TRILL for Service Provider Data Center and IXP

Francois Tallet, Cisco Systems
Agenda

TRILL: Transparent Interconnection of Lots of Links

- TRILL overview
- How TRILL works
- TRILL designs
- Conclusion
TRILL Goals

- IETF standard for Layer 2 multipathing
- Driven by multiple vendors

http://datatracker.ietf.org/wg/trill/

“TRILL brings Layer 3 routing benefits to flexible Layer 2 bridged Ethernet networks”
TRILL Overview
Connect a group of switches using an **arbitrary** topology
With minimal configuration, aggregate them into a fabric.
A control protocol based on Layer 3 technology provides fabric-wide intelligence and ties the elements together
Optimal, Low Latency Switching

- Single address lookup at the ingress edge identifies the exit port across the fabric
- Traffic is then switched using the shortest path available
- Reliable L2 connectivity any to any (as if it was the same switch, **no STP inside**)

<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>
<tr>
<td>B</td>
<td>s8</td>
</tr>
</tbody>
</table>
High Bandwidth, High Resiliency

Equal Cost MultiPathing (ECMP)

- Multipathing
- Traffic is redistributed across remaining links in case of failure, providing fast convergence
Layer 3 Integration
Can Route Anywhere

- The fabric provides seamless L3 integration
- An arbitrary number of routed interfaces can be created at the edge or within the fabric
How TRILL Works
New Control Plane
Plug-n-Play L2 IS-IS Manages Forwarding Topology

- IS-IS assigns an address (nickname) to all TRILL switches
- Compute shortest, pair-wise paths
- Support equal-cost paths between any TRILL switch pairs
New Data Plane

- The association MAC address/Nickname is maintained at the edge

• Traffic is encapsulated across the Fabric

Nickname space: Routing decisions are made based on the TRILL routing table

MAC address space: Switching based on MAC address tables

S300: TRILL Routing Table

<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>S100</td>
<td>L1, L2, L3, L4</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>1/2</td>
</tr>
<tr>
<td>A</td>
<td>S100</td>
</tr>
</tbody>
</table>
Unknown Unicast

Classical Ethernet

S100: CE MAC Address Table

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1/1</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>A</td>
<td>S100</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>A</td>
<td>S100</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Lookup B: Miss Flood

TRILL
Known Unicast

Classical Ethernet

S100: CE MAC Address Table

<table>
<thead>
<tr>
<th>MAC</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1/1</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>A</td>
<td>S100</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>A</td>
<td>S100</td>
</tr>
<tr>
<td>B</td>
<td>1/2</td>
</tr>
</tbody>
</table>

S300: TRILL Routing Table

<table>
<thead>
<tr>
<th>Switch</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S100</td>
<td>L1, L2, L3, L4</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Lookup A: Hit Learn source B

Lookup A: Hit Send to S100
Loop Mitigation with TRILL

Time To Live (TTL) and Reverse Path Forwarding (RPF) Check

- The control protocol is the only mechanism preventing loops
- If STP fails → loop
  - No backup mechanism in the data plane
  - Flooding impacts the whole network

- TTL in TRILL header
- RPF Check for multi-destination traffic
- The data plane is protecting against loops
Fine Grained Labeling
Uses 2 Q-Tag on the Fabric
TRILL Designs
Classical POD with TRILL

TRILL vs. STP/Distributed Port Channel (DPC)

- Simple configuration
- No constraint in the design
- Seamless L3 integration
- No STP, no traditional bridging
- Virtually unlimited bandwidth
- Can extend easily and without operational impact
TRILL Core

Efficient POD Interconnect

- TRILL in the Core
- VLANs can terminate at the distribution or extend between PODs.
- STP is not extended between PODs, remote PODs or even remote data centers can be aggregated.
- Bandwidth or scale can be introduced in a non-disruptive way
TRILL as Site Interconnect

- Requires dark fiber
- Arbitrary interconnect topology (not dependent of port channels)
- Any number of sites
- High bandwidth, fast convergence
- Spanning tree isolation
- VLANs can be selectively extended/terminated
Internet Exchange Point (IXP)

IXP Requirements
- Layer 2 Peering
- 10GE non-blocking Fabric
- Scale to thousands of ports

TRILL Benefits for IXP
- Layer 2 Fabric
- Non-blocking up to thousands 10GE ports
- Simple to manage
- No design constraint, easy to grow
TRILL Flexibility

The Network Can Evolve With No Disruption

- Need more edge ports? → Add more leaf switches
- Need more bandwidth? → Add more links and spines
Scaling with TRILL

Example: 2,048 x 10GE Server Design

- 16X improvement in bandwidth performance
- 6 to 1 consolidation (from 74 managed devices to 12 devices)
- 2X+ increase in network availability
- Simplified IT operations (fewer devices, vlans anywhere)
Conclusion
Key Takeaways

- TRILL is simple, keeps the attractive aspects of Layer 2
  - Transparent to L3 protocols
  - No addressing, simple configuration and deployment

- TRILL is efficient
  - High bi-sectional bandwidth (ECMP)
  - Optimal path between any two nodes

- TRILL is scalable
  - Can extend a bridged domain without extending the risks generally associated to Layer 2 (frame routing, TTL, RPFC)
Thank you