CSC 525:
Computer Networks

Cloud Computing

- Popular services are hosted in the cloud
  - Web search, email, social networks, data mining, etc.
  - More cost effective due to resource pooling.
- The cloud is the infrastructure
  - Storage, computing, bandwidth, multiple locations.
- The network of global services:
  - Internet: connect many ISPs
  - Front End: a large number of geographically distributed servers. Redirect requests, cache results, etc.
  - Back End: large data centers. Does the heavy lifting of the computation and services.

Data Center Networks (DCN)

- Power requirement and economy of scale determine data center size: usually 50K-200K servers
- Elastic services by virtual machine management
  - Servers divided up among hundreds of different services.
  - Scale-out is paramount: some services have 10s of servers, some have 10s of 1000s
  - Need to constantly create/destroy/migrate VMs.
- Some network characteristics
  - Failures are the norm.
  - Majority of the traffic stay within the data center.
  - Low latency
  - Traffic matrix is volatile.

A Big LAN

- Two high level choices for Interconnections within DCN:
  - Specialized hardware and communication protocols (Infiniband, Myrinet)
    - Very expensive; Not natively compatible with TCP/IP applications
  - Commodity Ethernet switches and routers
    - Compatible and Cheap
- Current trend
  - Topology – Typical architectures today consist of either two- or three level trees of switches or routers as FAT-trees
  - Routing / Switching Devices– Cheap off-the-shelf Ethernet switches are increasingly becoming the basis for large scale Data Center networks
  - Routing – Most enterprise core switches support Equal Cost Multi Path routing

Motivation

- Requirements for scalable, easily manageable, fault tolerant and efficient Data Center Networks (DCN):
  - **R1:** Any VM may migrate to any physical machine without changing their IP addresses
  - **R2:** An administrator should not need to configure any switch before deployment
  - **R3:** Any end host should efficiently communicate with any other end hosts through any available paths
  - **R4:** No forwarding loops
  - **R5:** Failure detection should be rapid and efficient

- Implication on network protocols:
  - A single layer2 fabric for entire data center (R1&R2)
  - MAC forwarding tables with hundreds of thousands entries (R3)
  - Efficient routing protocols which disseminate topology changes quickly to all points (R4 & R5)

Network Design Choices

- Desired properties for scalable, easily manageable, fault tolerant and efficient Data Center Networks (DCN):
  1. Any VM may migrate to any physical machine without changing their IP addresses
  2. An administrator should not need to configure any switch before deployment
  3. Any end host should efficiently communicate with any other end hosts through any available paths
  4. Small switching/routing table size
  5. No forwarding loops
  6. Failure detection should be rapid and efficient

- Layer 2 or layer 3?
  - Layer 2 (Ethernet switching) for 1 & 2.
  - Layer 3 (IP routing) for 3 & 4.

Network Design Choices

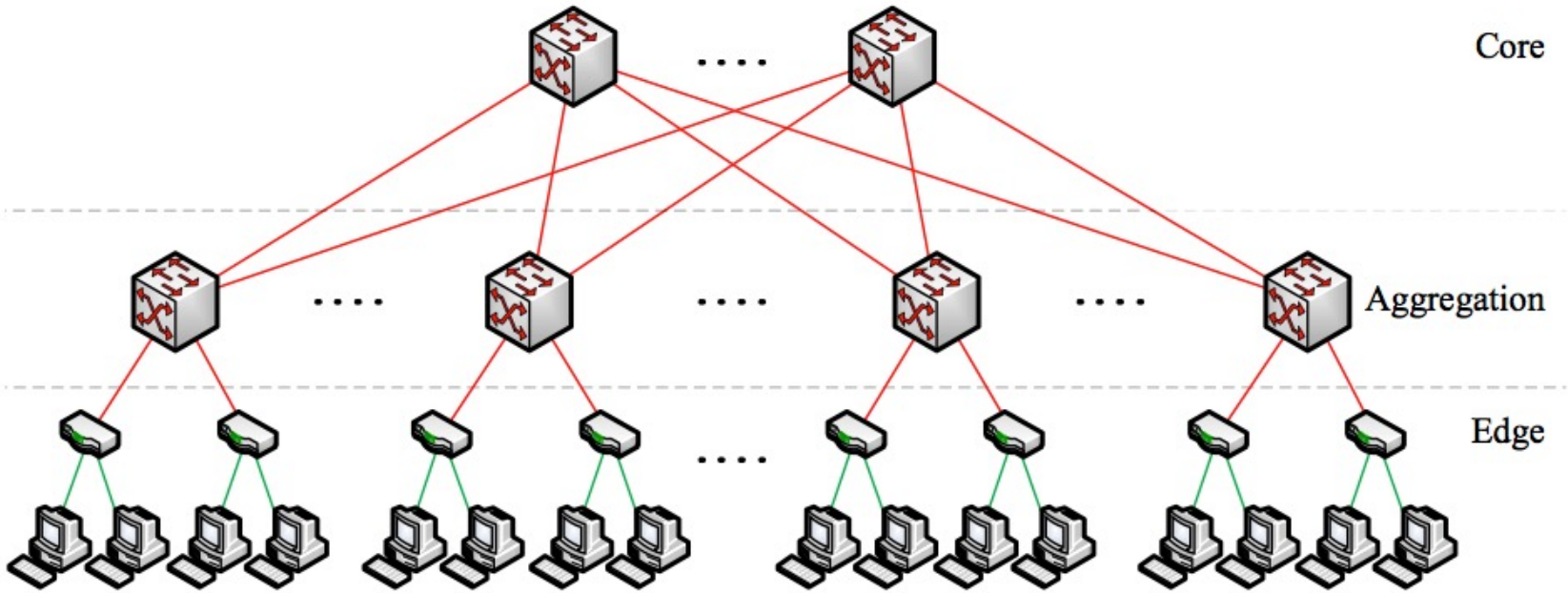
- Layer 3 approach:
  - Assign IP addresses to hosts hierarchically based on their directly connected switch.
  - Use standard intra-domain routing protocols, e.g. OSPF.
  - Large administration overhead, e.g., network configuration.
  - Need to change IP address when migrating VMs.

- Layer 2 approach has many advantages:
  - Forwarding on flat MAC addresses
  - No need to change address when moving VMs.
  - Plug and play, minimal admin overhead if any.
  - Traditional Ethernet cannot utilize all links in the network, and may have large switch table size.

Background

- DCN topology:
  - End hosts connects to top of rack (ToR) switches
  - ToR switches contains 48 GigE ports and up to 4 10GigE uplinks
  - ToR switches connect to one or more end of row (EoR) switches

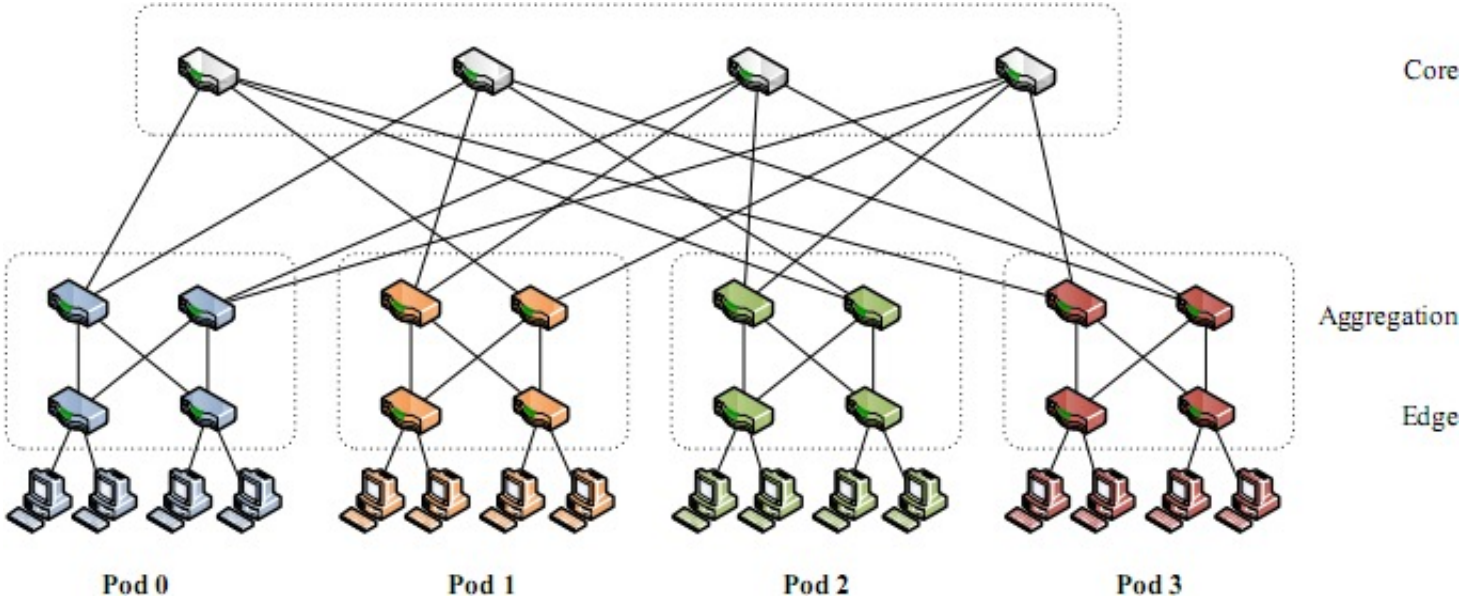
Conventional DCN Topology



- Hosts to switch links are GigE, links between switches are 10GigE.

Fat Tree Topology

- Fat Tree Networks:



- There're totally k pods. Each pod contains 2 layers of k/2 k-port switches; Each switch in the lower layer directly connects to k/2 hosts; Remaining k/2 ports connect to k/2 of the aggregation switches.
- Total $k^2/4$ core switches, each core has one port connecting to each pod.
- Each pod supports **non-blocking** operation among $k^2/4$ hosts. Each source and destination has $k^2/4$ paths.
- Total $k^3/4$ hosts, re-arrangeably non-blocking.

PortLand Design

- Goal:
  - Scalable layer-2 routing, forwarding and addressing for DCN
  - The entire network is a single IP subnet
  - Assuming Fat Tree network topology

- Design:
  - Fabric manager
  - Positional pseudo MAC addresses
  - Proxy-based ARP
  - Distributed Location Discovery
  - Loop free forwarding
  - Fault tolerant routing

Fabric Manager

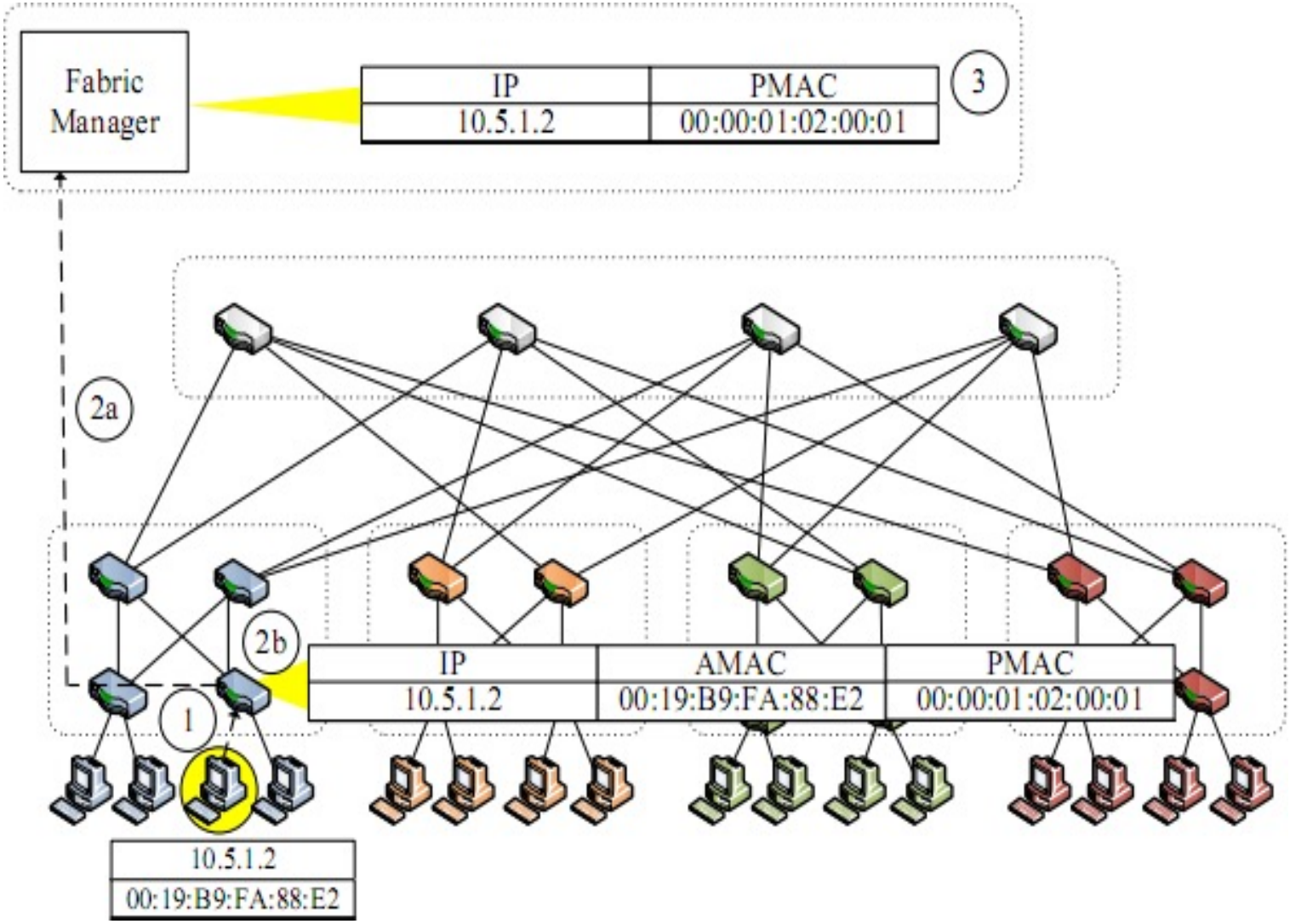
- Characteristics:
  - Logically centralized user process running on a dedicated machine
  - Maintains soft state about network configuration information
  - Responsible for assisting with ARP resolution, fault tolerance and multicast

- Why centralized?
  - Reduce the cost of configuration and management.

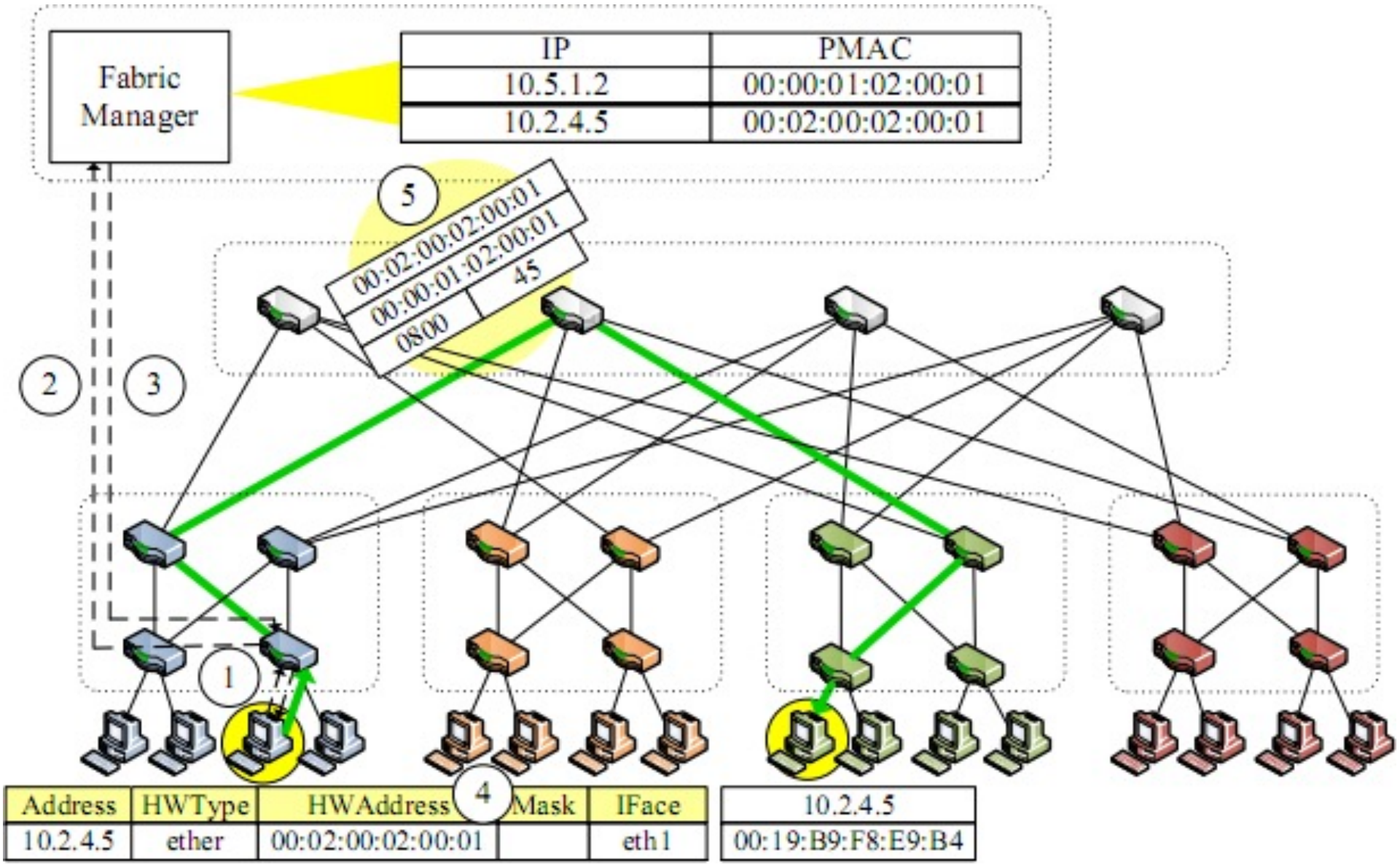
Positional Pseudo MAC Addresses

- Pseudo MAC (PMAC) address encodes the location of the host
  - 48-bit: pod.position.port.vmid
  - Pod (16 bit): pod number of the edge switch
  - Position (8 bit): position in the pod
  - Port (8 bit): the port number it connects to
  - Vmid (16 bit): VM id of the host

AMAC to PMAC mapping



Proxy-based ARP



Distributed Location Discovery

- Switches periodically send Location Discovery Message (LDM) out all of their ports to set their positions and to monitor liveness
- LDM contains: switch identifier, pod number, position, tree level, up/down

- Find position number for edge switch:
  - Edge switch randomly proposes a value in [0, k/2-1] to all aggregation switches in the same pod
  - If it is verified as unused and not tentatively reserved, the proposal is finalized.

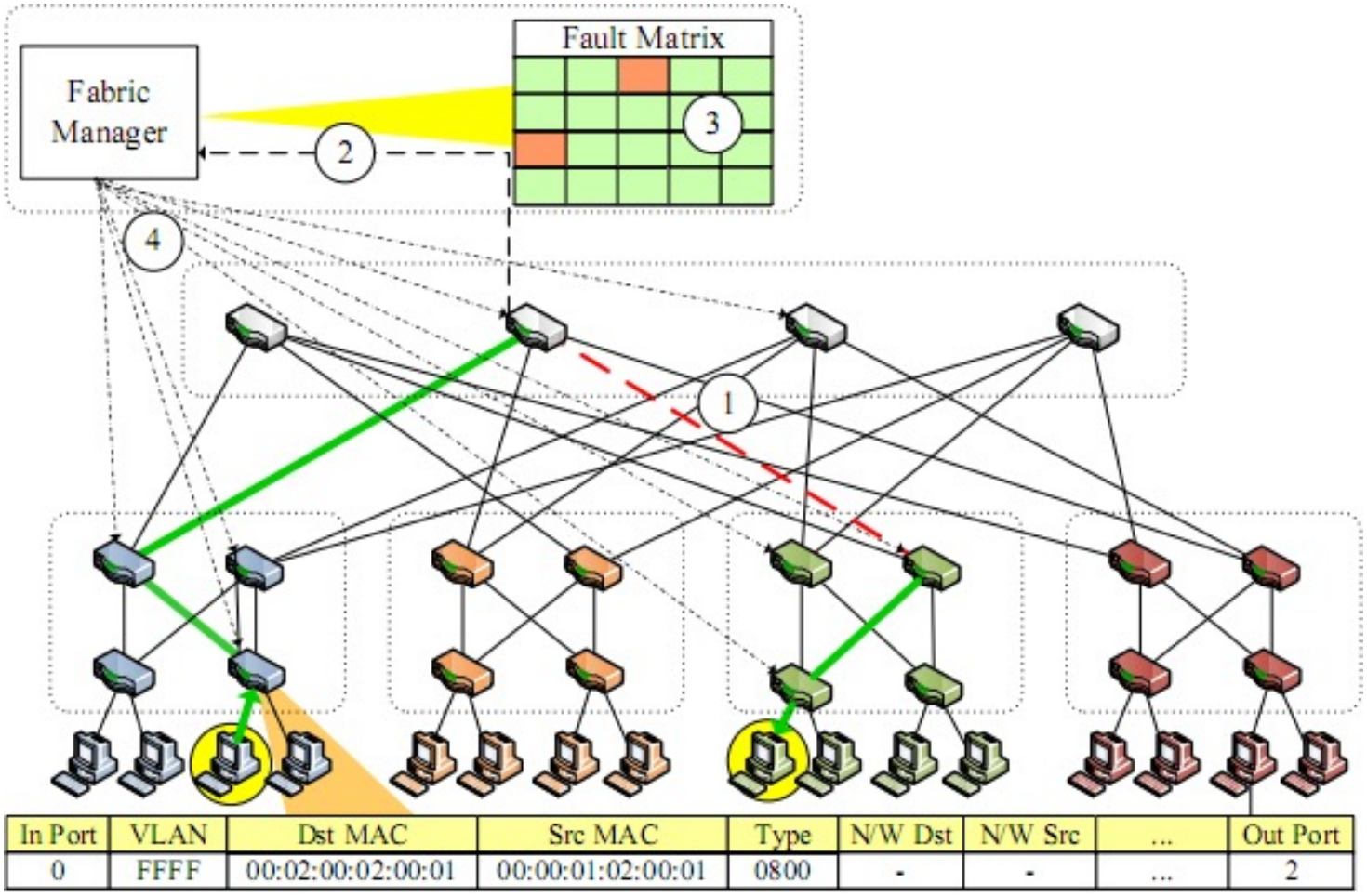
- Find tree level and up/down state:
  - Port states: disconnected, connected to end host, connected to another switch
  - A switch with at least half of ports connected to end hosts is an edge switch, and it infers on subsequent LDM that the corresponding incoming port is upward facing .
  - A switch getting LDM from edge switch is aggregation switch and corresponding incoming port is downward facing port.
  - A switch with all ports connecting to aggregation switch is core switch, all ports are downward.

Loop Free Forwarding

- Forwarding based on PMAC
  - pod.position.port.vmid
  - Core switches get pod value from PMAC, and send to corresponding port
    - Core switches learn the pod number of directly-connected aggregation switches
  - Aggregation switches get the pod and position value, if in the same pod, send to the port correspond to the position value, if not, send to the core switch
    - Aggregation switches learn the position number of all directly connected edge switches
    - Pod numbers are assigned by the fabric manager.

Fault Tolerant Routing

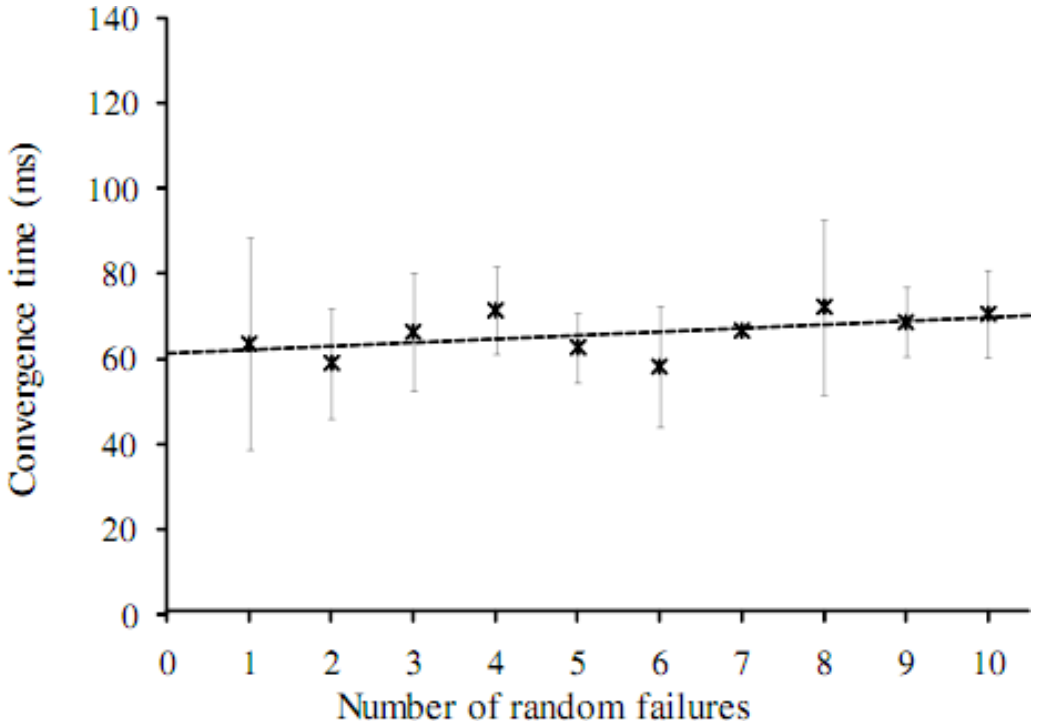
- Unicast: fault detection and action



In Port	VLAN	Dst MAC	Src MAC	Type	N/W Dst	N/W Src	...	Out Port
0	FFFF	00:02:00:02:00:01	00:00:01:02:00:01	0800	-	-	...	2

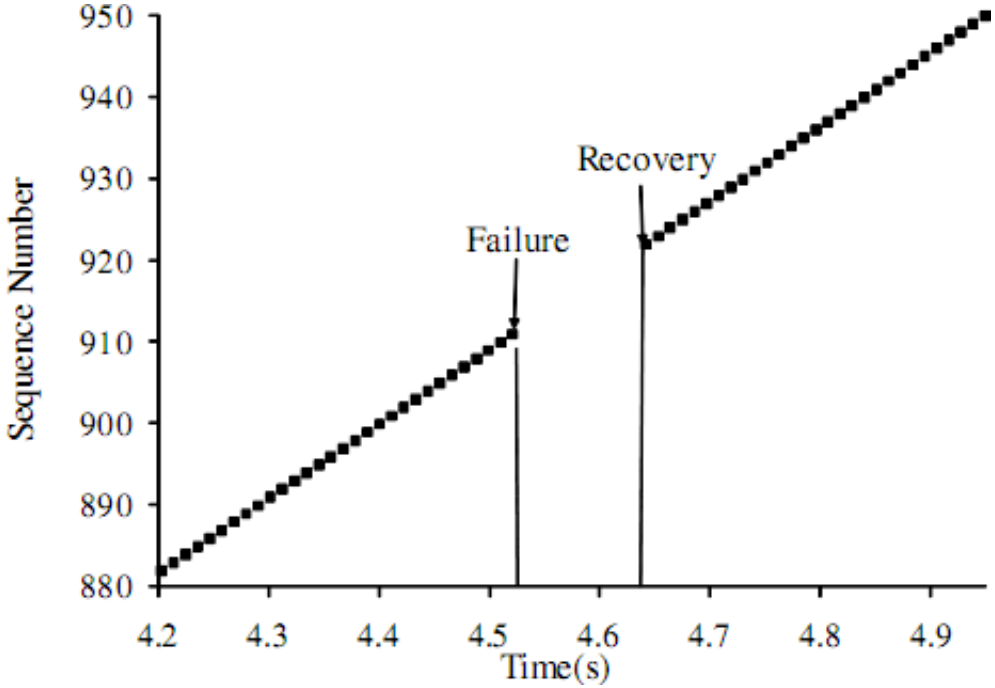
Implementation

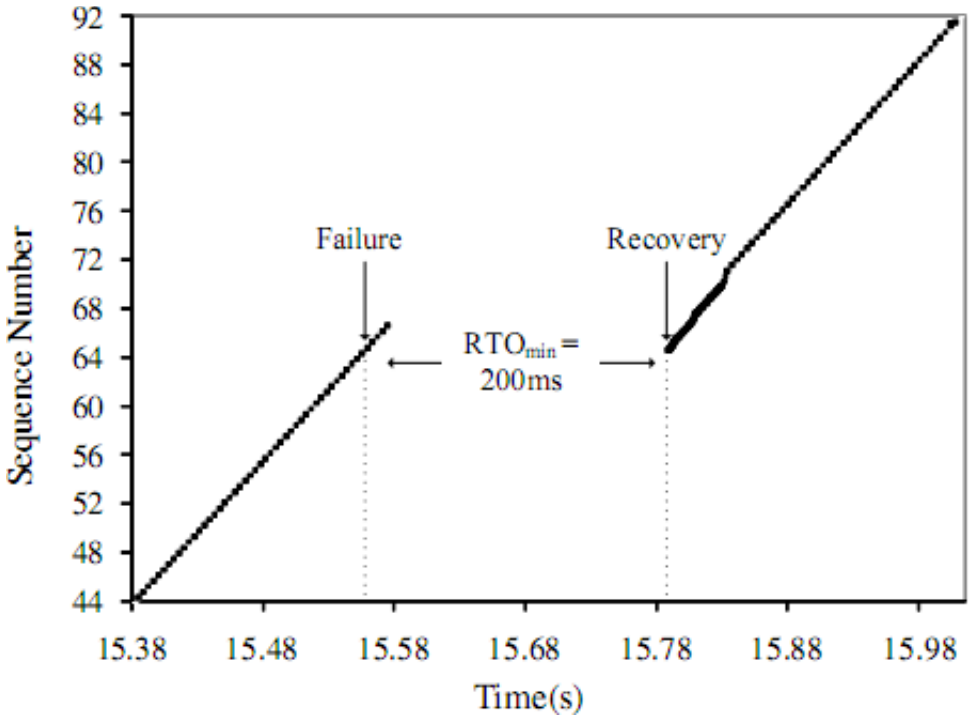
- Testbed:
  - 20 4-port NetFPGA PCI card switches, 16 end hosts
- Convergence time with increasing faults (UDP):
  - Transmit packets at rate 250Mpbs



Evaluation

- TCP convergence
  - Same experiment

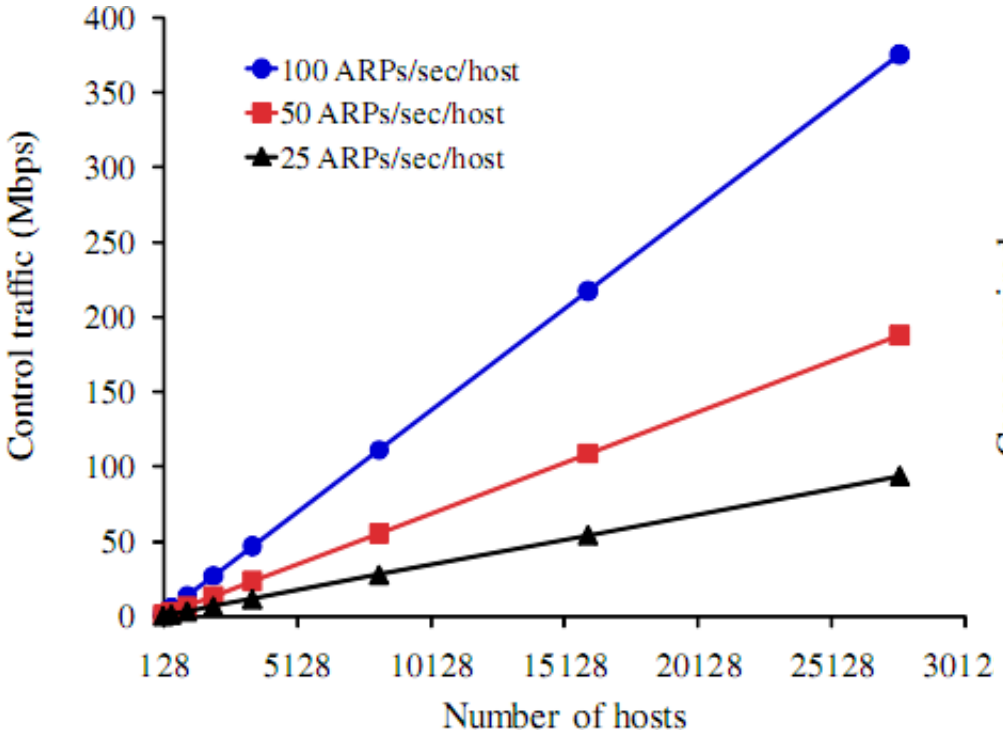


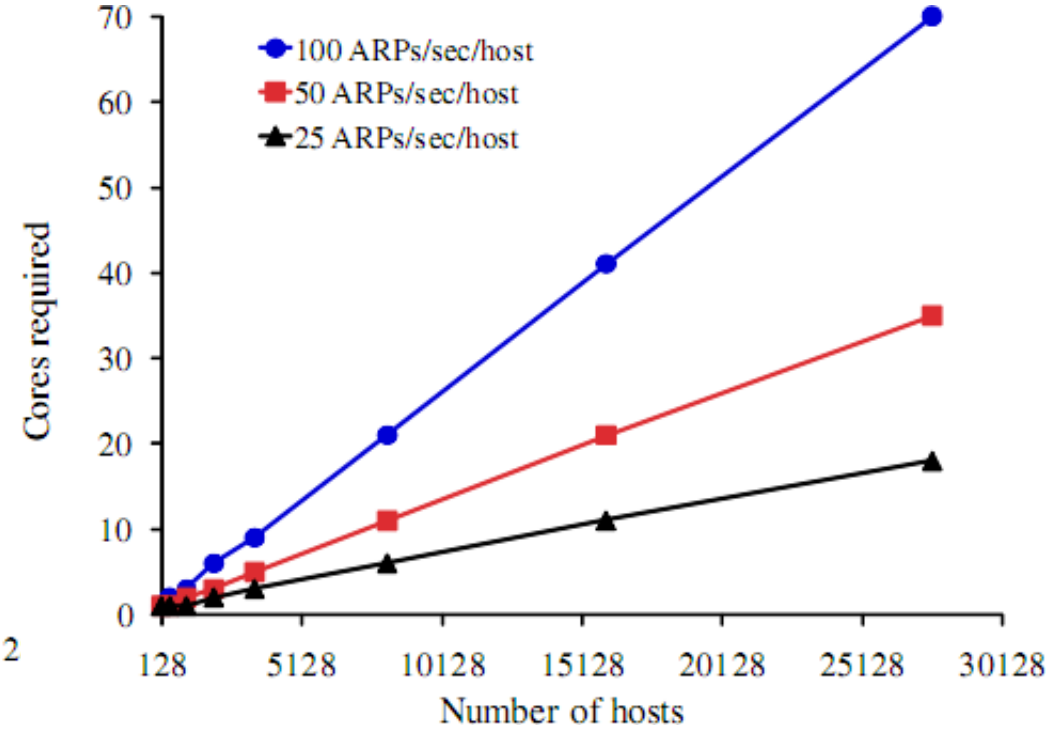


- Multicast convergence
  - Inject 2 link failure at time 4.5, 110 ms to converge

Evaluation

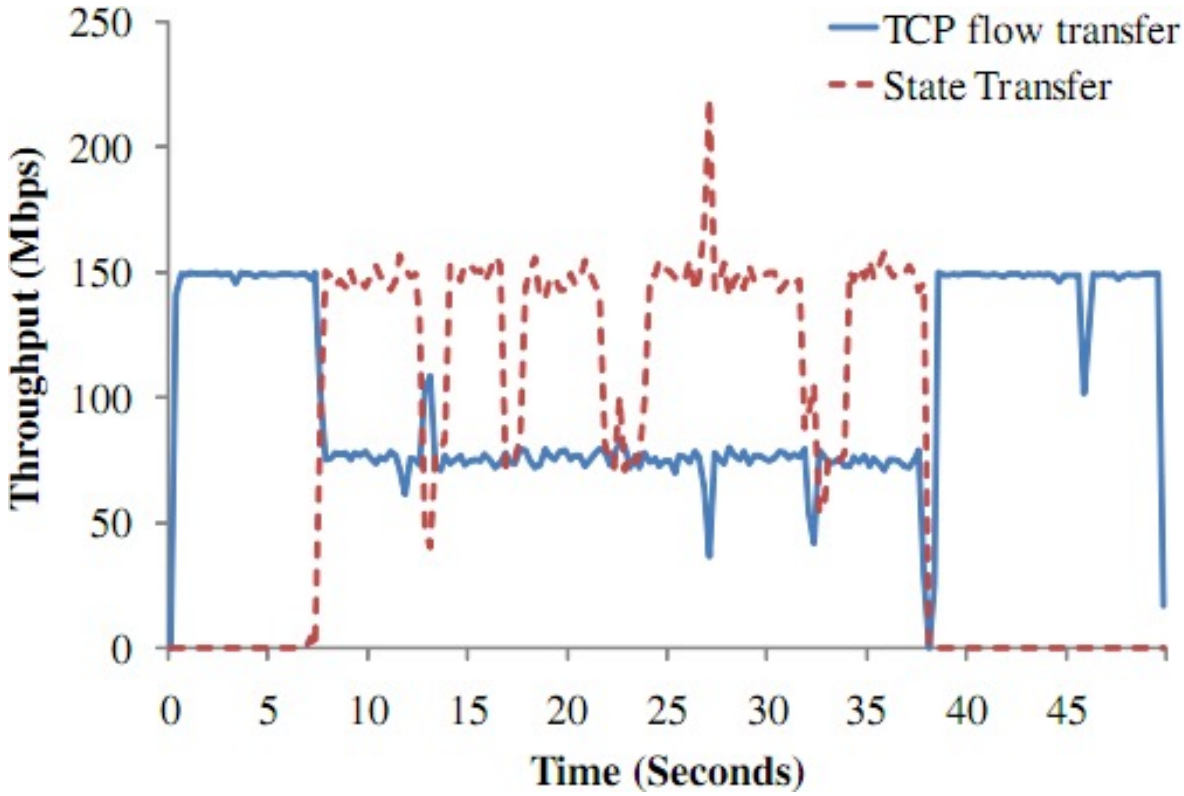
- Scalability
  - Each host transmits 25, 50, 100 ARP requests/sec to fabric manager





Evaluation

- VM migration
  - Sender transmits data at 150 Mbps to a VM
  - VM migrated to a host on another pod



Comparison

System	Topology	Forwarding		Routing	ARP	Loops	Multicast
		Switch State	Addressing				
TRILL	General	*O(number of global hosts)*	Flat; MAC-in-MAC encapsulation	Switch broadcast	All switches map MAC address to remote switch	TRILL header with TTL	ISIS extensions based on MOSPF
SEATTLE	General	*O(number of global hosts)*	Flat	Switch broadcast	One-hop DHT	Unicast loops possible	New construct: groups
PortLand	Multi-rooted tree	*O(number of local ports)*	Hierarchical	Location Discovery Protocol; Fabric manager for faults	Fabric manager	Provably loop free; no additional header	Broadcast-free routing; multi-rooted spanning trees

Summary

- Authors proposed PortLand, a scalable, fault tolerant layer 2 routing and forwarding protocol for DCN
- It is based on fat tree network topology.
- It uses PMAC to encode the location of the end host.
- It uses a centralized manager for ARP, failure recovery, etc.