TOMORROW starts here.
Cisco FabricPath Technology and Design

BRKDCT-2081

Tim Stevenson
Distinguished Engineer, Technical Marketing
Session Abstract

“Provides an introduction to Cisco's FabricPath technology, which enables simplified high-performance L2 switching fabrics. Topics include a review of FabricPath technology and implementation details, use-case/design considerations including interaction with Classic Ethernet environments, and a brief discussion of how Cisco FabricPath relates to the IETF TRILL (Transparent Interconnect of Lots of Links) standard.”
Session Goal

- To provide you with a conceptual and technical understanding of Cisco FabricPath, including control-plane functions, data-plane forwarding model, and typical network designs
Agenda

- Introduction to FabricPath
  - FabricPath Forwarding
  - FabricPath Design
  - Key Takeaways
Introduction to FabricPath
Why Layer 2 in the Data Center?

- Certain protocols / applications rely on it
- Provides “plug-and-play” setup
- Allows virtual machine / workload mobility
Typical Data Center Design

Layer 2 benefits limited to a POD
Possible Solution for End-to-End L2?

Just extend STP to the whole network (!?)
Limitations of Traditional Layer 2

- Local problems have network-wide impact
- Tree topology provides limited bandwidth
- Tree topology introduces sub-optimal paths
- MAC address tables don’t scale
Cisco FabricPath combines benefits of Layer 3 routing with simplicity of Layer 2 switching.
FabricPath – An Ethernet Fabric

Fabric-wide intelligence ties network elements together
Why FabricPath?

- Reduction / elimination of Spanning-Tree Protocol (STP)
- Better stability and convergence characteristics
- Simplified configuration
- Leverage parallel paths
- Deterministic throughput and latency using typical designs
- “VLAN anywhere” – flexibility, L2 adjacency, and VM mobility
TRILL

- Transparent Interconnection of Lots of Links
- IETF Proposed Standard that “suggests applying modern network-layer routing protocols at the link layer”
  - RFC 5556, RFC 6325, etc.
- Many parallels to Cisco FabricPath
  - Cisco instrumental in definition of TRILL standard
- Standard continues to evolve
  - “Emulated switch”
  - “Multi-topology”
### FabricPath / TRILL Comparison

<table>
<thead>
<tr>
<th>Feature / Capability</th>
<th>FabricPath</th>
<th>TRILL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame routing (ECMP, TTL, RPF check, etc.)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VPC+</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Active-active FHRP / Anycast HSRP</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Multiple topologies</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Conversational learning</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>STP domain interaction</td>
<td>Isolated</td>
<td>Integrated</td>
</tr>
<tr>
<td>Inter-switch links</td>
<td>Point-to-point only</td>
<td>Point-to-point or shared</td>
</tr>
</tbody>
</table>
FAQ: When Will Nexus Products Support TRILL?

- All shipping hardware that supports FabricPath can support TRILL
- No near-term plan to implement TRILL control plane
- Why?
  - TRILL lacks capabilities critical for practical / real-world deployments
  - FabricPath provides superset of TRILL capabilities
  - No 3rd party platform provides full TRILL capability today
Agenda

- Introduction to FabricPath
  - FabricPath Forwarding
- FabricPath Design
- Key Takeaways
FabricPath Forwarding
FabricPath Forwarding – Control Plane

Key FabricPath control plane elements:

- **Routing table** – FabricPath IS-IS learns switch IDs (SIDs) and builds routing table
- **Multidestination trees** – FabricPath IS-IS elects roots and builds multidestination forwarding trees
- **Mroute table** – IGMP snooping learns group membership at the edge, FabricPath IS-IS floods group-membership LSPs (GM-LSPs) into the fabric
FAQ: Is This MAC-Based Routing?

- NO!
- Routing information consists of Switch IDs
- Forwarding in fabric based on Switch IDs, *not* MAC addresses
FabricPath Routing Table

- Contains shortest path(s) to each SID, based on link metrics / path cost
- Equal-cost multipath (ECMP) supported on up to 16 next-hop interfaces
FabricPath Routing Table

S100# sh fabricpath route
FabricPath Unicast Route Table
'a/b/c' denotes ftag/switch-id/subswitch-id
'[x/y]' denotes [admin distance/metric]
ftag 0 is local ftag
subswitch-id 0 is default subswitch-id

FabricPath Unicast Route Table for Topology-Default

0/100/0, number of next-hops: 0
via ----, [60/0], 0 day/s 04:43:51, local
1/10/0, number of next-hops: 1
via Po10, [115/20], 0 day/s 02:24:02, isis_fabricpath-default
1/20/0, number of next-hops: 1
via Po20, [115/20], 0 day/s 04:43:25, isis_fabricpath-default
1/30/0, number of next-hops: 1
via Po30, [115/20], 0 day/s 04:43:25, isis_fabricpath-default
1/40/0, number of next-hops: 1
via Po40, [115/20], 0 day/s 04:43:25, isis_fabricpath-default
1/200/0, number of next-hops: 4
via Po10, [115/40], 0 day/s 02:24:02, isis_fabricpath-default
via Po20, [115/40], 0 day/s 04:43:06, isis_fabricpath-default
via Po30, [115/40], 0 day/s 04:43:06, isis_fabricpath-default
via Po40, [115/40], 0 day/s 04:43:06, isis_fabricpath-default
1/300/0, number of next-hops: 4
via Po10, [115/40], 0 day/s 02:24:02, isis_fabricpath-default
via Po20, [115/40], 0 day/s 04:43:25, isis_fabricpath-default
via Po30, [115/40], 0 day/s 04:43:25, isis_fabricpath-default
via Po40, [115/40], 0 day/s 04:43:25, isis_fabricpath-default

S100#
FAQ: How Are ECMP Load-Sharing Decisions Made?

- ECMP path chosen based on hash function
- Hash uses **SIP/DIP + L4 + VLAN** by default
- Use **show fabricpath load-balance unicast** to determine ECMP path for a given packet
FabricPath Multidestination Trees

- Multidestination traffic constrained to tree topology
  - Network-wide identifier (Ftag) assigned to each tree
- Support for multiple trees provides multipathing for multidestination traffic
  - Two trees per topology supported today
- Root switch elected for each multidestination tree in each FabricPath topology

```
S100 S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S100 S10 S20 S30
```

```
S100 S20 S30 S40
```

```
S100 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```

```
S10 S20 S30 S40
```
Multidestination Root Selection

- FabricPath network elects a primary root switch for the first multidestination tree in the topology
- Switch with highest priority value becomes root for the tree
  - Tie break: root priority → highest system ID → highest SID
- Primary root determines roots of additional trees and announces them in Router Capability TLV
  - Roots spread among available switches to balance load
FAQ: Trees? Roots? Sounds Like Spanning Tree...

- NO! – More like IP multicast routing
- Trees do NOT dictate forwarding path of unicast frames, only multidestination frames
- Multiple trees allow load-sharing for any multidestination frames
- Control plane state further constrains IP multicast forwarding (based on mrouter and receiver activity)
Best Practice: Identify the Roots

- Use the **root-priority** command to explicitly identify primary, secondary, and tertiary root switches.
- Optimizes forwarding paths for multidestination frames.
- Simplifies troubleshooting.
FabricPath Forwarding – Data Plane

Key FabricPath data plane elements:

- **MAC table** – Hardware performs MAC lookups at CE/FabricPath edge only
- **Switch table** – Hardware performs destination SID lookups to forward unicast frames to other switches
- **Multidestination table** – Hash function selects tree*, multidestination table identifies on which interfaces to flood based on selected tree

* Nexus 7000 F1 modules always use Tree 1 for broadcast and unknown unicast
FabricPath MAC Table

- Edge switches perform MAC table lookups on ingress frames
- Lookup result identifies output interface or destination FabricPath switch
## FabricPath MAC Table

S100# `sh mac address-table dynamic vlan 100`

Legend:
- `*` - primary entry, `G` - Gateway MAC, `(R)` - Routed MAC, `O` - Overlay MAC
- `age` - seconds since last seen, `+` - primary entry using vPC Peer-Link,
- `(T)` - True, `(F)` - False

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure</th>
<th>NTFY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0000.0000.000a</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>0000.0279.5af1</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>0000.02b0.550e</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>0000.09c3.d9d2</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>0000.0c9f.f001</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>0026.51bf.fb41</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>0026.51cf.ae41</td>
<td>dynamic</td>
<td>660</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>0c00.060b.5481</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>0000.0000.000b</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>1000.021a.ba87</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>1400.0579.0395</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>1400.0640.5dc8</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>1400.092d.9cd4</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>1800.0536.0ded</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>1800.0992.3254</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>1c00.0128.901e</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>0000.0000.000c</td>
<td>dynamic</td>
<td>120</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>1c00.0ae2.7759</td>
<td>dynamic</td>
<td>120</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>2000.0134.9a04</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>2400.029b.533e</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

**Legend:**
- `Local MAC entry` (directly connected to CE edge port)
- `Remote MAC entries` (reached through FabricPath)
Forwarding through the Fabric – FabricPath Encapsulation

**Classical Ethernet Frame**

- DMAC
- SMAC
- 802.1Q
- Etype
- Payload
- CRC

**Cisco FabricPath Frame**

- DMAC
- SMAC
- 802.1Q
- Etype
- Payload
- CRC (new)

- **Switch ID** – Unique number identifying each FabricPath switch
- **Ftag (Forwarding tag)** – Unique number identifying topology or multidestination tree
FabricPath Switch ID (SID)

- Every FabricPath switch automatically assigned a **Switch ID**
  - Optionally, network administrator can manually configure SIDs
- FabricPath network automatically detects conflicting SIDs and prevents data path initialization on violating switch
- Encoded in “Outer MAC addresses” of FabricPath MAC-in-MAC frames
Best Practice: Manually Assign SIDs

- Use the `fabricpath switch-id` command to manually assign Switch IDs
- Simplifies management and eases troubleshooting
- Enables deterministic number schemes, e.g.:
  - Spine switches assigned two-digit SIDs
  - Leaf switches assigned three-digit SIDs
  - VPC+ virtual SIDs assigned four-digit SIDs
  - etc.
FabricPath Forwarding Tag (Ftag)

- **Forwarding tag** – Unique 10-bit number encoded in FabricPath header
- Overloaded field that identifies FabricPath topology or multidestination tree
- For unicast packets, identifies which FabricPath IS-IS topology to use
- For multidestination packets (broadcast, multicast, unknown unicast), identifies which multidestination tree to use
FAQ: What about VLANs in FabricPath?

- VLANs **are still relevant** in FabricPath!
- Every frame in fabric carries 802.1Q
- VLANs still define a broadcast domain in FabricPath – define scope of flooding
- FabricPath switches still look at VLAN ID – including ‘core / spine’ switches
- Frames tagged with VLAN ID that does not exist in the VLAN database are dropped
Best Practice: Configure All VLANs on All Switches in Topology

- If a FabricPath switch belongs to a topology, configure **all** VLANs in that topology on that switch
- Failure to do so can result in multidestination forwarding issues (black-holing)
- Deviate from this Best Practice carefully!
  - Specific designs can work (one such design discussed later)
  - But make sure to account for all failure cases!
FAQ: What about QoS in FabricPath?

- FabricPath hardware designed to accommodate for FabricPath encapsulation
- All FabricPath-encapsulated frames carry 802.1Q/802.1p (COS)
- IP packets in FabricPath also carry DSCP
- For FabricPath-encapsulated frames, hardware can still:
  - Queue based on COS/DSCP
  - Match based on L2/L3/L4 header information
  - Match/set DSCP
Putting It All Together – Host A to Host B

(1) Broadcast ARP Request

FabricPath MAC Table on S100

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/13 (local)</td>
</tr>
</tbody>
</table>

Multidestination Trees on Switch 10

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>po100,po200,po300</td>
</tr>
<tr>
<td>2</td>
<td>po100</td>
</tr>
</tbody>
</table>

Hash Result

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>po10</td>
</tr>
<tr>
<td>2</td>
<td>po10,po20,po30,po40</td>
</tr>
</tbody>
</table>

Root for Tree 1

SMAC→A

Hash

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>po10,po20,po30,po40</td>
</tr>
<tr>
<td>2</td>
<td>po40</td>
</tr>
</tbody>
</table>

Root for Tree 2

SMAC→A

MAC of directly-connected devices unconditionally

Don’t learn MACs from unknown flood frames
### MAC Address Tables After Broadcast ARP

#### S100:

```bash
S100# sh mac-address-table dynamic
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen, + - primary entry using vPC Peer-Link

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure</th>
<th>NTFY Ports/SWID.SSID.LID</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 10</td>
<td>0000.0000.000a</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>Eth1/13</td>
</tr>
</tbody>
</table>
```

S100#

#### S10 (and S20, S30, S40, S200):

```bash
S10# sh mac-address-table dynamic
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen, + - primary entry using vPC Peer-Link

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure</th>
<th>NTFY Ports/SWID.SSID.LID</th>
</tr>
</thead>
</table>
```

S10#

#### S300:

```bash
S300# sh mac-address-table dynamic
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen, + - primary entry using vPC Peer-Link

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure</th>
<th>NTFY Ports/SWID.SSID.LID</th>
</tr>
</thead>
</table>
```

S300#
Broadcast Forwarding

- Ingress FabricPath switch determines which tree to use based on hash result
- Outer Destination MAC remains all-ones (same as Inner DMAC)
- Other FabricPath switches honor Tree ID selected by ingress switch (Tree 1 in this case) – flood frame on all core ports belonging to selected tree
- Edge FabricPath switches remove FabricPath header and flood in VLAN – Flood FabricPath encapsulated frame on other core ports as well, if necessary
FAQ: What Is the Destination SID for a Multidestination Frame?

- Broadcast – Copy inner DMAC to outer DMAC
- Multicast – Copy inner DMAC to outer DMAC
- Unknown Unicast – Use reserved multicast MAC “MC1” (010F.FFC1.01C0)
Putting It All Together – Host A to Host B
(2) Unicast ARP Reply

Multidestination Trees on Switch 10

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>po100, po200, po300</td>
</tr>
<tr>
<td>2</td>
<td>po100</td>
</tr>
</tbody>
</table>

Ftag → Multidestination Trees on Switch 100

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>po10</td>
</tr>
<tr>
<td>2</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

Ftag → FabricPath MAC Table on S100

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/13 (local)</td>
</tr>
<tr>
<td>B</td>
<td>300 (remote)</td>
</tr>
</tbody>
</table>

If DMAC is known, then learn remote MAC

Hash Result → FabricPath MAC Table on S300

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>2</td>
<td>po40</td>
</tr>
</tbody>
</table>

Ftag → FabricPath MAC Table on S300

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>e2/29 (local)</td>
</tr>
</tbody>
</table>

Multidestination Trees on Switch 300

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>2</td>
<td>po40</td>
</tr>
</tbody>
</table>

DMAC → A
SMAC → B
Payload

010F.FFC1.01C0*
SA → 300
Ftag → 1
DMAC → A
SMAC → B
Payload

© 2013 Cisco and/or its affiliates. All rights reserved.
Cisco Public
MAC Address Tables After Unicast ARP Reply

- **S100:**
  S100# sh mac address-table dynamic
  Legend:
  * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
  age - seconds since last seen, + - primary entry using vPC Peer-Link
  VLAN     MAC Address      Type      age     Secure NTFY Ports/SWID.SSID.LID
  ----------------- ------- ------ -------- -----------------+---------------------------
  * 10 0000.0000.000a    dynamic  90       F    F  Eth1/13
  10 0000.0000.000b    dynamic  60       F    F  300.0.64
  S100#

- **S300:**
  S300# sh mac address-table dynamic
  Legend:
  * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
  age - seconds since last seen, + - primary entry using vPC Peer-Link
  VLAN     MAC Address      Type      age     Secure NTFY Ports/SWID.SSID.LID
  ----------------- ------- ------ -------- -----------------+---------------------------
  * 10 0000.0000.000b    dynamic  0        F    F  Eth2/29
  S300#

S100 learns MAC B as remote entry reached through S300
MAC B learned as local entry on e2/29
Unknown Unicast Forwarding

- Ingress FabricPath switch determines which tree to use based on hash result.
- Outer Destination MAC set to well-known “flood to fabric” multicast address (MC1)*
- Other FabricPath switches honor Tree ID selected by ingress switch (Tree 1 in this case) – flood frame on all core ports belonging to selected tree.
- Edge FabricPath switches remove FabricPath header and flood in VLAN – Flood FabricPath encapsulated frame on other core ports as well, if necessary.

*MC1 = 010F.FFC1.01C0
FAQ: What Is Conversational MAC Learning?

- **New MAC learns** performed only on unicast frames destined to a local MAC address
- Edge switches only need to learn:
  - Locally connected host MACs
  - MACs with which those local hosts are bidirectionally communicating
- Reduces MAC table capacity requirements on edge switches
- Beware of devices that have conversations with every other host (e.g. default gateway, file server, firewall, etc.)
  - Design options discussed later
FAQ: Is MAC Learning in FabricPath Software-Based?

- NO!
- Binding of MAC address to destination SID at FabricPath edge switches is completely hardware based
- FabricPath-capable platforms have hardware logic to perform conversational MAC learning without punting anything to software
FAQ: What Happens If a Host Moves?

FabricPath rules for MAC learning on flood frames:

- **Do not** perform new learns based on broadcast / unknown unicast frames
- **Do** perform MAC table updates based on broadcast / unknown unicast frames
- **Do** perform new learns based on multicast frames (required for learning gateway MACs)

So how are host moves handled?

- Same as in Classical Ethernet
- Incumbent on moving host to tell the network it moved
  - Gratuitous ARP, Reverse ARP are typical mechanisms
- FabricPath switches update existing entries based on these frames
FAQ: What Happens When the MAC Table Capacity Is Exceeded?

- Same thing as in traditional Ethernet – new MAC learns may fail
  - MAC lookups based on hash function – hash collisions are possible
  - If a hash collision occurs on a new SMAC, the MAC is not learned
- If a DMAC lookup returns a Miss, the packet is flooded as unknown unicast

![Diagram showing MAC table hash collisions and lookups](image-url)
Putting It All Together – Host A to Host B
(3) Unicast Data

FabricPath Routing Table on S30
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S100
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S300
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

MAC Table on S100
<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/13 (local)</td>
</tr>
<tr>
<td>B</td>
<td>300 (remote)</td>
</tr>
</tbody>
</table>

If DMAC is known, then learn remote MAC

If SMAC is known, then...

Putting It All Together – Host A to Host B
(3) Unicast Data

FabricPath Routing Table on S30
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S100
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S300
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

MAC Table on S100
<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/13 (local)</td>
</tr>
<tr>
<td>B</td>
<td>300 (remote)</td>
</tr>
</tbody>
</table>

If DMAC is known, then learn remote MAC

If SMAC is known, then...

Putting It All Together – Host A to Host B
(3) Unicast Data

FabricPath Routing Table on S30
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S100
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S300
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

MAC Table on S100
<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/13 (local)</td>
</tr>
<tr>
<td>B</td>
<td>300 (remote)</td>
</tr>
</tbody>
</table>

If DMAC is known, then learn remote MAC

If SMAC is known, then...

Putting It All Together – Host A to Host B
(3) Unicast Data

FabricPath Routing Table on S30
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S100
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S300
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

MAC Table on S100
<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/13 (local)</td>
</tr>
<tr>
<td>B</td>
<td>300 (remote)</td>
</tr>
</tbody>
</table>

If DMAC is known, then learn remote MAC

If SMAC is known, then...

Putting It All Together – Host A to Host B
(3) Unicast Data

FabricPath Routing Table on S30
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S100
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S300
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

MAC Table on S100
<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/13 (local)</td>
</tr>
<tr>
<td>B</td>
<td>300 (remote)</td>
</tr>
</tbody>
</table>

If DMAC is known, then learn remote MAC

If SMAC is known, then...

Putting It All Together – Host A to Host B
(3) Unicast Data

FabricPath Routing Table on S30
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S100
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
<tr>
<td>S30</td>
<td>po30</td>
</tr>
<tr>
<td>S40</td>
<td>po40</td>
</tr>
<tr>
<td>S200</td>
<td>po10, po20, po30, po40</td>
</tr>
<tr>
<td>S300</td>
<td>po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath Routing Table on S300
<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>po10</td>
</tr>
<tr>
<td>S20</td>
<td>po20</td>
</tr>
</tbody>
</table>
# MAC Address Tables After Unicast Data

## S100:

S100# `sh mac address-table dynamic`

Legend:

* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen, + - primary entry using vPC Peer-Link

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure</th>
<th>NTFY Ports/SWID.SSID.LID</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0000.0000.000a</td>
<td>dynamic</td>
<td>90</td>
<td>F</td>
<td>F Eth1/13</td>
</tr>
<tr>
<td>10</td>
<td>0000.0000.000b</td>
<td>dynamic</td>
<td>60</td>
<td>F</td>
<td>300.0.64</td>
</tr>
</tbody>
</table>

S100#

## S300:

S300# `sh mac address-table dynamic`

Legend:

* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen, + - primary entry using vPC Peer-Link

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure</th>
<th>NTFY Ports/SWID.SSID.LID</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0000.0000.000a</td>
<td>dynamic</td>
<td>30</td>
<td>F</td>
<td>F Eth2/29</td>
</tr>
<tr>
<td>*</td>
<td>0000.0000.000b</td>
<td>dynamic</td>
<td>90</td>
<td>F</td>
<td>100.0.12</td>
</tr>
</tbody>
</table>

S300#

S100 learns MAC A as remote entry reached through S100
Creating Multicast State – IGMP Snooping

- IGMP snooping learns about interested receivers on edge switches
- Membership tracked on CE ports based on receiving IGMP reports / leaves
Creating Multicast State – GM-LSPs

- FabricPath IS-IS uses Group Membership LSPs (GM-LSPs) to build multicast forwarding state for the fabric
- Flooded to other switches to advertise which edge switches need which multicast groups
- Builds Layer 2 multicast forwarding state for FabricPath core ports
Multicast State
FabricPath Edge Switch with Receiver

S100# sh ip igmp snooping groups
Type: S - Static, D - Dynamic, R - Router port, F - Fabricpath core port

<table>
<thead>
<tr>
<th>Vlan</th>
<th>Group Address</th>
<th>Ver</th>
<th>Type</th>
<th>Port list</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td><em>/</em></td>
<td>v2</td>
<td>D</td>
<td>Eth1/13</td>
</tr>
<tr>
<td>10</td>
<td>239.0.0.1</td>
<td>v2</td>
<td>D</td>
<td>Eth1/13</td>
</tr>
</tbody>
</table>

S100# sh fabricpath isis ip mroute | section 239.0.0.1
VLAN 10: (*, 239.0.0.1)
- Switch-id 200, uptime: 00:00:28, isis
- Interface Ethernet1/13, uptime: 00:00:30, igmp

S100# sh fabricpath mroute | section 239.0.0.1
(vlan/10, 0.0.0.0, 239.0.0.1), uptime: 00:00:30, isis igmp
- Outgoing interface list: (count: 2)
  - Switch-id 200, uptime: 00:00:28, isis
  - Interface Ethernet1/13, uptime: 00:00:30, igmp

IGMP snooping knows local OIF...
IS-IS knows remote receiver...
M2RIB knows about both
Pruned Forwarding Trees for IP Multicast Groups

**Multidestination Tree 1**

- Multidestination Tree 1
- G1 Pruned Tree
- Root 1
- FabricPath
- Rcvr-G1
- Rcvr-G1
- Can't go this way – not part of Tree 1!

**Multidestination Tree 2**

- Multidestination Tree 2
- G1 Pruned Tree
- Root 2
- FabricPath
- Rcvr-G1
- Rcvr-G1
- Can't go this way – not part of Tree 2!
FabricPath IP Multicast Data Plane

Tree Selection and Group Lookup on Ingress Switch

Packet data → Hash

Tree 1

<table>
<thead>
<tr>
<th>VLAN</th>
<th>Tree (Flag)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

FabricPath MAC Table

<table>
<thead>
<tr>
<th>Tree (Flag)</th>
<th>VLAN</th>
<th>Group</th>
<th>SID</th>
<th>IFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>G1</td>
<td>S100,S200</td>
<td>po2</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>G1</td>
<td>S100,S200</td>
<td>po3</td>
</tr>
</tbody>
</table>
FabricPath IP Multicast Data Plane
Group Lookup on Core Switch

FabricPath MAC Table

<table>
<thead>
<tr>
<th>Tree (Flag)</th>
<th>VLAN</th>
<th>Group</th>
<th>SID</th>
<th>IFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>G1</td>
<td>S100,S200</td>
<td>po4,po5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>G1</td>
<td>S100,S200</td>
<td>po4,po5</td>
</tr>
</tbody>
</table>

Multidestination Tree 1
G1 Pruned Tree

Tree 1 →

Data Traffic
FabricPath IP Multicast Data Plane
Group Lookup on Egress Switches

FabricPath MAC Table

<table>
<thead>
<tr>
<th>Tree (Flag)</th>
<th>VLAN</th>
<th>Group</th>
<th>SID</th>
<th>IFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>G1</td>
<td>S100,S200</td>
<td>po6,e1/29</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>G1</td>
<td>S100,S200</td>
<td>po6,e1/29</td>
</tr>
</tbody>
</table>

Data Traffic

Rcvr-G1

Multidestination Tree 1
G1 Pruned Tree
Best Practice: Connect Dual-Homed CE Devices via VPC+

- Dual-homed devices should be connected using VPC+
- Provides active/active uplinks from CE to FabricPath
- ECMP toward CE-attached hosts within fabric
- Removes complexity of STP integration
  - BPDUs still filtered at edge
  - TCNs not propagated through fabric
VPC+ – Physical Topology

- Peer link runs as FabricPath core port
- Peer link and PKA required
- VPCs configured as normal
- VLANs must be FabricPath VLANs
- No requirements for attached devices other than port-channel support

FabricPath

MAC A

MAC B

MAC C
VPC+ – Logical Topology

Virtual switch introduced

FabricPath

MAC A

MAC B

MAC C
Remote MAC Entries for VPC+

```
S200# sh mac address-table dynamic
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen, + - primary entry using vPC Peer-Link

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure</th>
<th>NTFY Ports/SWID.SSID.LID</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0000.0000.000c</td>
<td>dynamic</td>
<td>1500</td>
<td>F</td>
<td>Eth1/30</td>
</tr>
<tr>
<td>10</td>
<td>0000.6500.000a</td>
<td>dynamic</td>
<td>1500</td>
<td>F</td>
<td>1000.11.4513</td>
</tr>
</tbody>
</table>
```

Remote Switch ID 1000, not S10 or S20

View from S200

MAC A
FabricPath Routing for VPC+

S200# sh fabricpath route topology 0 switchid 1000
FabricPath Unicast Route Table
'a/b/c' denotes ftag/switch-id/subswitch-id
'[x/y]' denotes [admin distance/metric]
ftag 0 is local ftag
subswitch-id 0 is default subswitch-id

FabricPath Unicast Route Table for Topology-Default
1/1000/0, number of next-hops: 2
via Po1, [115/10], 0 day/s 01:09:56, isis_l2mp-default
via Po2, [115/10], 0 day/s 01:09:56, isis_l2mp-default

Route to S1000 (from S200)
ECMP toward S1000 from S200
VPC+ and Active/Active HSRP

Physical Topology

Logical Topology

FabricPath MAC Table on S200

FabricPath Routing Table on S200

FabricPath MAC Table on S200

FabricPath Routing Table on S200

HSRP Active

HSRP Standby

0100.5E00.0002
SSID→1000
0100.5E00.0002
SMAC→HSRP
Payload

FabricPath

Logical Topology

FabricPath

FabricPath

FabricPath

FabricPath

MAC | IF/SID
---|---
HSRP | S1000 (remote)
SID | IF
---|---
S1000 | po1,po2

HSRP MAC

SID | IF
---|---
S1000 | po1,po2

SSID→1000

Payload

0100.5E00.0002
SSID→1000
0100.5E00.0002
SMAC→HSRP
Payload

FabricPath

FabricPath

FabricPath

FabricPath

MAC | IF/SID
---|---
HSRP | S1000 (remote)
SID | IF
---|---
S1000 | po1,po2

HSRP MAC

SID | IF
---|---
S1000 | po1,po2
Anycast HSRP (NX-OS 6.2)

Physical Topology

HSRP Active

SVI

HSRP Standby

SVI

HSRP Listen

SVI

HSRP Listen

SVI

0100.5E00.0002
SSID--1000
0100.5E00.0002
SMAC--HSRP
Payload

S10
S20
S30
S40

FabricPath

MAC Table on S200

MAC | IF/SID
-----|-----
HSRP | S1000 (remote)

Routing Table on S200

SID | IF
----|-----
S1000 | po1,po2,po3,po4

Logical Topology

FabricPath

MAC Table on S200

MAC | IF/SID
-----|-----
HSRP | S1000 (remote)

Routing Table on S200

SID | IF
----|-----
S1000 | po1,po2,po3,po4

HSRP MAC

po1

po2

po3

po4
n-Way Active HSRP in FabricPath

VPC+ with FHRP / Anycast HSRP

- Hellos sent by Active router
  - Outer destination MAC is VMAC
  - Outer source is virtual SID

- FabricPath edge switches learn VMAC as reached through virtual SID

- Traffic destined to VMAC leverages ECMP

- Any VPC+ peer / Anycast member can route traffic destined to VMAC
## VPC+ versus Anycast for $n$-Way Active HSRP

<table>
<thead>
<tr>
<th></th>
<th>VPC+</th>
<th>Anycast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of active routers</td>
<td>Two</td>
<td>Four (NX-OS 6.2)</td>
</tr>
<tr>
<td>Peer link / Peer keepalive link</td>
<td>Required</td>
<td>Not Required</td>
</tr>
<tr>
<td>Leaf software requirement</td>
<td>None</td>
<td>NX-OS 6.2-based</td>
</tr>
</tbody>
</table>
Agenda

- Introduction to FabricPath
- FabricPath Forwarding
  - FabricPath Design
- Key Takeaways
FabricPath Designs
FabricPath Designs

- Explore a variety of FabricPath designs and evaluate how they meet key design criteria.
- Assumption is the choice to go L2/FabricPath has already been made.
  - Objective is not to debate "Layer 2 vs. Layer 3" or "why Layer 2 in the Data Center?"
- Introduce concepts / design building-blocks to help you build a design that meets your requirements.
FabricPath Designs

High-level design options considered in this presentation:

- Routing at Aggregation
- Centralized Routing
- Multi-POD
Routing at Aggregation Designs

Routed core

Aggregation

Access

Layer 3 Link
Layer 2 CE
Layer 2 FabricPath
Routing at Aggregation

Key Design Highlights

- Evolution of current design practices

- Aggregation layer functions as FabricPath spine and L2/L3 boundary
  - FabricPath bridging for East ↔ West intra-VLAN traffic
  - SVIs for East ↔ West inter-VLAN routing
  - Routed uplinks for North ↔ South routed flows

- Access layer provides pure L2 functions
  - FabricPath core ports facing aggregation layer
  - CE edge ports facing hosts
  - Option for VPC+ for active/active host connections
Routing at Aggregation
Two Spine Design

- Simplest design option
- Extension of traditional aggregation/access designs

Immediate benefits:
- Simplified configuration
- Removal of STP
- Traffic distribution over all uplinks without VPC port-channels
- Active/active gateways
- “VLAN anywhere” at access layer
- Topological flexibility
  - Direct-path forwarding option
  - Easily provision additional access ↔ aggregation bandwidth
  - Easily deploy L4-7 services
  - Option for VPC+ for legacy access switches

This design meets MANY customer requirements!
Routing at Aggregation

Two Spine Design Details

- SVIs/routed ports provided by Nexus 7000 M-Series or F2E modules, or Nexus 6004
- FabricPath core ports provided by F-Series modules or Nexus 6004
- HSRP between agg switches for FHRP
- L2/L3 boundary
- Can run VPC+ for active/active HSRP
- Nexus 7000 F-Series modules for EoR/MoR access
- Nexus 5500/6000 for ToR access
- SVIs
- Active
- Standby
- VPC+
- Nexus 5500/6000 for ToR access
- Layer 3 Link
- Layer 2 CE
- Layer 2 FabricPath
Routing at Aggregation
Anycast HSRP (NX-OS 6.2)

Anycast HSRP between agg switches

All Anycast HSRP forwarders share same VIP and VMAC

L2/L3 boundary

Routed traffic spread over spines based on ECMP

Hosts resolve shared VIP to shared VMAC

Anycast HSRP

GWY MAC A → L1, L2, L3, L4

GWY MAC A

GWY IP X

GWY MAC A

GWY IP X

GWY MAC A

GWY IP X

GWY MAC A

GWY IP X

GWY MAC A

FabricPath

Layer 3 Link
Layer 2 CE
Layer 2 FabricPath
MAC Scale in Routing at Aggregation Designs

F2/F2E at aggregation:
- 16K unique host MACs when SVIs enabled
  - With SVIs, any ingress SOC must know enough information to route packets to any other VLAN, regardless of whether that VLAN exists on one of its ports
  - $n \times 16K$ if SVI VLAN-ranges spread over multiple router pairs

M+F at aggregation:
- 16K unique host MACs due to mixed chassis learning behavior prior to NX-OS 6.2
  - FabricPath core ports must learn SMACs on ingress
  - Several typical topologies can result in MAC table overflow (e.g., aggregation ISL/VPC+ peer-link)
- 128K unique host MACs with “proxy L2 learning” in NX-OS 6.2
  - Disables core port learning in mixed chassis, and uses M-series MAC table only
  - Requires that access switches install all their local MACs on their core ports (behavior starts in NX-OS 6.1)

Nexus 6004 at aggregation:
- 256K unique host MACs with Nexus 6000
  - High-scale, but requires 40G uplinks, or breakout cables
  - Loss of high availability/ISSU and certain aggregation features
Centralized Routing Designs

Layer 3 services leaf switches
FabricPath spine
Server access leaf switches
Centralized Routing Designs

Alternative View

Leaf switches each have “personality” – most for server access…

…but some for Layer 3 services (routing) and/or L4-7 services (SLB, FW, etc.)

FabricPath spine

Server access leaf switches

FabricPath

Layer 3 services leaf switches

L2/L3 boundary

L3
Centralized Routing

Key Design Highlights

- Paradigm shift with respect to typical designs
- Traditional “aggregation” layer becomes pure FabricPath spine
  - Provides uniform any-to-any connectivity between leaf switches
  - In simplest case, only FabricPath bridging occurs in spine
  - Optionally, some CE edge ports exist to provide external router connections
- FabricPath leaf switches, connecting to spine, have specific “personality”
  - Most leaf switches provide server connectivity, like traditional access switches in “Routing at Aggregation” designs
  - Two or more leaf switches provide L2/L3 boundary, inter-VLAN routing and North ↔ South routing
  - Other (or same) leaf switches have L4-7 services personality (future)
- Decouples L2/L3 boundary and L4-7 services provisioning from spine
  - Simplifies spine design
Centralized Routing
Single Router Pair (FabricPath-Connected Leaf)

FabricPath spine with F-Series modules or Nexus 6004 provides transit fabric (no routing, no MAC learning)

FabricPath core ports provided by F-series modules or Nexus 6004

Can run VPC+ or Anycast for active/active HSRP

All VLANs available at all leaf switches

SVIs for all VLANs on leaf L3 services switch pair (provided by M-series, F2/F2E modules, or Nexus 6004)

HSRP between L3 services switches for FHRP

L2/L3 boundary
Centralized Routing
Single Router Pair (FabricPath-Connected Leaf)
Centralized Routing
Multiple Router Pairs (FabricPath-Connected Leaf)

Routing adjacencies provide transit path for inter-set routing

Variations include >2 L3 services routers with Anycast HSRP, etc.

All VLANs available at all access switches

This router pair has SVIs for some VLANs (VLAN set 1)

This pair has SVIs for other VLANs (VLAN set 2)

Layer 3 Link
Layer 2 CE
Layer 2 FabricPath
Centralized Routing
Details of Multiple Router Pairs Option

- Discreet SVI “sets”, with one set per L3-services leaf pair
  - Transit VLAN to provide inter-set routing

- Requires appropriate platform for L3 services leaf switches to avoid MAC learning on core ports
  - Nexus 7000 with F2E modules, or M+F with “proxy L2 learning” feature (NX-OS 6.2)
  - Nexus 6004

- All leaf switches must have all VLANs defined (due to multidestination tree-building behavior)
  - With multi-topology (NX-OS 6.2), can prune VLANs from certain leaf switches
Centralized Routing
Multiple Router Pairs (FabricPath-Connected Leaf)

INTER-VLAN ROUTED FLOWS (Inter-VLAN-set)

Transit routing

Layer 3 Link
Layer 2 CE
Layer 2 FabricPath
MAC Scale with Nexus 7000 F-Series at Spine

With F1/F2/F2E FabricPath core ports only at spine
- Core ports do not learn MAC addresses*
- MAC scale not gated by spine switches

With F1/F2E FabricPath core ports plus CE edge ports at spine
- Core ports do not learn MAC addresses
- CE edge ports perform per-SOC conversational learning
  - Only MACs of VLANs on SOC front-panel ports learned
  - No practical limit to MAC scale – theoretically allows for (16K * num_of_SOCs) MACs

Note: F2 requires no hardware fabricpath mac-learning option when functioning as pure spine
Multi-Pod Designs

- **Routed core**
- **Aggregation**
- **Access**

POD 1: L3
POD 2: Layer 2/L3 boundary
POD 3: Layer 3 Link, Layer 2 CE, Layer 2 FabricPath

FabricPath core
Multi-Pod Design

Key Design Highlights

- Hybrid of elements used in other design alternatives
  - Combines “Routing at Aggregation” and “Centralized Routing” design elements

- Three possible classes of VLAN in FabricPath domain
  - POD-local – VLANs exist only in one POD
  - DC-wide – VLANs exist in all PODs
  - Multi-POD – VLANs exist only in subset of PODs (not illustrated)
Multi-Pod Design

FabricPath core interconnects PODs for bridging in DC-wide VLANs

Active/Active HSRP for VLANs 100-199

Active/Active HSRP for VLANs 200-299

Active/Active HSRP for VLANs 300-399

Core must include VLANs from all PODs due to multidestination tree behavior

L3

Core

VLANs 100-199
VLANs 200-299
VLANs 300-399
VLANs 2000-2099

Mixed FabricPath/CE POD

Any device

Layer 3 Link
Layer 2 CE
Layer 2 FabricPath

POD 1
VLANs 100-199
VLANs 2000-2099

POD 2
VLANs 200-299
VLANs 2000-2099

POD 3
VLANs 300-399
VLANs 2000-2099

DC-wide VLANs

Native FabricPath PODs

Active/Active HSRP for VLANs 2000-2099

POD local VLANs

Layer 3 Link
Layer 2 CE
Layer 2 FabricPath

Active/Active HSRP for VLANs 2000-2099

Mixed FabricPath/CE POD

Any device
Multi-Pod Design
FabricPath Multi-Topology (NX-OS 6.2)

Core ports in POD belong to default topology and also mapped to POD-local topology.

Only DC-wide VLANs exist in FabricPath core.

CORE VLANs 2000-2099

Layer 2 FP Default Topology
POD 1 Topology + Default Topology
POD 2 Topology + Default Topology
POD 3 Topology + Default Topology

Layer 3 Link
Layer 2 CE
Layer 2 FabricPath

L3

Pod 1
VLANs 100-199
VLANs 2000-2099

Pod 2
VLANs 200-299
VLANs 2000-2099

Pod 3
VLANs 300-399
VLANs 2000-2099

Layer 2 FabricPath

Pod-local VLANs exist only in POD switches, mapped to POD-specific topology.

Pod-local VLANs exist only in POD switches, mapped to POD-specific topology.
Multi-Pod Design
FabricPath Multi-Topology (NX-OS 6.2)

- Allows more elegant DC-wide versus POD-local VLAN definition/isolation
  - No need for POD-local VLANs to exist in core
  - Can support VLAN ID reuse in multiple PODs

- Define FabricPath VLANs → map VLANs to topology → map topology to FabricPath core port(s)
  - Depending on design, option exists to use a single non-default topology for all PODs (“disconnected” topology)

- Default topology always includes all FabricPath core ports
  - Map DC-wide VLANs to default topology

- POD-local core ports also mapped to POD-local topology
  - Map POD-local VLANs to POD-local topology
Agenda

- Introduction to FabricPath
- FabricPath Forwarding
- FabricPath Design
- Key Takeaways
Key Takeaways
Key Takeaways – FabricPath Technology

- **FabricPath is simple**
  - Keeps the attractive aspects of Layer 2 – No addressing, simple configuration and deployment
  - Integrates stability and scale of Layer 3 – Frame routing, TTL, RPF check

- **FabricPath is efficient**
  - High bisectional bandwidth (ECMP)
  - Optimal path between any two nodes

- **FabricPath is scalable**
  - Can extend a bridged domain without extending the risks generally associated with Layer 2
Key Takeaways – FabricPath Design

- You can deploy FabricPath today, with traditional network designs
- FabricPath introduces immediate, tangible benefits to any design:
  - Simple configuration, eliminate Spanning Tree, leverage parallel network paths, extend VLANs safely, mitigate loops, etc.
- Provides multiple design options to help you build a network that meets your requirements
Conclusion

- Thank you for your time today!
- You should now have a thorough understanding of FabricPath concepts, technology, and design considerations!
- Any questions?
Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Cisco Daily Challenge points for each session evaluation you complete.
- Complete your session evaluation online now through either the mobile app or internet kiosk stations.

Maximize your Cisco Live experience with your free Cisco Live 365 account. Download session PDFs, view sessions on-demand and participate in live activities throughout the year. Click the Enter Cisco Live 365 button in your Cisco Live portal to log in.