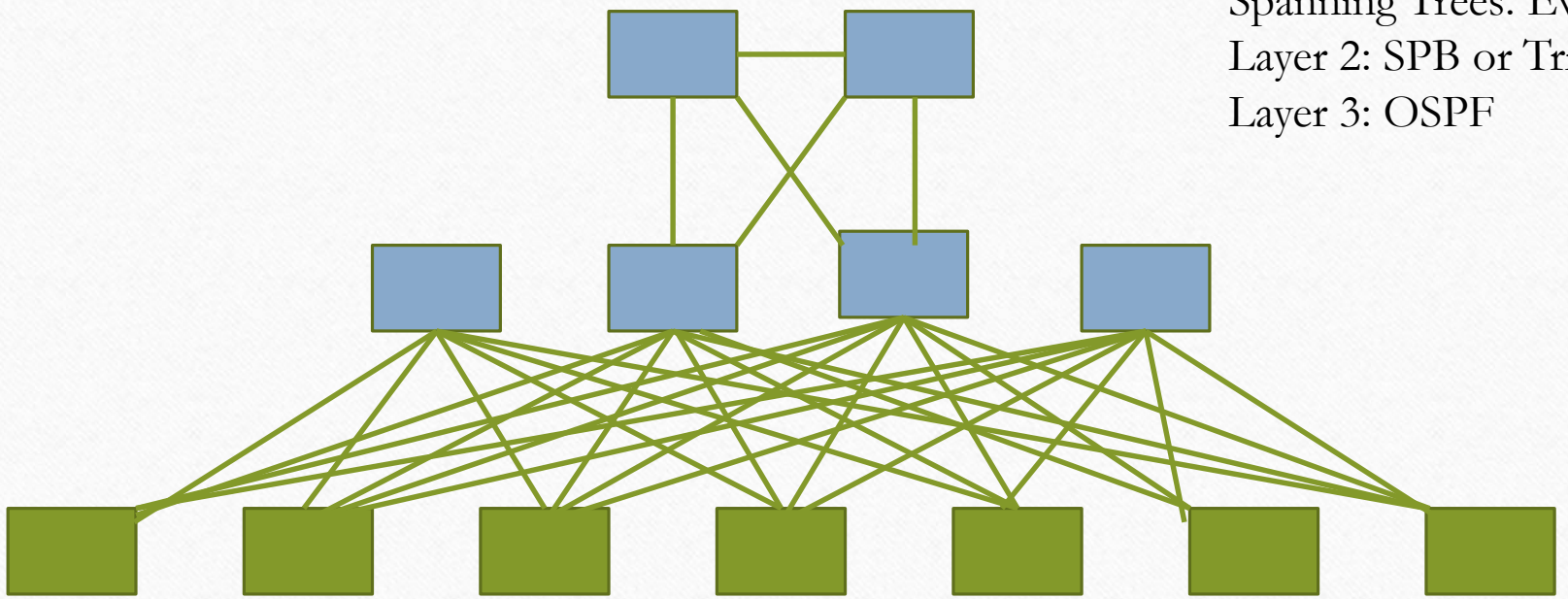
Cloud Computing

Lecture 2: Leaf-Spine and PortLand Networks

Leaf and Spine



Spanning Trees: Ewwww!
Layer 2: SPB or Trill
Layer 3: OSPF

Leaf and Spine

- Type of Clos network
  - Essentially folded, but still N-to-N connections
  - Derived from old phone company architecture, invented in 1950s.
- All paths are same length from edge to edge
- Great for switch vendors
- Need to pick path, as can choose any middle router
- Very redundant
- Can implement at layer-2 or layer-3

Networking over time

- Flat – Gee whiz! We have a LAN!
  - Ut-Oh! Too many addresses to remember what to do with them all, never mind to manage each one individually
- Hierarchical – Look ma'! Global scale!
  - Ut-Oh! Hierarchy requires management. Creates hot spots and fragments address space. IP addresses limit migration. Forwarding is painful.
- Fat Tree – Many roots means many paths, relieves hot spots
  - Still hierarchical. Still fragment. Still needs managing. Etc.

Goals

- Connectivity – inside, outside, no loops, etc.
- Efficiency – short paths, no hot spots, etc.
- Self configuring – Humans effort can't be scaled in this way
- Robust – Consequences of failure aren't disproportional; Failure is recoverable
- VMs and Migration – Network structure doesn't compartmentalize VMs
- Commodity hardware – Certainly facilitates adoption

Implications

- Connectivity – Scale and hierarchy requires big routing/forwarding tables or impossibly large flat space. Routing protocols adapt to change slowly, leaving temporary inconsistency, e.g. loops, from failure and other changes.
- Efficiency – Look toward fat trees, clos networks
- Self-configuring – Look toward a discovery protocol, p2p or coordinated
- Robust – Rapidly learn changed, avoid single points of failure w/high responsibility
- Migration – Want a flat address space to avoid compartmentalization. Trying to fix above IP layer is a mess as it all comes back down and has to find its way.
- Commodity hardware – Services provided in off-switch software

Key Idea

- Minimize use of distributed state, because coordination for recovery or change takes time
- Maximize use of central state for lower volume services, because it can change/recover faster, and won't bottleneck at low volume

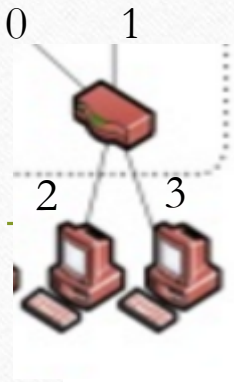
Portland Solution

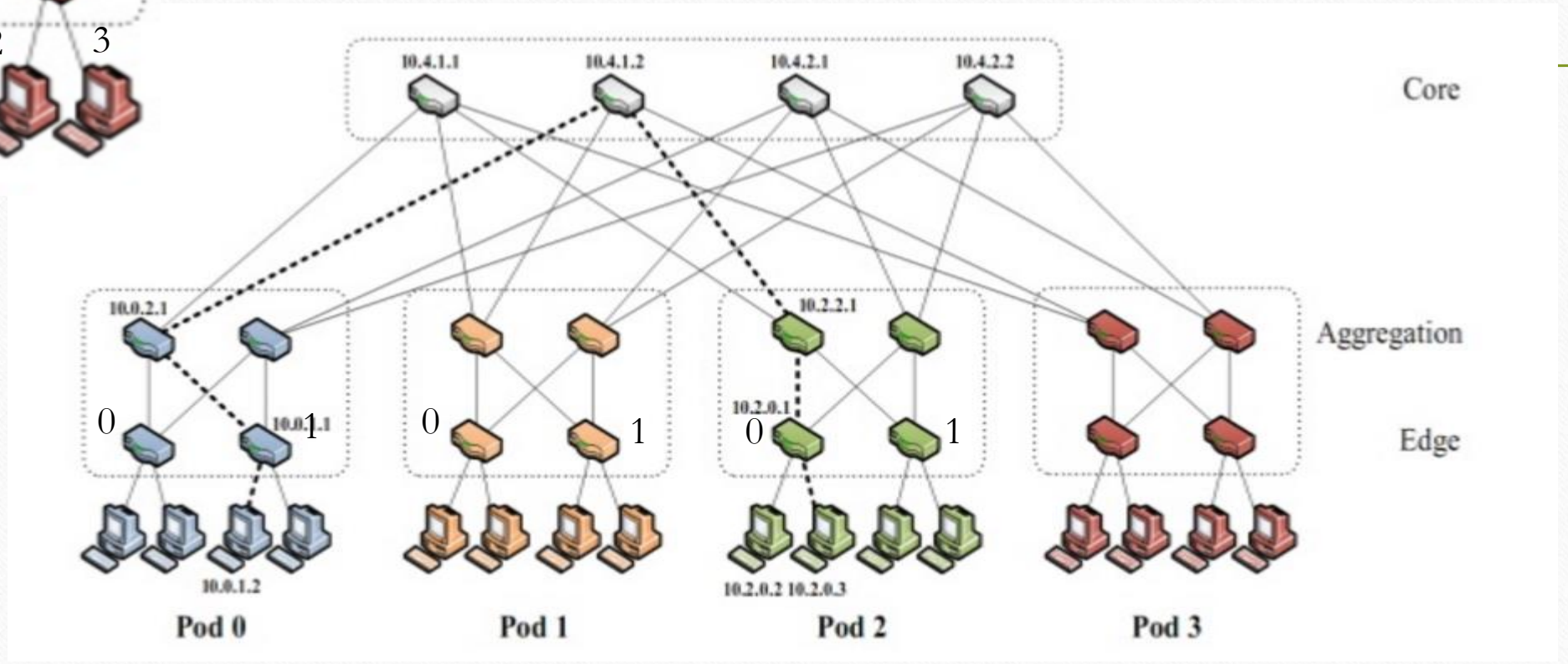
- Use commodity switches and off-load services into software on commodity server
- Start With Fat Tree for a topology without hot spots
- Use layer-2 to avoid routing, forwarding, and related complexity
- Separate host identifier from host location
  - IP addresses identify host, but not location, just and ID
  - Use “Pseudo MAC Address” to identify location at Level-2

PortLand Addresses

- Normally MAC addresses are arbitrary – no clue about location
  - IP normally is hierarchical, but here we are using it only as a host identifier
  - If MAC addresses are not tied to location, switch tables grow linearly with growth of network, i.e. O(n)
- PortLand uses hierarchical MAC addresses, called "Pseudo MAC" or PMAC addresses to provide for switch location
  - <pod:port:position:vmid>

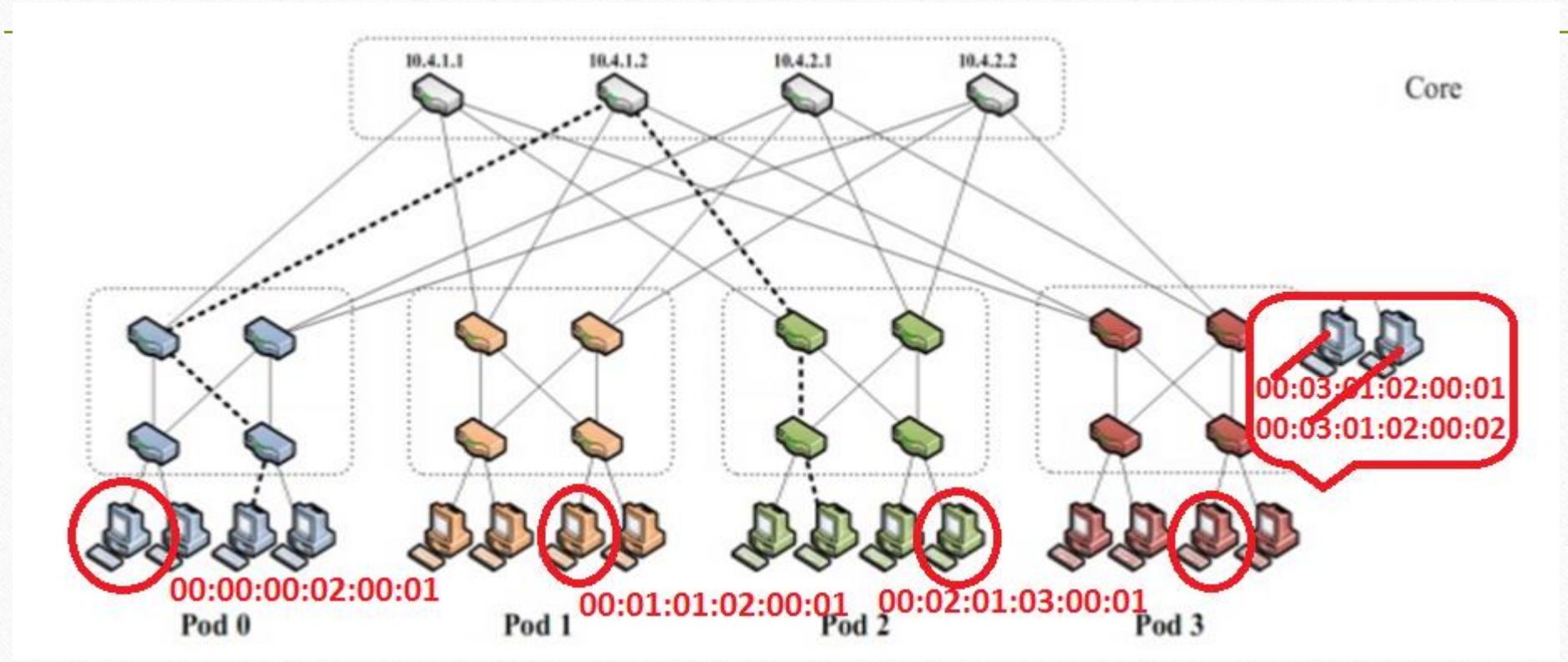
PortLand PMAC Addresses





PMAC: <pod.position.port.vmid> 48 bits: <16-bits.8-bits.8-bits.16-bits>

Portland PMAC Addresses



VM Migration

- Flat address space.
- IP address unchanged after migration, higher level doesn't see state change
- After migration IP<->PMAC changes, as PMAC is location dependent
- VM sends gratuitous ARP with new mapping.
- Fabric Manager receives ARP and sends invalidation to old switch
- Old switch sets flow table to software, causing ARP to be sent to any stray packets
  - Forwarding the packet is optional, as retransmit (if reliable) will fix delivery

Location Discovery: Configuring Switch IDs

- Humans = Not right Answer
- Discovery = Right Answer
- Send messages to neighbors – Get Tree Level
  - Hosts don't reply, so edge only hears back from above
  - Aggregate hears back from both levels
  - Core hears back only from aggregate
- Contact *Fabric Manager* with tree level to get ID
  - Fabric Manager is service running on commodity host
  - Assigns ID

Name Resolution: MAC→PMAC→IP

- End hosts continue to use *Actual* MAC (AMAC) addresses
  - Switches convert PMAC<->AMAC for the host
- Edge switch responsible for creating PMAC:AMAC mapping and telling Fabric Manager
  - Software on commodity server, can be replicated, etc. Simplicity is a virtue.
  - Mappings timed out of Fabric Manager's cache, if not used.
- ARPs are for PMACs
  - First ask fabric manager which keeps cache. Then, if needed, broadcast.

No loops, No Spanning Trees

- Forwarding can only go up the tree.
- Cycles not possible.

Failure

- Keep-alives like the link discovery messages
- Miss a keep alive? Tattle to the Fabric Manager
- Fabric manager tells effected switches, which adjust own tables.
- O(N) vs O(N^2) for traditional routing algorithms (Fabric Manager tells every switch vs every switch tells every switch)

Looking Back

- Connectivity – Hosts can talk! No possibility of loops
- Efficiency – Much less memory needed in switches, O(N) fault handlingh
- Self configuring – Discovery protocol + ARP
- Robust – Failure handling coordinated by FM
- VMs and Migration – Each has own IP address, each has own MAC address
- Commodity hardware – Nothing magic.