SPBB – emulated LAN segment (VPLS) interworking

Glenn Parsons, David Allan, Nigel Bragg
May 2009
Some SPBB Interworking Scenarios

N NI Interworking with VPLS
• PBBN with MSTP $\leftrightarrow$ B-tagged NNI $\Rightarrow$ VPLS with LDP

N NI Overlay with VPLS
• PBBN with SPBB $\leftrightarrow$ B-tagged NNI $\Rightarrow$ VPLS with LDP

N NI Interworking with PBN/PBBN
• PBBN with SPBB $\leftrightarrow$ I-tagged NNI $\Rightarrow$ PBBN with SPBB
• PBN with SPB $\leftrightarrow$ S-tagged NNI $\Rightarrow$ PBN with MSTP
• PBBN with SPBB $\leftrightarrow$ B-tagged NNI $\Rightarrow$ PBBN with MSTP

UNI Interworking
• PBBN with MSTP $\leftrightarrow$ I-tagged UNI $\Rightarrow$ VPLS with LDP
• PBBN with SPBB $\leftrightarrow$ S-tagged UNI $\Rightarrow$ VPLS with LDP
• PBBN with SPBB $\leftrightarrow$ S-tagged UNI $\Rightarrow$ PBN with SPB
• PBBN with SPBB $\leftrightarrow$ S-tagged UNI $\Rightarrow$ PBN with MSTP

The first & second scenarios are considered here.
Problem Statement

- PBB is endorsed by IETF L2VPN WG as an “access network” for a VPLS core network offering virtualised LAN emulation
  - It improves scalability since VPLS has only visibility of B-MACs, C-MACs are learned only in the PBB nodes
- PBBN-VPLS interworking is different from PBN-VPLS….
  - It is a VPN of VPNs
  - Need for “per-service” multicast containment
  - Resilient and efficient interconnect
  - Large number of I-tagged services are aggregated onto any NNI
- Current arrangement is with MSTP control plane on PBBN or single subtending BEBs
  - Isolate and interconnect active topologies with VPLS
- The SPBB control plane will scale well enough (target is 1000 nodes in a single area) to permit overlay of LAN segments and emulated LANs
Scaling VPLS using PBBN with MSTP

To maximize scalability

1. Peer at the B-component (and not I-component)
   - Appropriate filtering of B-component multicast addresses is sufficient to isolate services
   - Elimination of the need for hosting I-components in VPLS PEs eliminates need for any PBB specific LDP signaling enhancements

2. This requires a single emulated LAN for all services carried by the PBBN
   - Operational decoupling of service provisioning from infrastructure when VPLS used for PBBN transit

3. Use multicast MAC filtering at ingress to the emulated LAN for multicast efficiency
   - Dynamic registration using 802.1ak MMRP is one option

4. Eliminates requirement for LDP “MAC withdraw” messaging
   - Current practice for selective invalidation of MAC entries in VPLS PEs

   • Resiliency capabilities in a scaled environment should include handling the severing of a subtending MSTI domain
Operational Model - VPLS ⇄ PBBN / MSTP

1. PBBN/VPLS locally selects active uplink

2. LDP preferential forwarding status bit selects PW to tie together active uplinks (independent mode)

3. MMRP registrations for I-SID multicast MAC program filtering

Well known multicast MACs

<table>
<thead>
<tr>
<th>802.1ah OUI</th>
<th>I-SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>22 bits</td>
<td>24 bits</td>
</tr>
</tbody>
</table>
Operational Model - VPLS \(\leftrightarrow\) PBBN / MSTP

4. Far end Change in active uplink signaled by change of active/standby status

5. Change from active to standby triggers flush of MACs mapped to that PW, and resets VPLS facing MMRP filtering

6. Change from standby to active triggers generation of VPLS facing MMRP registration messages
We can go further with a PBBN avec 802.1aq

• We will keep
  1. Peering at the B-component layer
  2. A single pre-provisioned PW mesh for all services carried by the PBBN with SPBB
  3. Use of multicast MAC filtering at ingress to LAN emulation for multicast efficiency
     • But can also drive it directly from IS-IS in an integrated model

• And we can
  • Eliminate requirement for LDP participation in the active topology entirely
     • and so achieve $O(N)$ messaging load for any change in the attachment configuration
       • no need to inform all VPLS service endpoints
There are two Implementation Models

- The “Arms length” model
  - VPLS PE has no knowledge of PBBNs
  - VPLS VSI FDB is populated by traditional flooding, learning & registration

- The “Integrated” model
  - VPLS PE is also a PBB/SPBB BCB
  - VPLS VSI FDB is populated by the SPBBB control plane

The “arms length” model represents inter-working between standards, and so is the model considered
A key property of SPBB NNIs

- SPBB can fully utilize a physical mesh

Traditionally you would have needed to block one of these links to avoid looping
Another key property of SPBB NNIs

- SPBB can coordinate the PBBN side of VPLS points of attachment, IS-IS simply overlays LAN emulation
- SPBB loop avoidance works across LAN emulation

**Full mesh of LDP adjacencies to coordinate VPLS end points and elect a single PW is not required**
Extending this to interconnect multiple PBBNs

- Provision two parallel meshes = two parallel LAN segments
  - Each singly homed on each PBBN
    - each ¼ of the PWs of the MSTP solution
  - Invariant PW meshes eliminate need for LDP “MAC withdraw” or “preferred forwarding” handshaking
    - PWs are not an “active” component
Routing - VPLS as transit LAN segment

- LAN segments have existed “forever” and are already accommodated by link-state control plane architectures:
  - There are already data-plane models for use in SPF calculations;
  - The control plane architecture collapses the number of adjacencies from $O(N^2)$ to $O(N)$;

![Physical Network](image1)
![Link State Topology View](image2)
Filtering

- We can surround a LAN segment (VPLS) with filtering to teach it what the SPBB control plane wants:
  - it will naturally reconcile with the SPBB SPF computations
  - guaranteed symmetric congruence of go and return paths means that what SPBB wants to teach, the (emulated) LAN segment will learn….

- FDB requires an additional filtering option to avoid creation of duplicate frames when VPLS is “learning”
  - Port of arrival filtering for a DA as defined in 802.1ap is either permit one port or permit all ports
  - 802.1aq requires ability to define a valid port map for receipt of a DA
    - “some” ports … those on a valid shortest path to a DA in SPBB
Multicast - implementation

When overlaying a LAN segment, SPBB has a complete topology view:

- SPBB builds (S, G) trees to support shortest path routing;
- A LAN segment only requires (*, G) trees,
- but MMRP can be used to signal and build SPBB (S, G) trees:
  - they install full bi-directional state, but this is not harmful,
  - since only one SPBB source ever transmits on each tree.
Summary

- PBBN-VPLS interworking already embraced by industry due to scalability enhancements
  - Move to single PW mesh for PBBN interconnect
  - MMRP awareness
  - Reduced core MAC table consumption

- SPBB-VPLS interworking offers further simplification in the form of significantly reduced messaging load on any VPLS core
  - No need for “MAC withdraw” or “preferred forwarding” LDP message exchange as part of fault recovery

- PBBN-emulated LAN segment interworking requires
  1. Per PW multicast filtering
     • IETF already considering MMRP “awareness” for PBB-VPLS
  2. Addition port of arrival filtering options for the DA in BCBs
     • “PortList” syntax for FDB MIB object
Backup
Connection of SPBB nodes over VPLS (1)

- Set up ELAN services over VPLS (as many as needed, only 1 shown)
- SPBB SPF computation only installs forwarding state in the SPBB Gateway on the shortest path to any B-MAC DA reachable through it
  - the other Gateway blocks that DA, by “discard on unknown”, preventing more than one copy of a unicast packet entering the SPBB network.
- SPBB returns traffic by the shortest path only, “teaching” VPLS the route
Connection of SPBB nodes over VPLS (2)

- SPBB SPF computation only installs forwarding state in the SPBB Gateway on the shortest path to any B-MAC DA reachable through it
- VPLS can flood a B-MAC as unknown
- but SPBB does know where to find it
- With VPLS (*, G) multicast, and even under multi-homing, the ingress check prevents VPLS re-circulating frames into SPBB
Implications of single VPLS service

SPBB will work with one VPLS service dual-homed on SPBB “regions”

- *but* a failure requires both SPBB and VPLS to react

The network is not really re-converged after failure until both:

1. SPBB has converged
2. VPLS has done all the required MAC withdraw messages to reset learning
Single homing onto each of dual parallel VPLS service instances eliminates requirement for VPLS to re-converge for restoration:

- SPBB can use parallel connectivity without looping

Any failure requires only SPBB convergence for complete recovery:

- MACs will simply move from one service to the other
- VPLS does not need to “unlearn” anything

Single points of SPBB attachment need to be dual-homed
We can extend SPBB dual B-VID load balancing over VPLS

• set up as many ELAN services as needed to support the traffic
• each service supports **both** SPBB B-VIDs (blue & red above)
• SPBB SPF computation only installs forwarding state in the SPBB Gateway on an ECMT VID for any **B-MAC DA** reachable through it.
Problem With Broadcast Segments (e.g. VPLS)

Multicast packets from “A” are fine, only one path to “H”

Packet from A will pass RPFC at F if received from broadcast segment.
F is also on shortest path to H
So FDB contains both SA and DA

Without additional filtering multiple copies of any packet from A that is flooded will appear at H

SPF tree from A

SPF tree to H

3 nodes connected to the shared segment are on H’s tree and will accept from “A”
Broadcast Segment Solution

- Slightly enhanced filtering for boxes with a “unary” FDB
- In QBRIDGE-MIB of 802.1ap there is
  `ieee8021QBridgeStaticUnicastReceivePort` entry
  in the `ieee8021QBridgeStaticUnicastTable`,
  - Current “PortNumberorZero” syntax only allows us to
    specify a either promiscuous reception or reception
    restricted to a single port for the DA MAC referred to by
    the table entry…
  - We need the ability to specify some ports, in addition to
    none, one or all….

- A corresponding “PortList” syntax is needed
  - Same as for the objects
    - `ieee8021QBridgeStaticUnicastStaticEgressPorts`
    - `ieee8021QBridgeStaticUnicastForbiddenEgressPorts`