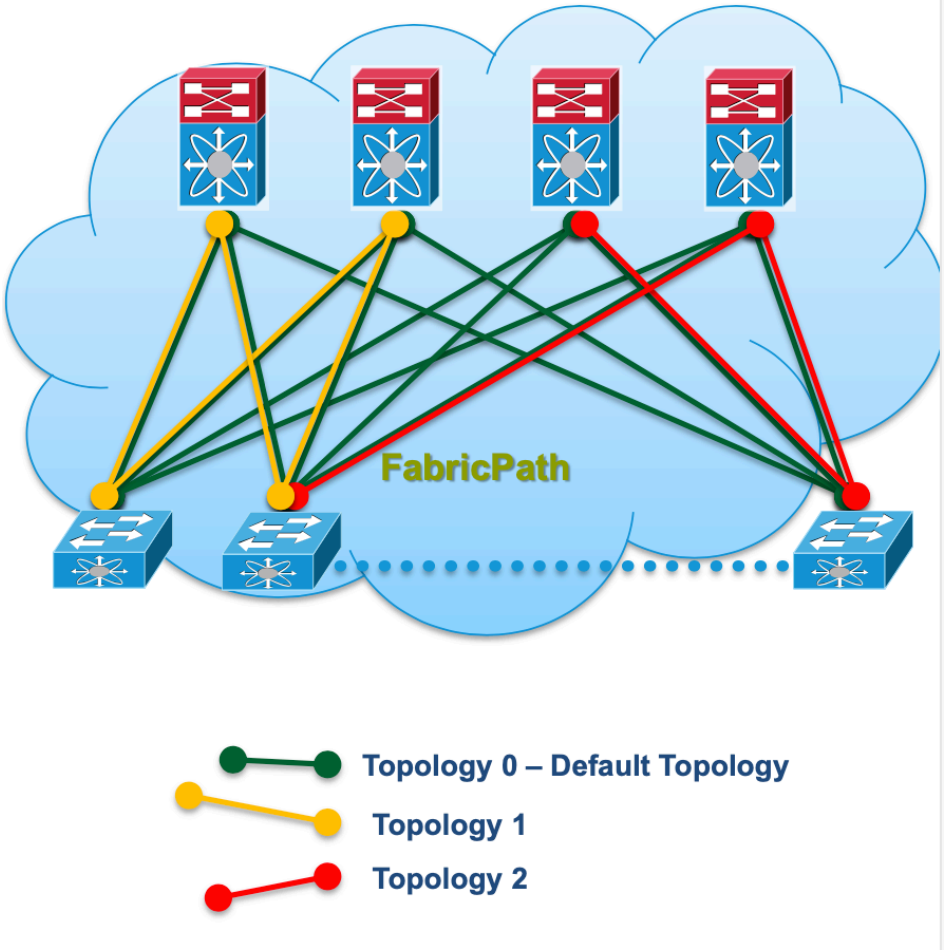
Switch ID – Unique number identifying each FabricPath switch

Ftag (Forwarding tag) – Unique number identifying topology and/or distribution tree

TTL – Decremented at each switch hop to prevent frames looping infinitely

Multiple Topologies

- Topology: A group of links in the Fabric. By default, all the links are part of topology 0
- A VLAN is mapped to a unique topology
- Other topologies can be created by assigning a subset of the links to them. A given link can belong to several topologies
- Topologies can be used for migration designs (i.e. VLAN localization), traffic engineering, security etc...



FabricPath Summary

- The association MAC address/Switch ID is maintained at the edge
- Introduce routing and ECMP (equal cost multi-paths) to the core network
- Traffic is encapsulated across the FabricPath by using MAC-in-MAC encapsulation to separate forwarding in the core and the edge
- Use conversational address learning (also called selective address learning) to reduce MAC table size (i.e., record the source MAC and its edge switch ID mapping only if the destination MAC is associated with the switch)

FabricPath Disadvantages

- Network is still 2-tier, not flat, per-packet store-and-forward in each single switch
- Hash-based load balancing among multiple paths may not give good load balancing when the flow sizes are not even.
- Broadcast and address learning could be a potential problem even with an improvement by using selective address learning
- In case of failure, the protocol needs to re-converge (Using SDN should be faster)

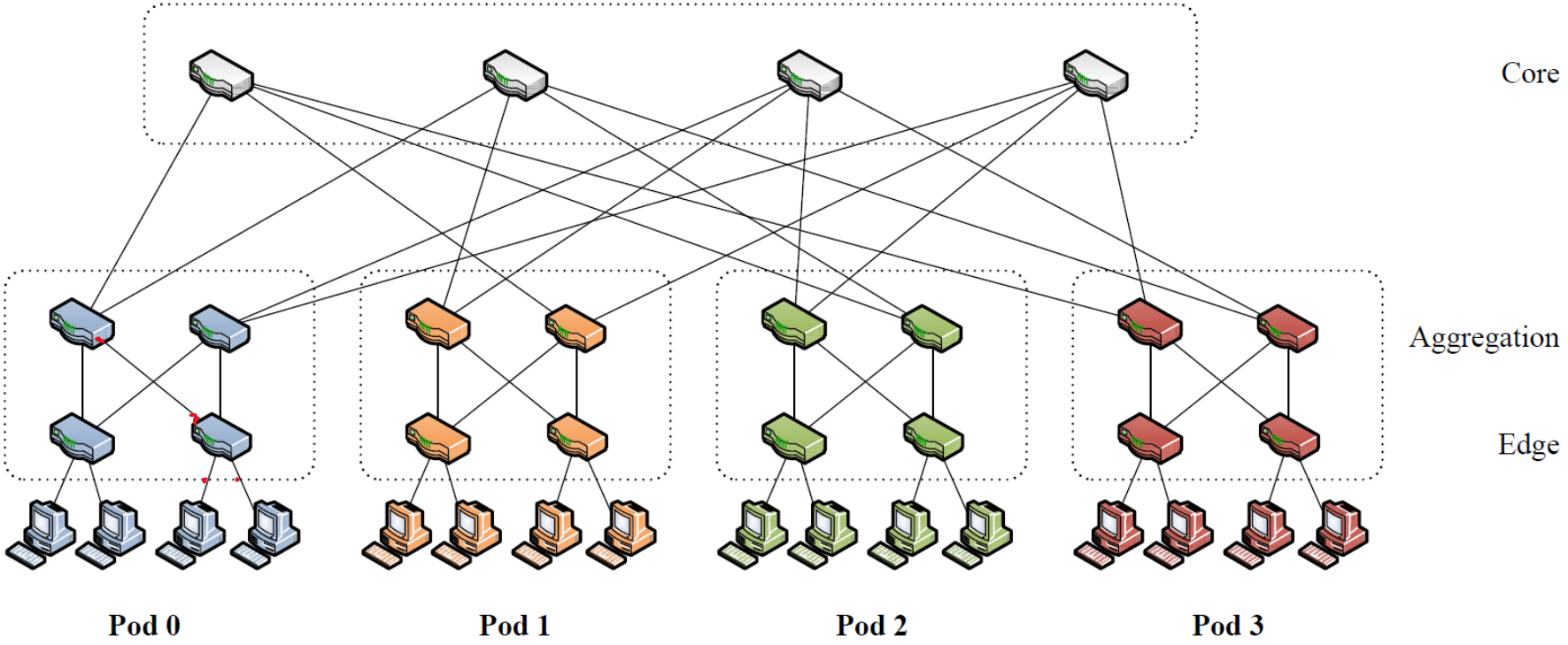
3. Portland Data Center Network (DCN)

PortLand Data Center Network Fabric

- Use many small switches
- All switches have the same number of ports
- Modify switches, no modifications to servers
- PortLand internally separates host identity from host location
  - Uses IP address as host identifier
  - Introduces “Pseudo MAC” (PMAC) addresses internally to encode endpoint location

- R. Mysore, et al, “PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric”, Sigcomm 2009

Fat Tree Topology



Switch: k ports

aggregation switches/pod: k/2

core switches: (k/2)^2

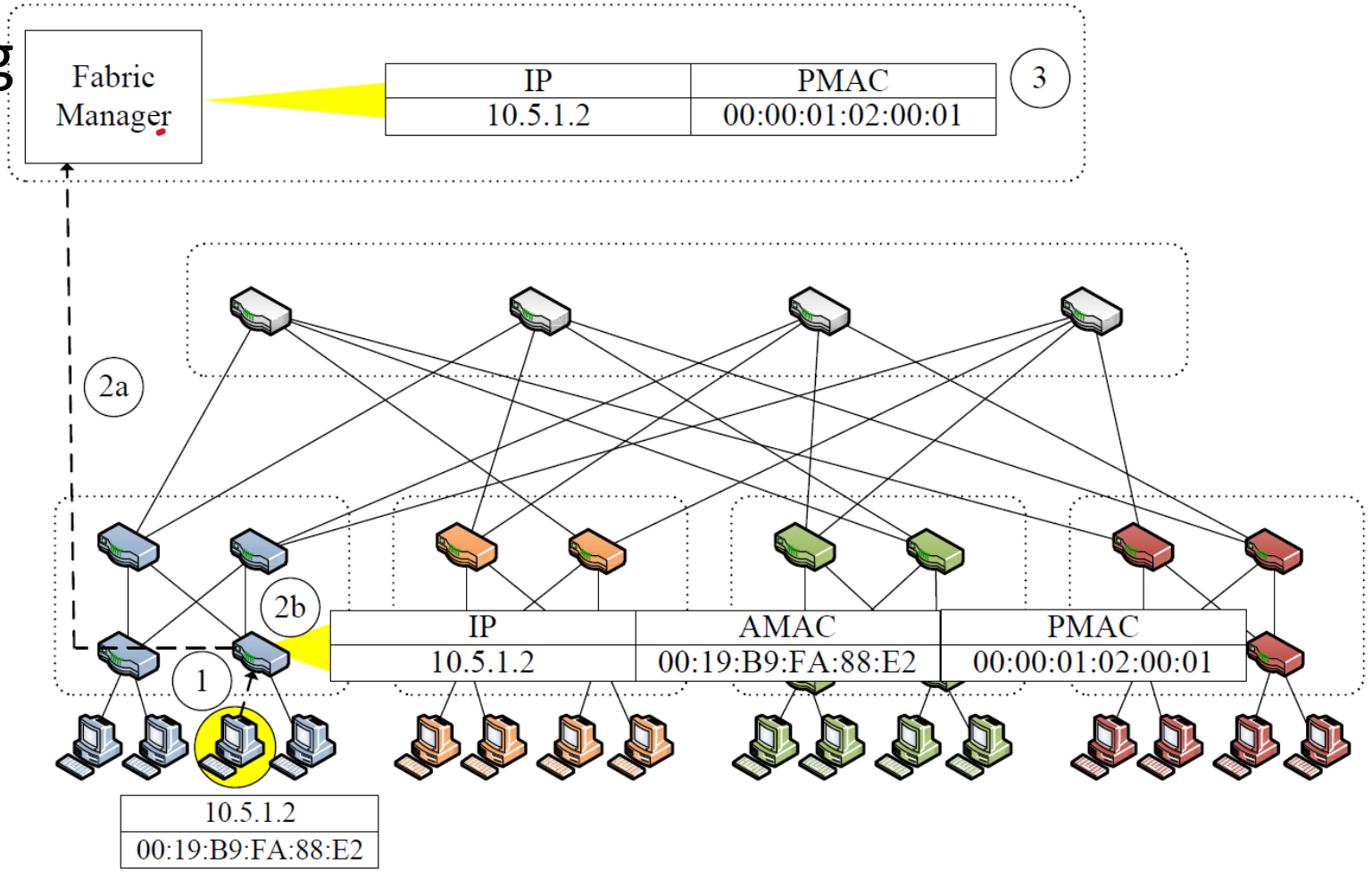
Total # of servers: k^3/4

pods: k

edge switches/pod: k/2

Total # of switches: 5k^2/4

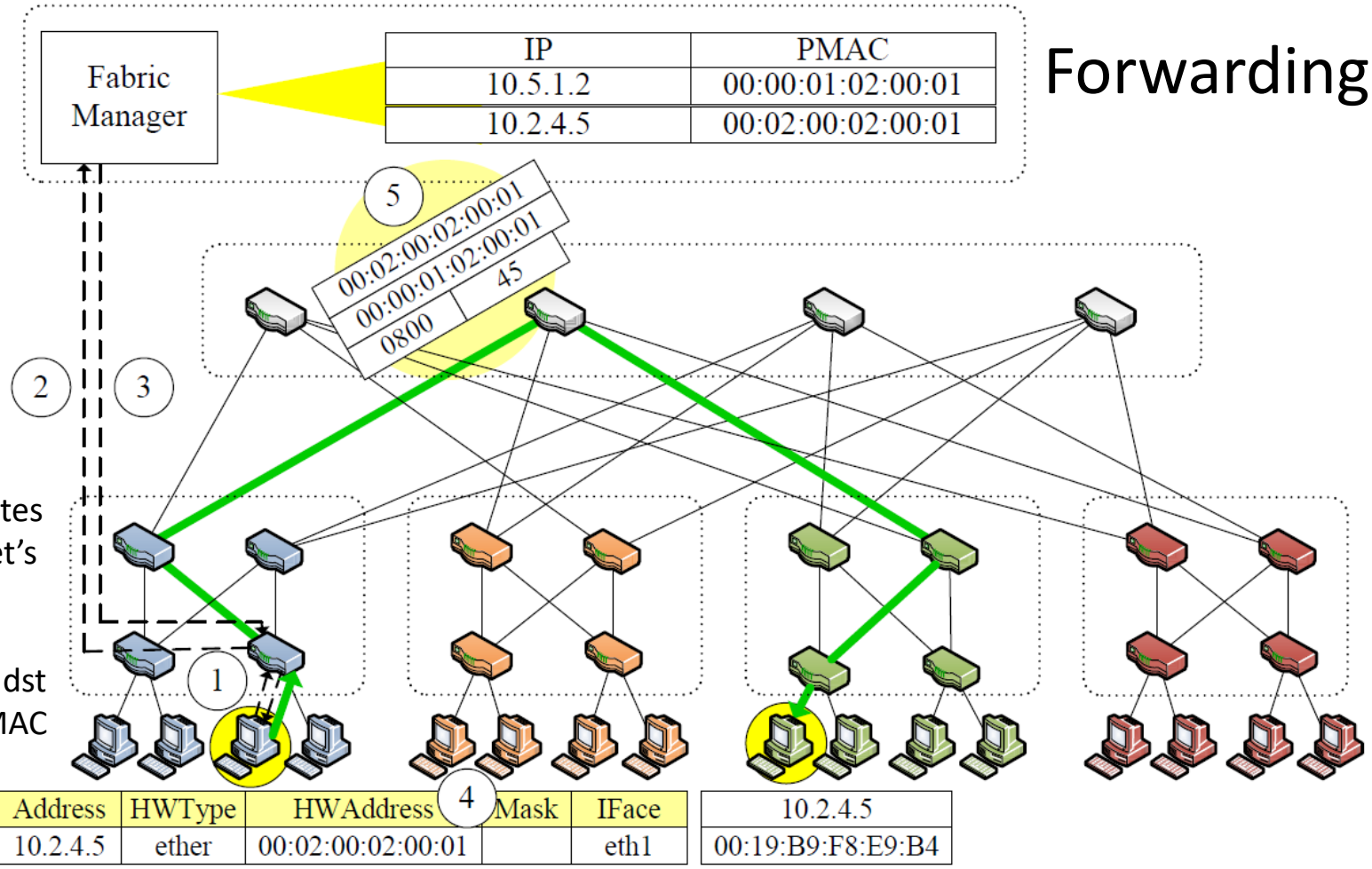
Addressing



- The actual MAC (AMAC) is completely flat, and thus hard for routing
- Portland introduces a pseudo MAC (PMAC) for each VM, with a structured format: **pod.position.port.vmid**
- Using the PMAC, an encoded endpoint location, a switch knows how to forward a packet to its destination
- E.g., a host with IP=10.5.1.2 has AMAC=00:19:B9:FA:88:E2 and PMAC=00:00:01:02:00:01 (16,8,8,16 bits)
- Registration: (1) A host sends its IP and AMAC to its edge switch, which then (2a) informs the Fabric Manager with IP/PMAC for ARP use, and (2b) records IP/AMAC/PMAC at its local table

Forwarding

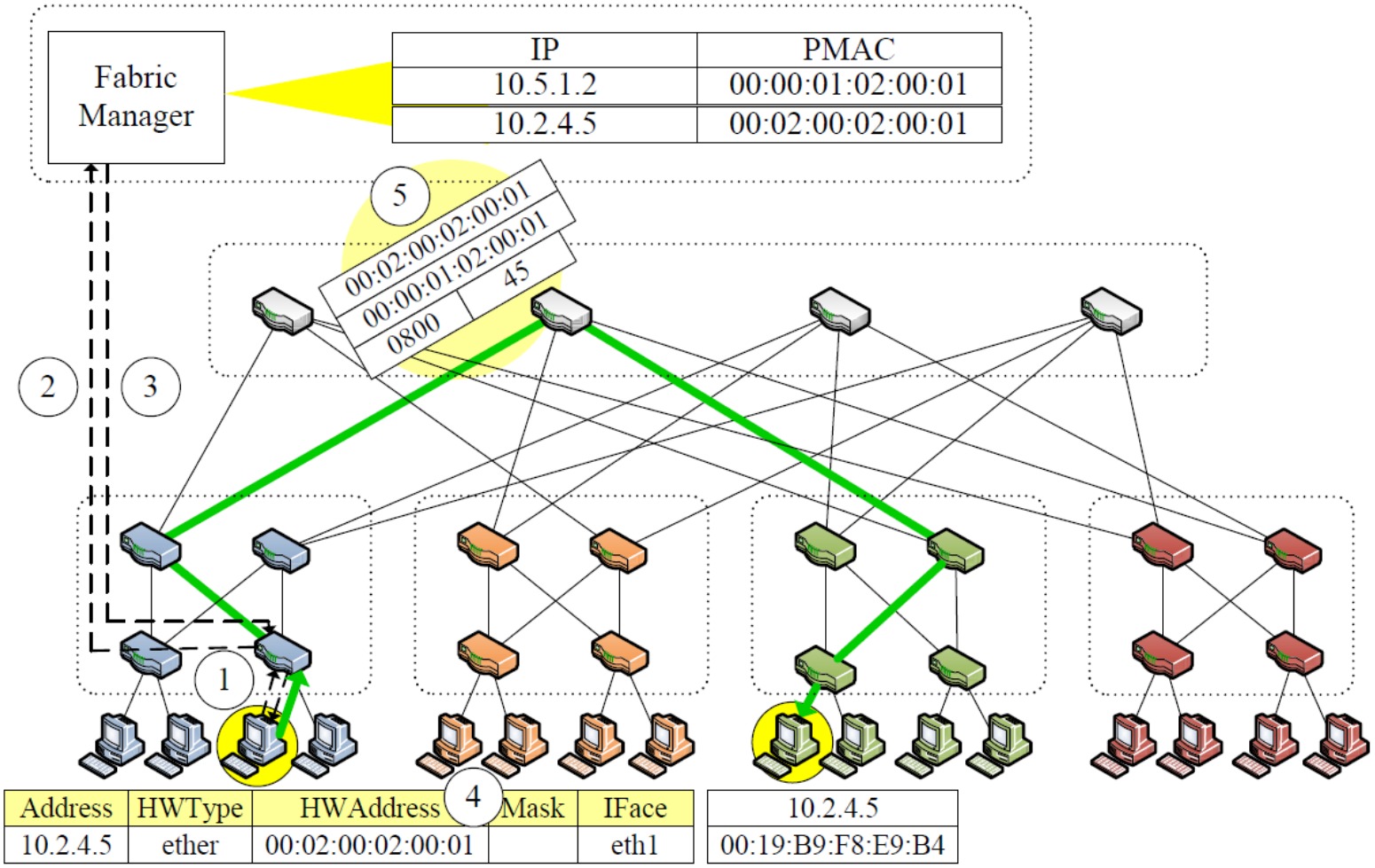
Fabric Manager servers as a proxy-ARP agent, and answers ARP queries



The edge sw rewrites (1) outgoing packet's src AMAC with src PMAC, and (2) incoming packet's dst PMAC with dst AMAC

1. A host sends an ARP to get dst PMAC of dst IP 10.2.4.5
2. The ARP query is captured by the edge switch and sent to the Fabric Manager
3. ARP-reply from the Fabric Manager carries the dst PMAC of dst IP 10.2.4.5
4. After receiving the ARP reply, the host sends a frame with src AMAC and dst PMAC to the src edge switch
5. The src edge switch rewrites the src AMAC with the src PMAC using the mapping of AMAC to PMAC, and sends the packet to an aggregate sw
6. The packet is then forwarded to the destination edge switch using dst PMAC, where dst PMAC is then replaced with dst AMAC using the mapping of PMAC to AMAC

Routing



Up: choose any port, independent of the destination address

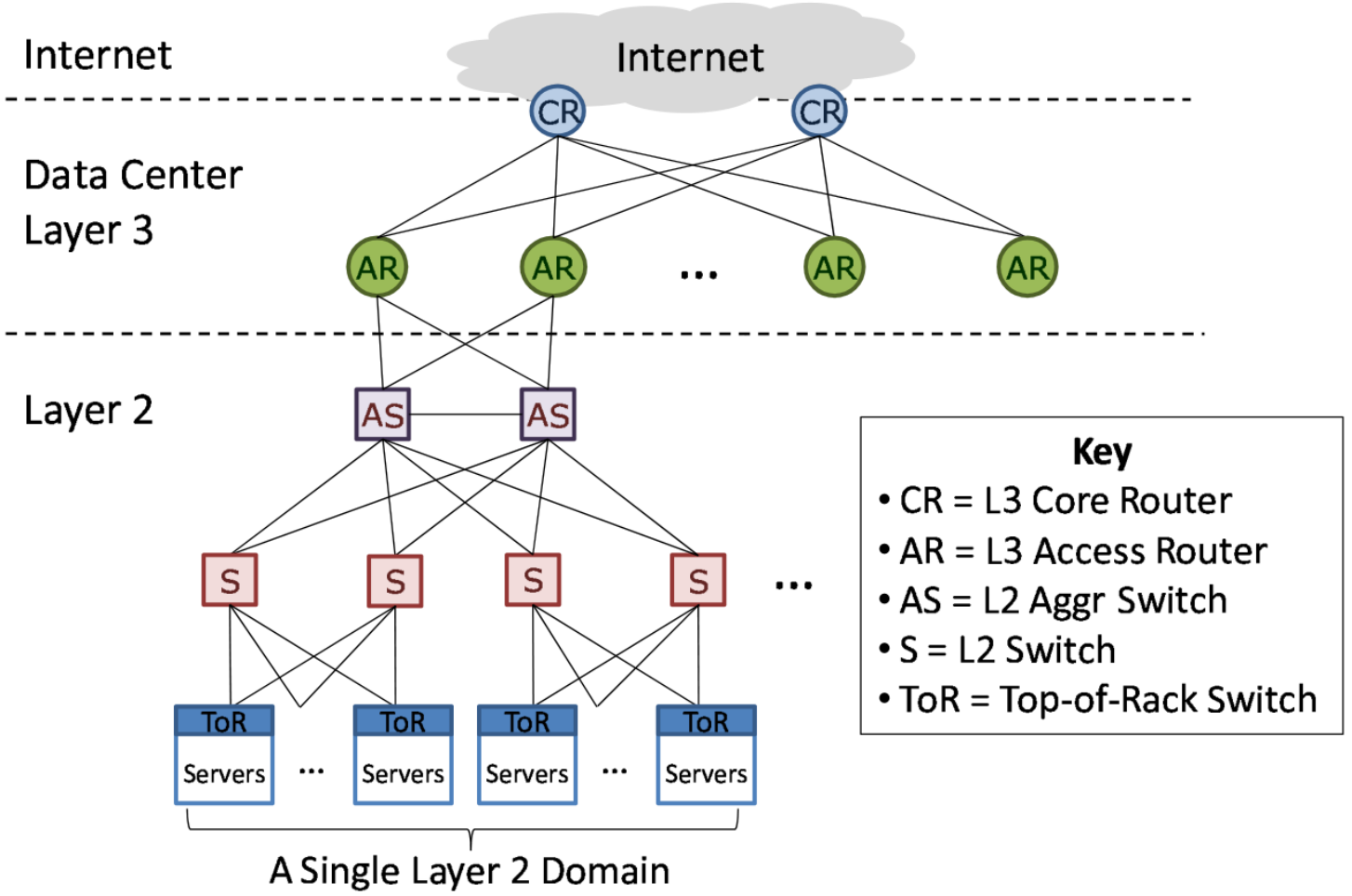
Down: only one path from a core to the destination

Portland Summary

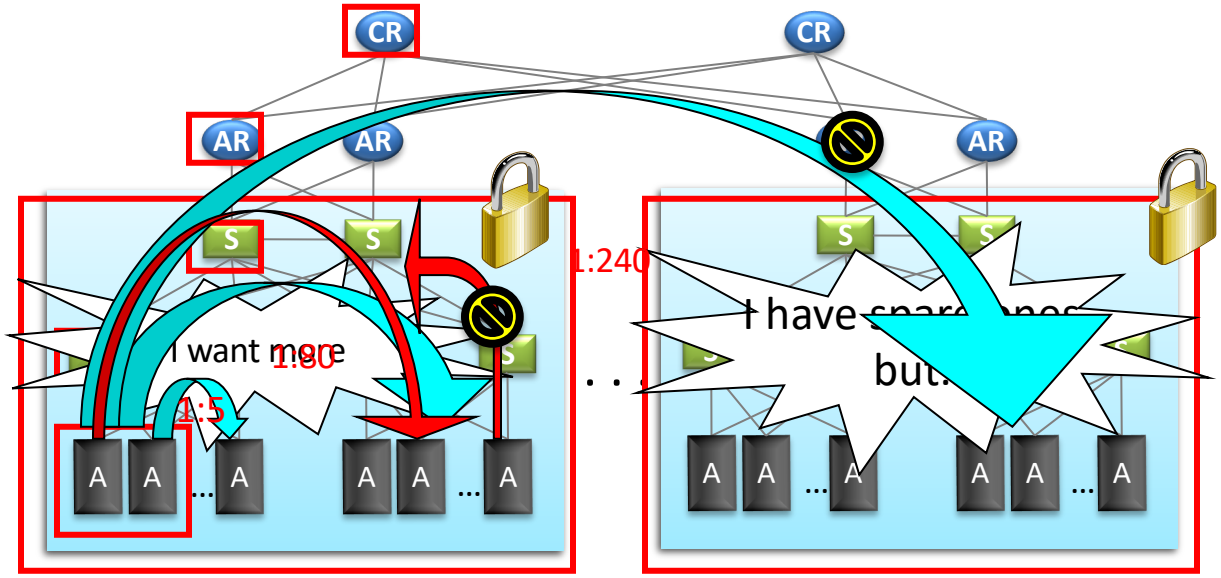
- How does Portland provide high capacity?
  - Topology
  - Routing
- How does Portland provide a flat address space?
  - PMAC
  - MAC rewriting

4. VL2 (Virtual Layer 2) DCN

Architecture of Data Center Networks (DCN)



Conventional DCN Problems



- Static network assignment
- Fragmentation of resource

- Poor server to server connectivity
- Traffics affects each other
- Poor reliability and utilization

Hakim Weatherspoon, High Performance Systems and Networking Lecture Slides
https://www.cs.cornell.edu/courses/cs5413/2014fa/lectures/09-vl2.pptx

VL2 (Virtual Layer 2) DCN

- Design philosophy 1: use commodity IP switches without any changes
- Design philosophy 2: build a flat L2 network
- Design philosophy 3: it is more feasible to modify computer software than modify switches/routers

- A. Greenberg, et al, "VL2: A Scalable and Flexible Data Center Network", Sigcomm 2009

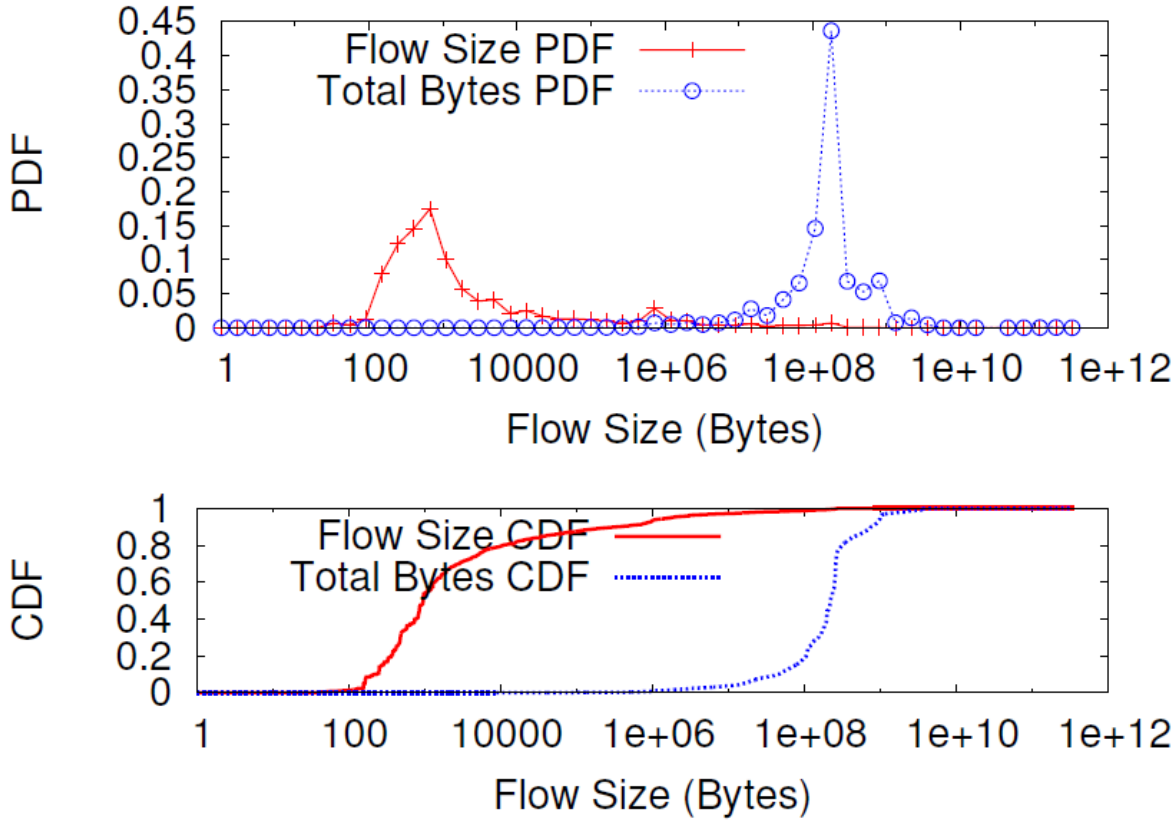
Design Objectives

- Uniform high capacity:
  - Maximum rate of server to server traffic flow should be limited only by capacity on network cards
  - Assigning servers to service should be independent of network topology
- Performance isolation:
  - Traffic of one service should not be affected by traffic of other services
- Layer-2 semantics:
  - Easily assign any server to any service
  - Configure server with whatever IP address the service expects
  - VM keeps the same IP address even after migration

Microsoft Data-Center Traffic Analysis

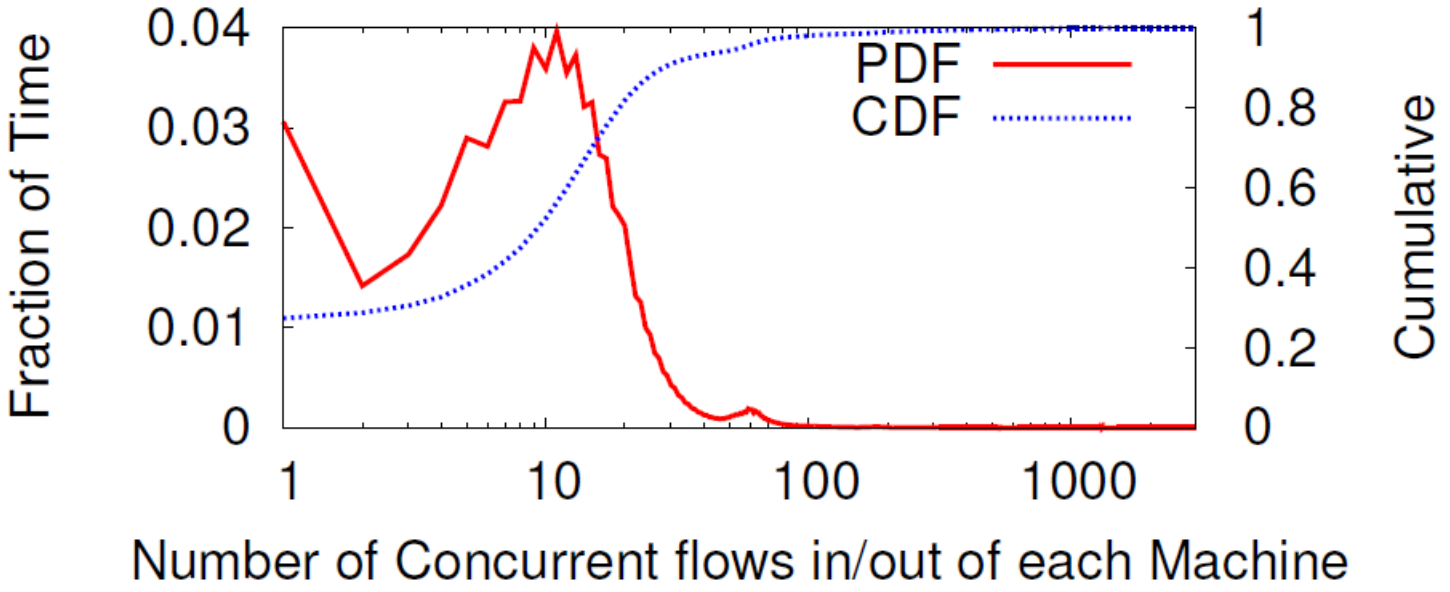
1. The ratio of traffic volume between servers in their data centers to traffic entering/leaving the data centers is currently around 4:1 (excluding CDN applications).
2. Data-center computation is focused where high speed access to data on memory or disk is fast and cheap.
3. The demand for bandwidth between servers inside a data center is growing faster than the demand for bandwidth to external hosts.
4. The network is a bottleneck to computation. ToR switch's uplinks are often above 80% utilization.
5. Pattern of networking equipment failures:
    - 95% of failures resolved within < 1min
    - 98% < 1hr
    - 99.6% < 1 day
    - 0.09% > 10 days

Flow Sizes Distribution



- Mice are numerous; 99% of flows are smaller than 100 MB. However, more than 90% of bytes are in flows between 100MB and 1 GB.
- "Total Bytes PDF" is a probability density function of each flow size contributing to the total byte count. For instance, 100MB flows contribute 45% of total number of bytes.

No. of Concurrent Flows of a Server

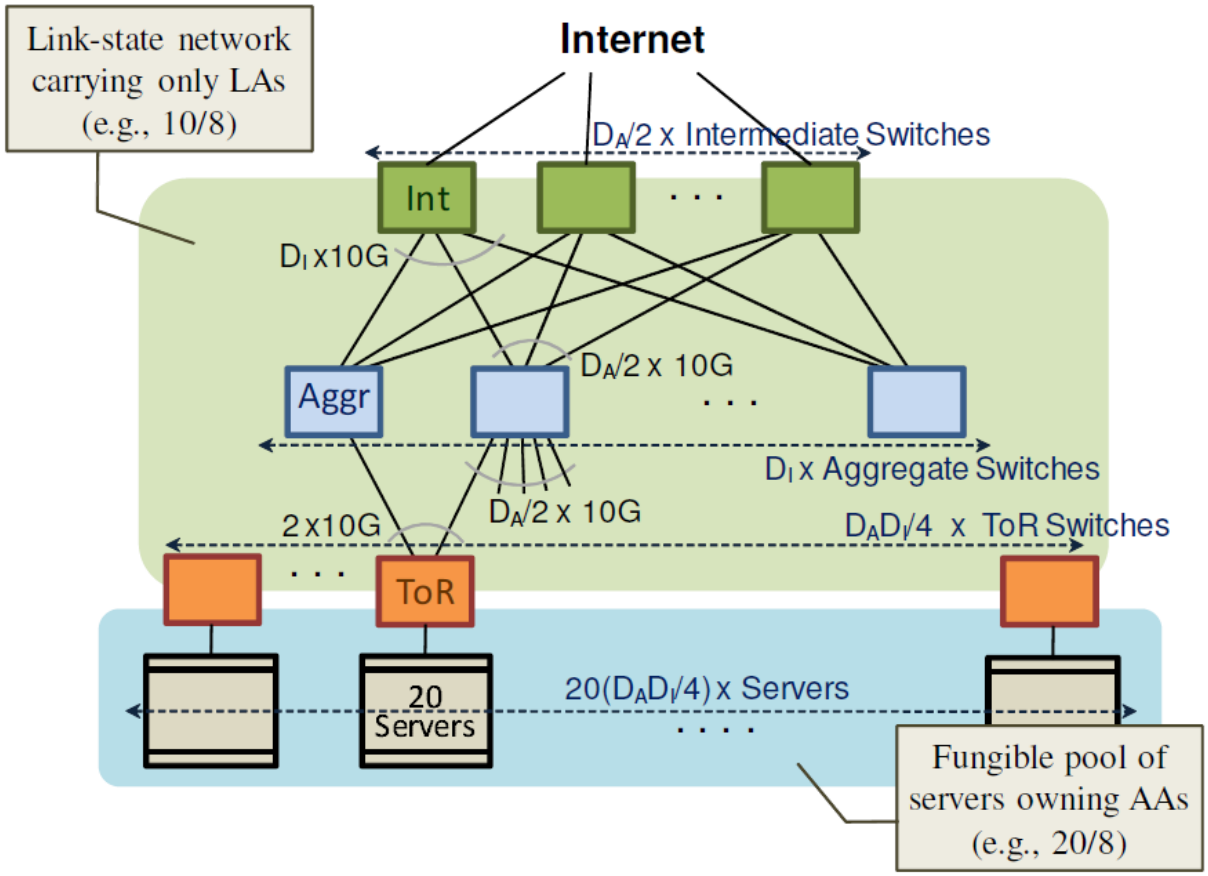


- Observed a total of 1,500 servers for a representative day
- More than 50% of the time, an average server has 10 concurrent flows or more
- At least 5% of the time, it has greater than 80 concurrent flows.
- Never observed more than 100 concurrent flows.

Design Principles of VL2

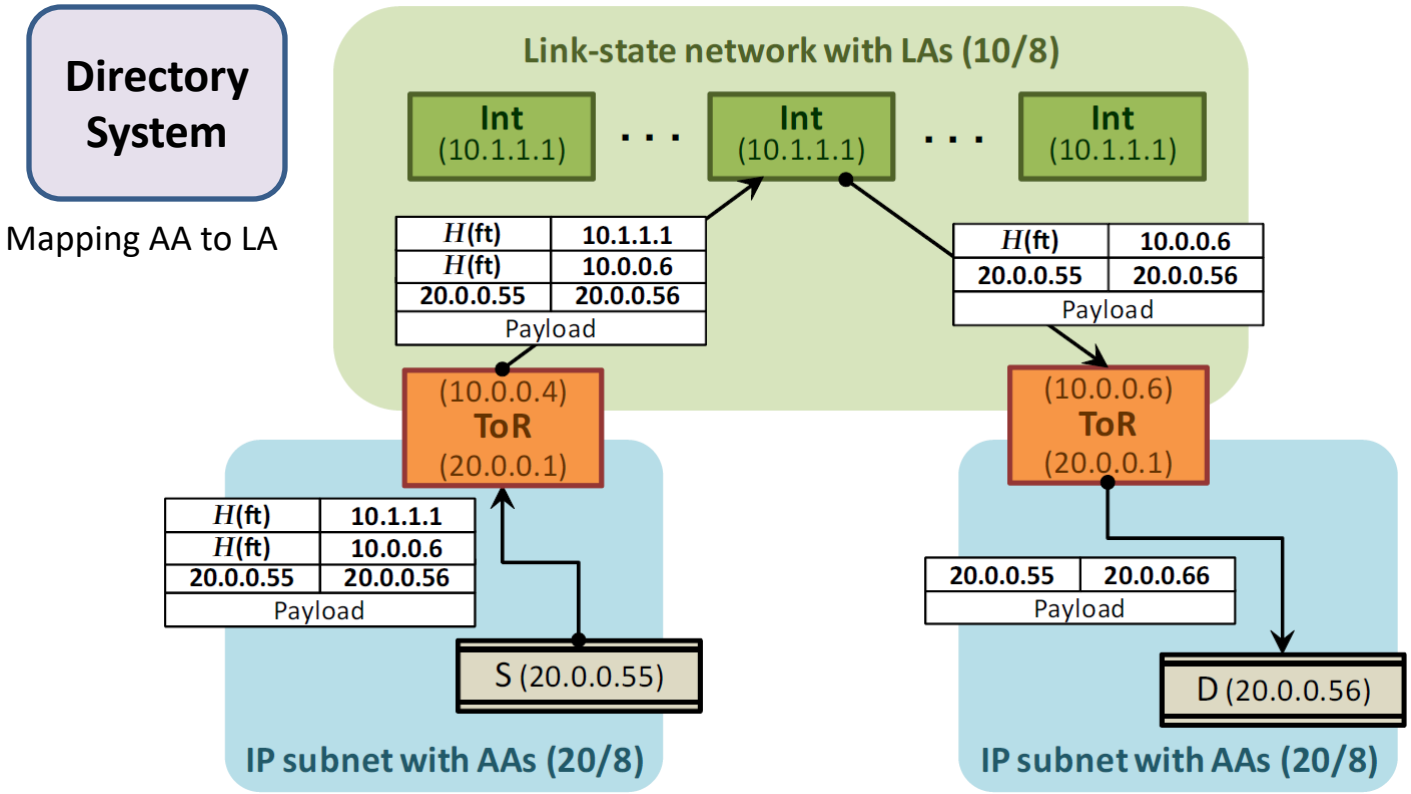
- Randomizing to cope with volatility:
  - Using Valiant Load Balancing (VLB) to do destination independent traffic spreading across multiple intermediate nodes
- Building on proven networking technology:
  - Using IP routing and forwarding technologies available in commodity switches
- Separating names from locators:
  - Using directory system to maintain the mapping between names and locations
- Embracing end systems:
  - A VL2 agent at each server

VL2 Network Architecture



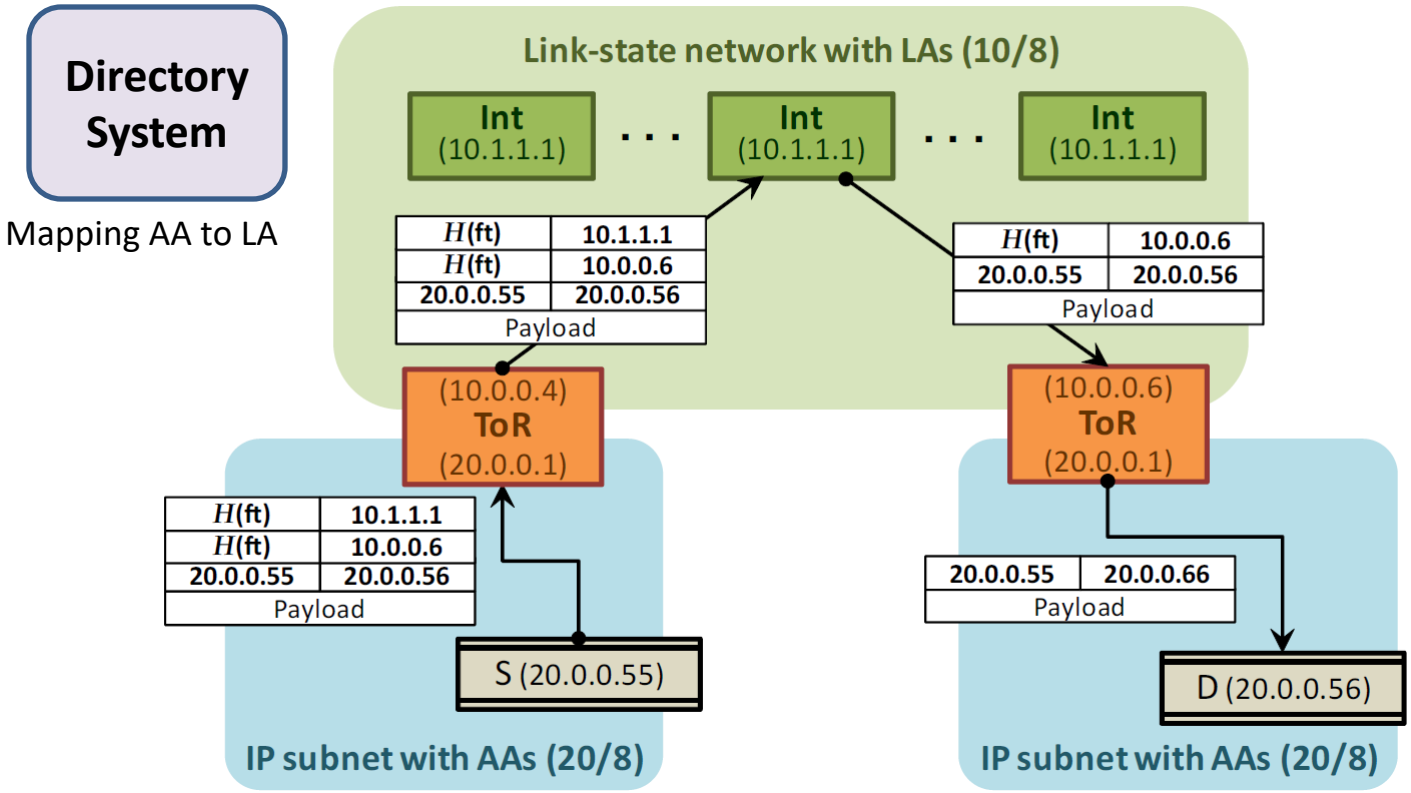
- Clos network: provide high bandwidth
- Use commodity IP switches: scale out
- Two kinds of address: application-specific addresses (AAs) in the blue domain and location-specific address (LAs) in the green domain

Packet Forwarding



- Sender **S** sends packets to destination **D** via a randomly-chosen intermediate switch using IP-in-IP encapsulation. AAs are from 20/8, and LAs are from 10/8. There are 3 layers of encapsulation.
- To route traffic between servers, which use AA addresses, on an underlying network that knows routes for LA addresses, the VL2 agent at each server traps packets from the host and encapsulates the packet with the LA address of the ToR of the destination.
- Once the packet arrives at the LA (the destination ToR), the switch decapsulates the packet and delivers it to the destination AA carried in the inner header.

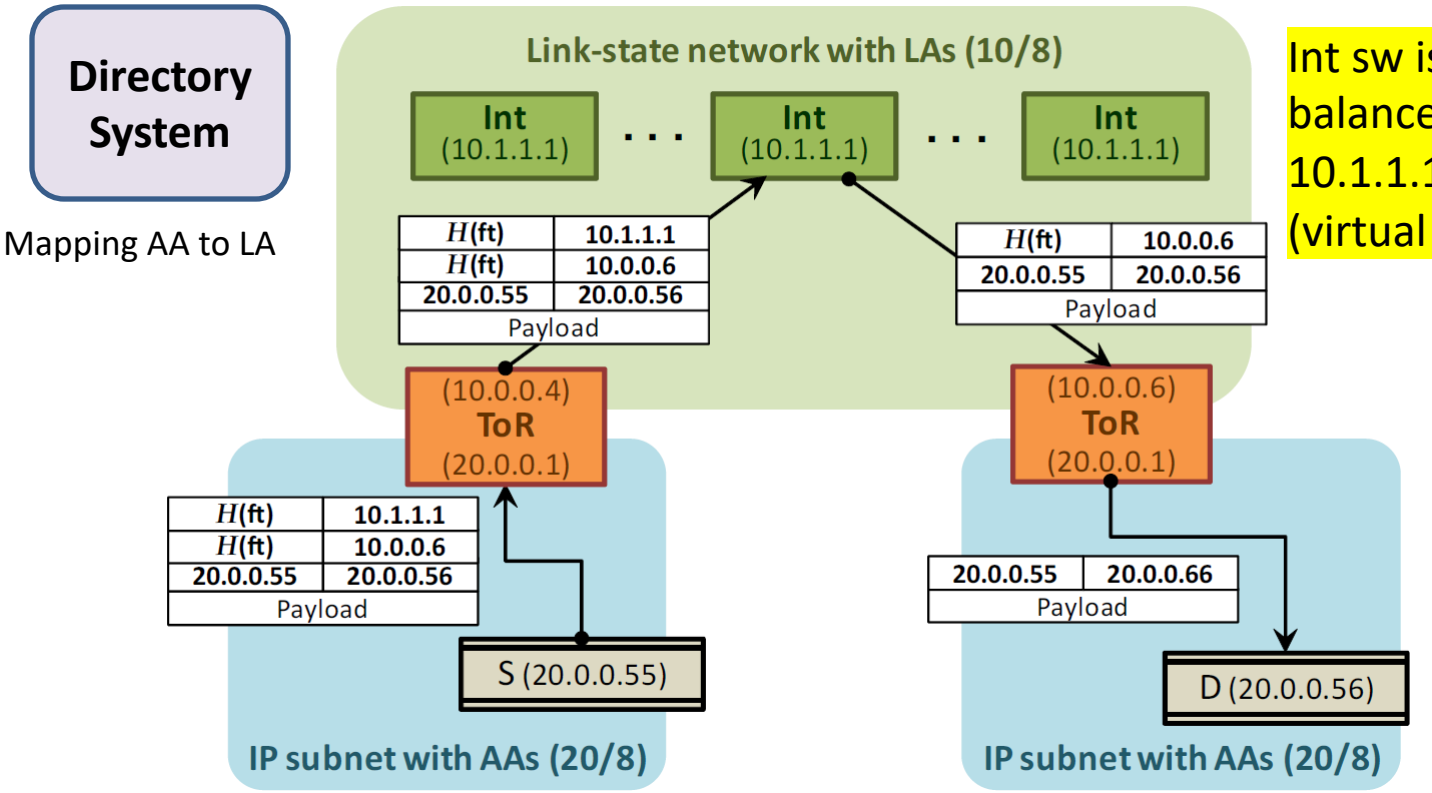
Address Resolution



Servers in each service are configured to believe that they all belong to the same IP subnet. Hence, when an application sends a packet to an AA for the first time, the networking stack on the host generates a broadcast ARP request for the destination AA. The VL2 agent running on the host intercepts this ARP request and converts it to a unicast query to the VL2 directory system. The directory system answers the query with the LA of the ToR to which packets should be tunneled. The VL2 agent caches this mapping from AA to LA addresses, similar to a host's ARP cache, such that subsequent communication need not entail a directory lookup.

Access control via the directory service: A server cannot send packets to an AA if the directory service refuses to provide it with an LA through which it can route its packets.

VL2 Load Balancing

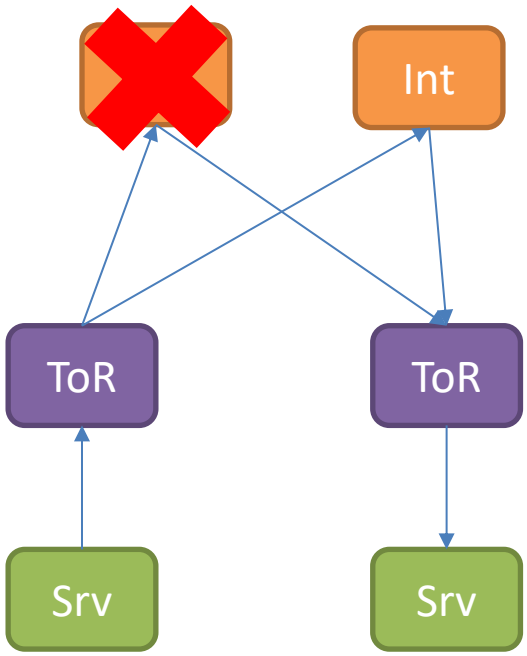


Int sw is like a load balancer, and 10.1.1.1 is like a VIP (virtual IP address).

- VL2 uses ECMP (Equal-Cost MultiPath) to perform load balancing to achieve better link utilization
- ECMP is readily supported by commodity IP routers. H(ft) denotes a hash of the five tuples and is placed at the source IP address field.
- This design and FabricPath share the same principle: the core network has semi-static routing tables. But what are the major differences?

VL2 Load Balancing

- Valiant Load Balancing (VLB) for random route
  - When going up, choose random node
- Equal Cost Multi Path Forwarding (ECMP) for fail-safe
  - Assign the same LA to multiple node
  - When a node fail, use IP anycast to route to backup node
- TCP for congestion control



VL2 Summary

- How does VL2 provide high capacity and throughput?
  - Topology: 3 or 5 stages of Clos network
  - Routing: ECMP
- How does VL2 provide a flat address space?
  - Use two kinds of IP addresses (AA and LA) and encapsulation of them