Cisco FabricPath
Technology and Design

Max Ardica
Technical Leader

Dubrovnik, Croatia, South East Europe
20-22 May, 2013
FabricPath
The Requirements
Cisco DC Fabric
Evolution of the Intelligent Layer 2 Domain POD

Shipping Nexus 7k

OTV
- Inter-POD Connectivity across L3
- Failure Boundary Preservation

LISP
- VM mobility

STP+
- STP Enhancements
- Bridge Assurance

vPC
- NIC Teaming
- Simplified loop-free trees
- 2x Multi-pathing

FabricPath
- 16x ECMP
- Low Latency / Lossless
- MAC Scaling

Shipping Nexus 5k/6k/7k
Cisco DC Fabric
Why Extended Layer 2 in the Data Center?

Because customers request it!

- Some protocols rely on the functionality
- Simple, almost plug and play
- No addressing
- Required for implementing subnets
- Allows easy server provisioning
- Allows virtual machine mobility
- Application clustering support
Cisco DC Fabric
Introducing FabricPath

FabricPath brings Layer 3 routing benefits to flexible Layer 2 bridged Ethernet networks

- Easy Configuration
- Plug & Play
- Provisioning Flexibility

Switching

- Multi-pathing (ECMP)
- Fast Convergence
- Highly Scalable

Routing

"FabricPath brings Layer 3 routing benefits to flexible Layer 2 bridged Ethernet networks"
FabricPath, an Ethernet Fabric

Shipping Since 2010: turn your network into a Fabric

- Connect a group of switches using an **arbitrary** topology
- With a simple CLI, aggregate them into a Fabric:

  ```
  N7K(config)# interface ethernet 1/1
  N7K(config-if)# switchport mode fabricpath
  ```

- An open protocol based on Layer 3 technology provides Fabric-wide intelligence and ties the elements together
Why Migrate from vPC to FabricPath?
Flexible Topologies Support

Traditional Access/Aggregation

Folded CLOS

Three Tiers (Fat Tree)

Full Mesh
FabricPath
The Technology
**FabricPath**

**Terminology**

- Send/receive FabricPath frame
- No STP, no MAC learning, no MAC address table
- Using a routing table computed by IS-IS

### Classical Ethernet (CE)

- Send/receive regular Ethernet frames
- Run STP, do MAC address learning using a MAC address table
New Control Plane
Plug-n-Play L2 IS-IS Manages Forwarding Topology

- IS-IS assigns addresses to all FabricPath switches automatically
- Compute shortest, pair-wise paths
- Support equal-cost paths between any FabricPath switch pairs

FabricPath Routing Table

<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>L1</td>
</tr>
<tr>
<td>S20</td>
<td>L2</td>
</tr>
<tr>
<td>S30</td>
<td>L3</td>
</tr>
<tr>
<td>S40</td>
<td>L4</td>
</tr>
<tr>
<td>S200</td>
<td>L1, L2, L3, L4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>S400</td>
<td>L1, L2, L3, L4</td>
</tr>
</tbody>
</table>

New Control Plane
Plug-n-Play L2 IS-IS Manages Forwarding Topology

- IS-IS assigns addresses to all FabricPath switches automatically
- Compute shortest, pair-wise paths
- Support equal-cost paths between any FabricPath switch pairs
The association MAC address/Switch ID is maintained at the edge.

Traffic is encapsulated across the Fabric.
FabricPath Encapsulation
16-Byte MAC-in-MAC Header

Classical Ethernet Frame

FabricPath Frame

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
# FabricPath Hardware Support
## Nexus 7000

<table>
<thead>
<tr>
<th>I/O Modules</th>
<th>CE Edge Port</th>
<th>FP core port</th>
<th>Routing for FP VLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>N7K-M108X2-12L</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>N7K-M132XP-12(L)</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>N7K-M148GS-11(L)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N7K-M148GT-11(L)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N7K-M224XP-23L</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>N7K-M206FQ-23L</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>N7K-M202CF-22L</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N7K-F132XP-15</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>N7K-F248XP-25*</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>N7K-F248XT-25E**</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>N7K-F248XP-25E**</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

*F2 module needs to be in their own VDC or system. It is not possible to mix F1 and F2 LC in the same VDC

**F2E module supported in proxy-routing mode (mixed in a VDC with Mx cards) from 6.2 release. No mix of F1 and F2E LC in the same VDC
## FabricPath Hardware Support

### Nexus 5000, 5500 & 6000

<table>
<thead>
<tr>
<th>Switch</th>
<th>CE Edge Port</th>
<th>FP Core Port</th>
<th>Routing for FP VLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nexus 5010</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Nexus 5020</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Nexus 5548P/UP*</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Nexus 5596UP/5596T*</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Nexus 6004 Nexus 6001</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* L3 daughter card is ‘not’ needed in 5550 to run FabricPath, only L3 routing for FabricPath VLANs.
The Nexus 7000 features two kinds of IO Modules: M series and F series.

M1/M2 I/O Modules cannot switch FabricPath traffic.

When running FabricPath, FP Core and CE Edge ports must be on an F module or 5500/6000.

New FabricPath/CE locally significant VLAN mode:

```plaintext
Leaf1(config)# vlan 10
Leaf1(config-vlan)# mode ?
    ce    Classical Ethernet VLAN mode
    fabricpath Fabricpath VLAN mode

Leaf1(config-vlan)# mode fabricpath
Leaf1(config-vlan)#
```

FabricPath VLANs can only be enabled on F modules (FEX+F2 with NX-OS 6.0 release and FEX+F2E with NX-OS release 6.1 as well) or 5500/6000 (5500/6000 + FEX as well)
FabricPath Key Concept
Multi-destination Trees

- Multi-destination traffic constrained to loop-free trees touching all FabricPath switches
- Root switch elected for each multi-destination tree in the FabricPath domain
- Loop-free tree built from each Root assigned a network-wide identifier (Ftag)
- Support for multiple multi-destination trees provides multipathing for multi-destination traffic

Two multi-destination trees currently supported in NX-OS
Multi-destination Trees
Role of the Ingress FabricPath Switch

- Ingress FabricPath switch determines which tree to use for each flow
- Other FabricPath switches forward based on tree selected by ingress switch
- Broadcast and unknown unicast typically use first tree
- Hash-based tree selection for IP multicast, with several configurable hash options
FabricPath Key Concept
Unknown Unicast Flooding

FabricPath Key Concept
Unknown Unicast Flooding

S100: CE MAC Address Table

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

S200: CE MAC Address Table

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

S300: CE MAC Address Table

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>e1/2</td>
</tr>
<tr>
<td>A</td>
<td>S100</td>
</tr>
</tbody>
</table>

Lookup B: Miss
Flood

Lookup B: Miss
Don't learn

Lookup B: Hit
Learn source A
FabricPath Key Concept
Known Unicast, Conversational MAC Learning

S100: CE MAC Address Table

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/1</td>
</tr>
<tr>
<td>B</td>
<td>S300</td>
</tr>
</tbody>
</table>

S200: CE MAC Address Table

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

S300: CE MAC Address Table

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>e1/2</td>
</tr>
<tr>
<td>A</td>
<td>S100</td>
</tr>
</tbody>
</table>

S300: FabricPath Routing Table

Switch  | IF |
---------|----|
         | ... |
         | ... |
S100     | L1, L2, L3, L4 |

Look up A: Hit
Send to S100

Lookup A: Hit
Learn source B

Conversational Learning
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>e1/1 (local)</td>
</tr>
</tbody>
</table>

**Multi-destination Trees on S10**

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

**Multi-destination Trees on S100**

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

**Multi-destination Trees on S300**

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

**Ftag**

- Broadcast to Host A
- Ftag to Host B
- Ftag to Host C

**DA**

- FF to Host A
- FF to Host B
- FF to Host C

**SA**

- 100 to Host A
- 200 to Host B
- 300 to Host C

**SMAC**

- A to Host A
- A to Host B
- A to Host C

**DMAC**

- FF to Host A
- FF to Host B
- FF to Host C

**payload**

- Packet data

**Putting it all together – Host A to Host B**

(1) Broadcast ARP Request

- Host A broadcasts ARP request on e1/1
- Ftag is added
- Packet is broadcast on all interfaces

(2) MAC Table on S100

- MAC A is added to MAC table
- No MAC entries for other hosts

(3) MAC Table on S100

- MAC A is added to MAC table
- MAC B is added to MAC table
- No MAC entries for other hosts

(4) MAC Table on S100

- MAC A is added to MAC table
- MAC B is added to MAC table
- No MAC entries for other hosts

(5) MAC Table on S300

- MAC A is added to MAC table
- MAC B is added to MAC table
- No MAC entries for other hosts

(6) MAC Table on S300

- MAC A is added to MAC table
- MAC B is added to MAC table
- No MAC entries for other hosts

Learn MACs of directly-connected devices unconditionally

Don’t learn MACs from flood frames
Putting it all together – Host A to Host B

(2) Broadcast ARP Reply

Multi-destination Trees on Switch 10

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
<th>MAC A</th>
<th>MAC B</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>po100,po200,po300</td>
<td>Root for Tree 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>po100</td>
<td>S10</td>
<td></td>
</tr>
</tbody>
</table>

Multi-destination Trees on Switch 100

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
<th>MAC A</th>
<th>MAC B</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>po10</td>
<td>Root for Tree 2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>po10</td>
<td>S20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>po10,po20,po30,po40</td>
<td>S30</td>
<td></td>
</tr>
</tbody>
</table>

Multi-destination Trees on Switch 300

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
<th>MAC A</th>
<th>MAC B</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>po10,po20,po30,po40</td>
<td>Root for Tree 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>po40</td>
<td>S40</td>
<td></td>
</tr>
</tbody>
</table>

FabricPath MAC Table on S100

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/1 (local)</td>
</tr>
<tr>
<td>B</td>
<td>300 (remote)</td>
</tr>
</tbody>
</table>

FabricPath MAC Table on S300

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/1 (local)</td>
</tr>
<tr>
<td>B</td>
<td>e1/2 (local)</td>
</tr>
</tbody>
</table>

If DMAC is known, then learn remote MAC

Putting it all together – Host A to Host B

(2) Broadcast ARP Reply

Multi-destination Trees on Switch 10

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
<th>MAC A</th>
<th>MAC B</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>po100,po200,po300</td>
<td>Root for Tree 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>po100</td>
<td>S10</td>
<td></td>
</tr>
</tbody>
</table>

Multi-destination Trees on Switch 100

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
<th>MAC A</th>
<th>MAC B</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>po10</td>
<td>Root for Tree 2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>po10</td>
<td>S20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>po10,po20,po30,po40</td>
<td>S30</td>
<td></td>
</tr>
</tbody>
</table>

Multi-destination Trees on Switch 300

<table>
<thead>
<tr>
<th>Tree</th>
<th>IF</th>
<th>MAC A</th>
<th>MAC B</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>po10,po20,po30,po40</td>
<td>Root for Tree 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>po40</td>
<td>S40</td>
<td></td>
</tr>
</tbody>
</table>

FabricPath MAC Table on S100

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/1 (local)</td>
</tr>
<tr>
<td>B</td>
<td>300 (remote)</td>
</tr>
</tbody>
</table>

FabricPath MAC Table on S300

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/1 (local)</td>
</tr>
<tr>
<td>B</td>
<td>e1/2 (local)</td>
</tr>
</tbody>
</table>

If DMAC is known, then learn remote MAC
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>...</td>
<td>...</td>
</tr>
<tr>
<td>S300</td>
<td>po30</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>S300 po10, po20, po30, po40</td>
</tr>
</tbody>
</table>

FabricPath MAC Table on S100

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF/SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>e1/1 (local)</td>
</tr>
<tr>
<td>B</td>
<td>300 (remote)</td>
</tr>
</tbody>
</table>

If DMAC is known, then learn remote MAC
FabricPath Design Considerations
vPC+ at the Edge

- Using vPC+ at the edge enables several options
  - Attaching servers with Port-channels
  - Attaching other Classic Ethernet Switches in vPC mode
  - Attaching FEX in Active/Active mode
- Each pair of Nexus 5500 configured for vPC+ runs FabricPath with the spine AND on the vPC peer-link too
- The vPC domain is advertised as a different Switch-id (Emulated switch)
- The area highlighted (i.e. its MAC addresses) are advertised as belonging to the “Emulated switch”
vPC+ at the Edge

Configuration

- Leaf devices advertise MAC_A and MAC_B of hosts attached in vPC+ mode as associated to the emulated switch-ID 1000
- North-to-south traffic directed to the hosts has two equal cost paths available

Leaf1(config)# vpc domain 1
Leaf1(config-vpc-domain)# fabricpath switch-id 1000

Spine1# sh fabricpath route
FabricPath Unicast Route Table
'a/b/c' denotes ftag/switch-id/subswitch-id
'[x/y]' denotes [admin distance/metric]

<snip>
1/1000/0, number of next-hops: 2
  via Eth1/1, [115/40], 1 day/s 04:12:00, isis_fabricpath-default
  via Eth1/2, [115/40], 1 day/s 04:12:00, isis_fabricpath-default

vPC+ configuration applied to both FP Leaf devices
FabricPath Design Considerations
vPC+ at the Spine (Dual Active Gateway)

- It is possible to distribute routed traffic to both spines by using vPC+
  - Spines function as HSRP Active/Standby on the control plane, Active/Active on the data plane
  - The HSRP MAC is advertised with the same Emulated Switch-Id to all leaf devices

- Each edge switch has an equal cost path to the Emulated Switch ID (via both spine devices)

- All you need to do on the spines is to configure a vPC domain and a peer-link
  - No need to define any vPC port

Leaf1# show mac address-table vlan 101
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</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>0000.0c07.ac01</td>
<td>dynamic</td>
<td>0</td>
<td>F</td>
<td>F</td>
<td>2000.0.1054</td>
</tr>
</tbody>
</table>

Leaf1# sh fabricpath route switchid 2000
<snip>
FabricPath Unicast Route Table for Topology-Default
1/2000/0, number of next-hops: 3
  via Eth1/1, [115/40], 1 day/s 09:38:15, isis_fabricpath-default
  via Eth1/2, [115/40], 1 day/s 09:38:15, isis_fabricpath-default
IGMP Snooping in FabricPath

- IGMP snooping learns of interested receivers on FabricPath edge switches
- Membership tracked on CE ports based on receiving IGMP reports/leaves
  - Only locally connected receivers tracked on a given edge switch
- Group membership advertised in FabricPath IS-IS using GM-LSPs
FabricPath IP Multicast Control Plane
IS-IS Creates Pruned Forwarding Trees Using GM-LSPs

**Multidestination Tree 1**
- G1 Pruned Tree
- G2 Pruned Tree

**Multidestination Tree 2**
- G1 Pruned Tree
- G2 Pruned Tree
FabricPath

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, VLAN re-use), traffic engineering, security etc…
- 2 topologies currently supported on Nexus 5500 (from release 5.2(1))
  - 2 multi-destination trees are supported for each topology
- 2 topologies supported on Nexus 7000 starting from release 6.2 (Q2CY13)
Multi-Topology Support
Understanding VLANs and FabricPath Topologies

```
spine1(config)# show fabricpath topology vlan
Topo-Description Topo-ID Configured VLAN List
----------------- ---------- ------------------
0     0           1-4095
```

Topology 0
- Tree 1
  - S1
  - S300
  - S400
  - S2

Topology 0
- Tree 2
  - S1
  - S300
  - S400
  - S2

Topology 1
- Tree 3
  - S10
  - S2

Topology 1
- Tree 4
  - S20
  - S1

```
s100(config)# fabricpath topology 1
s100(config-fp-topology)# member vlan 10
s100(config)# int po1
s100(config-if)# fabricpath topology 1

s100(config-fp-topology)# sh fabricpath topology vlan
Topo-Description Topo-ID Configured VLAN List
----------------- ---------- ------------------
0     0           1-9, 11-4095
1     1           10
```
**Single Topology Design**

FabricPath Topology

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

CORE
VLANs 100-199
VLANs 200-299
VLANs 300-399
VLANs 2000-2099
Multi-Topology Design
Localized Forwarding

- Core ports in POD belong to default topology and also mapped to POD-local topology.
- Only DC-wide VLANs exist in FabricPath core.
- Core ports belong only to default topology.

- POD 1:
  T1: VLANs 10,100-199
  T0: VLANs 2000-2099

- POD 2:
  T1: VLANs 10, 200-299
  T0: VLANs 2000-2099

- POD 3:
  T1: VLANs 10,300-399
  T0: VLANs 2000-2099

- POD-local VLAN 10 can be re-used between pods. VLAN 10 is unique to each POD.
- POD-local VLANs exist only in POD switches, mapped to POD-specific topology.

- Layer 2 FP Default Topology
- POD 1 Topology + Default Topology
- POD 2 Topology + Default Topology
- POD 3 Topology + Default Topology
FabricPath and STP
FabricPath Simple STP Interaction

- FP leaves send and receive STP BPDUs on CE edge ports
  No BPDUs is sent out of FP core port links
- FP leaves must be configured as STP root for all VLANs configured on vPC+ member ports
- All FP switches in the FP domain have the same STP bridge ID: c84c.75fa.6000
- There is no need to explicitly configure peer-switch in vPC+; both vPC+ peer devices can send/receive BPDU to/from CE access switches
FabricPath and STP

Root-Guard

- CE edge ports (i.e vPC+ member ports) have FabricPath root guard-like function enabled implicitly
- If superior BPDU received on edge port, port is placed in “L2 Gateway Inconsistent” state until condition cleared

%STP-2-L2GW_BACKBONE_BLOCK: L2 Gateway Backbone port inconsistency blocking port port-channel100 on VLAN0100
FabricPath and STP
Setting STP Priority on FP Leaves

- When connecting a Classical Ethernet (CE) switch to FP Leaves, pay attention to STP priority on vPC+ domain
- STP priority for the VLAN on FP leaves must be better than on CE access switch
  
  Recommendation is to decrease STP priority value on FP leaves
  Configure pseudo-information just to cover scenarios where the CE devices are attached in STP mode (no vPC)
FabricPath
The Design
Routing at Aggregation
Natural Evolution of the vPC Design

- Evolutionary extension of current design practices
- Design 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
- Scalability considerations
  - Today: 16K unique host MACs across all routed VLANs
  - With NX-OS 6.2 release: 128K MACs with “proxy L2 learning” on M1/M2+F1/F2E (Nexus 7000)
  - 128K MACs with Nexus 6000

Check out the corresponding design guide:
Spine-Leaf Design
Changes to the Approach to Structured High Availability

Number of Spine Switches
Need for HA

Number of Spine Switches
Routing at Aggregation
Options to Scale-Out the Spine Layer

- Some polarization
- Inter-VLAN traffic can be suboptimal

- Host is pinned to a single gateway
  - Less granular load balancing

- All active
  - Available in NX-OS 6.2 release
Centralized Routing
Removing Routing from the FP Spine Layer

Centralized Routing Design
Alternate View

L3
Layer 3 services leaf switches

L2/L3 boundary
FabricPath

FabricPath spine
Server access leaf switches

= =

Centralized Routing Design
Alternate View

FabricPath spine
Layer 3 services border leaf switches
L2/L3 boundary
Run VPC+ for active/active HSRP

L3
All VLANs available at all leaf switches
FHRP between L3 services switches for FHRP
Cisco Unified Fabric Evolution
Leading The Data Center Flexibility and Innovation With Nexus

Distributed Control Plane
Extensible Network Topologies
Enhanced Forwarding

Integrated L2/L3 Across All Nodes

Simplified Management
Secure Multi-Tenancy
Automation and Simplicity
Additional Resources

http://www.cisco.com/go/fabricpath
Thank you.