802.1aq: Review of Options for SPPBB Source-Tree ID Encoding

Don Fedyk & Ali Sajassi

dwfedyk@nortel.com    sajassi@cisco.com

IEEE meeting
November 13, 2007
Review Shortest Path Bridging

- Uses common mechanism for shortest path trees
  - Like MST but with a shortest path tree per bridge in a SPB Region
- Allows learning
- Introduces a Shortest Path VID per Bridge to identify the Source Tree Root.
- Introduce a Base VID
- Allows multiple control planes
  - SPB
  - Link State Protocols
Additional enhancements to Shortest Path Provider Backbone Bridging (SPPBB)

- We only consider Link State protocols
  - IS-IS is very suitable

- No Learning of provider addresses
  - All Provider addresses advertised by Link state protocol.
  - Complete unicast connectivity for every ingress PBB port to every destination PBB port.

- All Multicast Groups advertised by link state protocol
  - I-SID mapping to multicast groups
  - Link state driven multicast connectivity
Current Status

- We must make some progress on this project.
- We have considerable interest and technical content in applying SPB to 802.1ah (SPPBB)
  - Providers desire a robust and scalable control plane
- In order to go forward we need to have a common understanding of the technical issues.
- We have a white paper laying out the SPPBB options
We are making progress
Lots of alignment

<table>
<thead>
<tr>
<th>Feature</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN Topology</td>
<td>All support shortest path Trees</td>
</tr>
<tr>
<td>VLAN Partitioning</td>
<td>All use a logical B-VLAN</td>
</tr>
<tr>
<td>Link state topology</td>
<td>All use IS-IS</td>
</tr>
<tr>
<td>No Learning Mesh Networking</td>
<td>All use IS-IS to populate FIB</td>
</tr>
<tr>
<td>Forwarding: backwards compatibility</td>
<td>Two use a VID+DMAC context (not option3)</td>
</tr>
<tr>
<td>Control plane objects</td>
<td>Similar requirements</td>
</tr>
<tr>
<td>SPT computation</td>
<td>Similar requirements</td>
</tr>
<tr>
<td>Multicast Groups</td>
<td>Support Via IS-IS</td>
</tr>
<tr>
<td>Multicast and Unicast Congruency</td>
<td>Aligned</td>
</tr>
<tr>
<td>Forward &amp; Reverse Path Congruency</td>
<td>Aligned</td>
</tr>
<tr>
<td>Number of Trees for Unicast Forwarding</td>
<td>All use one tree per source BEB</td>
</tr>
<tr>
<td>Number of Trees for Multicast Forwarding</td>
<td>All use one per (S,G)</td>
</tr>
<tr>
<td>Multicast Trees</td>
<td>All use pruning of the broadcast source tree</td>
</tr>
<tr>
<td>Multicast Groups</td>
<td>All can use Groups to represent multiple I-SIDs</td>
</tr>
<tr>
<td>Single path per VID to a destination</td>
<td>Aligned No ECMP</td>
</tr>
<tr>
<td>Ingress Check</td>
<td>All support ingress check</td>
</tr>
<tr>
<td>MRP</td>
<td>All can use IS-IS for MRP equivalence</td>
</tr>
</tbody>
</table>
But, We have one main issue

- Need to identify the source tree in the Ethernet header for 802.1aq for two reasons:
  - We need a packet context that identifies the source for:
    1. To perform ingress check at each node to ensure that the right frames arrive at the right interface (Loop Mitigation)
    2. Forwarding context: To get (S,G) information for forwarding of a multicast frame
- This is the subject of the white paper
Three Solution Options

1. Use VID to identify the source-tree
2. Use part of Multicast MAC-DA to identify the source tree
3. Use part of MAC-SA to identify the source tree

Note: Option 1 is applicable to both 802.1Q and 802.1ah bridges; whereas, option 2 and option 3 are applicable to only 802.1ah bridges.
Option-1: Use VID to identify Source Tree

- Use a single VID per edge bridge to identify the source bridge (or source tree)

Two Main Issues:
- It consumes lots of VIDs - one VID per bridge
- It consumes too many TCAM entries (unless hw is modified as (b) below)

It requires the following modification to existing bridges
a. Have a two-bit VID vector per port for ingress checking because besides checking whether a VID is allowed on a port or not, need to also check the direction (ingress v.s. egress) for that port in order to support unidirectional tree
b. On a per VLAN basis to be able to perform IVL check for mcast data & SVL check for unicast data based on U bit in order to avoid replicating unicast B-MACs across different VIDs in TCAM
Option-1 Pros

- This scheme is backwards compatible with 802.1Q forwarding operations.
- Multicast (Source, Group) forwarding can be encoded as (SPVID, DMAC)
- The Ingress check can be performed on the VID since the SPVID represents the source tree root bridge.
- (*,G) encoding of the multicast DA is common with 802.1ah
- It allows for the separation of 802.1aq domains under the same I-SID space – e.g., when a single provider has several 802.1ah islands operating under the same I-SID space. With this approach, no multicast B-MAC translation is required at the domain boundaries.
- This approach allows for an administratively consistent structured set of B-MAC addresses across different domains by using the Local Admin capability.
- Only one lookup is needed to perform both ingress check and forwarding – no need to lookup MAC SA (for 802.1ah)
Option-1 Cons

- The VID space is limited. SPVIDs are consumed at a rate of 1 per shortest path tree per bridge. If several equal shortest path trees are computed per bridge, the number of shortest path trees that can be uniquely identified drops significantly.
  - # of nodes in the network is limited by 4K/VLANs for 802.1Q bridges. For 802.1ah, the situation is somewhat better since a single VLAN can be used; therefore # of nodes is limited to 4K.
  - SPVID may limit the number of B-VID topologies that could be used for other applications such as PBB-TE.
  - Equal Cost Multiple Trees reduces the number of SPVIDs available if used.
- The unicast entries need to be replicated across all the VIDs unless hardware modification (b) is done. However, with such modification, learning of mcast cannot be applied to unicast because of different FDs.
Option-2: Use Part of MAC-DA to Identify Source Tree

- Only applies to 802.1ah with administered B-MAC addresses
- Modify Multicast MAC-DA only and leave MAC SA alone (put source root bridge ID in the MAC DA (Uses local admin bit))
- Unicast addresses are unaffected
Option-2: Use Part of MAC-DA to Identify Source Tree – Cont.

- **2**: Backbone Destination Address (B-DMAC)
- **6**: OUI 1st Byte
- **10**: OUI 2nd Byte
- **14**: OUI 3rd Byte
- **18**: Encap Ethertype
- **22**: .1ad B-TAG TCI/VID
- **26**: .1ad I-TAG TCI/SID
- **30**: Destination Address (C-DMAC)
- **34**: Source Address (C-SMAC)
- **38**: Encap Ethertype

**Table:**

<table>
<thead>
<tr>
<th>OUI 1st Byte</th>
<th>OUI 2nd Byte</th>
<th>OUI 3rd Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
</tr>
</tbody>
</table>

**Diagram:**

- **B-DMAC**
  - M L Flags
  - SA Nickname
  - Group-Identifier
Option-2 Pros

- A full set of shortest path trees can be achieved using a single VID. (One logical or BASE VLAN). Another VID may be used for ECMT if load spreading is desired. By using a single VID for a whole set of shortest path trees, we have preserved the typical bidirectional nature of VIDs.

- For Unicast traffic a single forwarding entry (shared forwarding) is used for all shortest path to a destination, which scales O(N). This has tremendous scalability over the other options particularly for large meshes. In essence, with shared forwarding the VID source/destination pair for unicast becomes only a single VID + destination pair for all unicast traffic.

- Bidirectionality of the VID is preserved.
  - Similar to existing 802.1ag procedures for MIPs since the VID is common for request and response functions.

- B-VID allocation is independent from number of bridges in the network.

- It can theoretically scales to more than 4K bridges.
Option-2 Cons

- This application of the locally assigned address bit for multicast must be standardized. The scope of these addresses is only within the PBB domain.

- All multicast addresses take the local bit mapping. While being transported in the PBB domain. Global multicast DMACs would have an equivalent group mapping. For example the PBB multicast OUI is not supported but a locally assigned multicast is functionally the same as the PBB OUI.
  - It prevents the use of global structured Multicast MAC addresses
  - You cannot have unspecified multicast addresses which can get broadcasted – but not an issue w/ 802.1ah

- Multicast addresses are of the form (S,G) where both S and G are encoded in the DMAC

- An Ingress Check on SA may have an impact on forwarding lookups depending on specific hardware.
Option-3: Use Part of MAC-SA to Identify Source Tree

- Only applies to 802.1ah with administered B-MAC addresses
- Modify MAC-SA only (put source bridge ID in the MAC SA)
- It DOES require significant changes to existing bridges acting as intermediate nodes (e.g., BCB) – for (S,G) mcast data lookup, bridge need to perform the lookup based on B-VID + MAC-SA + MAC-DA (group info)
Option-3 Pros

- All the advantages of option 2 plus the followings:
- No need for a protocol to encode bridge ID in the frame
- No need for having administrative multicast B-MAC addresses & global MAC address format
Option-3 Cons

- Forwarding on SMAC for multicast data is a new operation.
- The SMAC will take up additional space in the Multicast forwarding tables. An effective 84-bit (worst case 108-bit) lookup must be performed instead of 60-bit lookup.
- An Ingress Check on SA may have an impact on forwarding lookups depending on specific hardware.
### Summary

<table>
<thead>
<tr>
<th>Source Root Tree Identifier</th>
<th>Solution 1: SPVID</th>
<th>Solution 2: Logical VID with Source encoded DMAC</th>
<th>Solution 3: VID+SMAC +DMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VID Usage</strong></td>
<td>(1 per Bridge) x (# of ECMT/BASE VID)</td>
<td>(1 per BASE VID) x (# of ECMT/BASE VID)</td>
<td>(1 per BASE VID) x (# of ECMT/BASE VID)</td>
</tr>
<tr>
<td><strong>Unicast Forwarding</strong></td>
<td>VID+DMAC</td>
<td>VID+DMAC</td>
<td>VID+DMAC</td>
</tr>
<tr>
<td><strong>Multicast Forwarding</strong></td>
<td>VID+DMAC</td>
<td>VID+DMAC</td>
<td>VID+SMAC+DMAC</td>
</tr>
<tr>
<td><strong>Unicast Forwarding</strong></td>
<td>1 entry per # of Unicast Destination x BASE VIDs x # of SPVIDs</td>
<td>1 entry per # of Unicast Destinations x # BASE VIDs</td>
<td>1 entry per # of Unicast Destinations x # BASE VIDs</td>
</tr>
<tr>
<td><strong>Multicast FIB size</strong></td>
<td>1 entry per Source Tree /Multicast DMAC</td>
<td>1 entry per Source Tree /Multicast DMAC</td>
<td>1 entry per Source Tree / Multicast DMAC</td>
</tr>
<tr>
<td><strong>Maximum Flat Network</strong></td>
<td>4000 Bridges/((# of ECMT/BASE VID) * (# of BASE VID))</td>
<td>10,000+ Bridges Limited only by FIB entries and Link State</td>
<td>10,000+ Bridges Limited only by FIB entries and Link State</td>
</tr>
</tbody>
</table>
### Summary Cont.

<table>
<thead>
<tr>
<th># of Active topologies</th>
<th>Low Each Base VID Group reduces the number of SPVIDs</th>
<th>High Each BASE VID consumes 1 VID Comparable to B-VID</th>
<th>High Each BASE VID consumes 1 VID Comparable to B-VID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingress Check</td>
<td>SPVID or SMAC</td>
<td>SMAC</td>
<td>SMAC</td>
</tr>
<tr>
<td># of Lookups (ingress check + forwarding)</td>
<td>1 (0+1)</td>
<td>2 (1+1)</td>
<td>2 (1+1)</td>
</tr>
<tr>
<td>Global representation of B-MACs</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Need Administrative Addresses</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Additional Protocols for unique node-id assignmnet</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CFM Aggregation</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Decisions Perspective

Ethernet Bridges

Provider Backbone Bridges only

Solution Option 1

Solution Option 2

Solution Option 3

Change to Ethernet Forwarding (YMMV)
(Increasing probability of hardware change)

Scalability

Ingress Check SA
SPVID

Source Specific MAC DA

Ingress Check SA

Multicast Forward On DA in SA context

All: IS-IS equivalent for MRP

Lower

Equal Cost Multiple Tree Impact

Higher

11/13/2007

IEEE 802.1aq November Plenary

YMMV = Your mileage may vary
Discussion

- How to resolve this?