Shortest Path Provider Backbone Bridging Forwarding Solution Options

By Don Fedyk and Ali Sajassi

Introduction

Recently, the Ethernet Shortest Path Bridging [SPB] project has been debating two proposals for source tree identification when Shortest Path bridging is applied in a Provider Backbone Bridge [PBB] context (SPPBB). A number of papers [PLSB] and presentations [SPB-P1], [SPB-P2], [SPB-P3] have been made on the subject and there has been a lot of alignment on many of the technical aspects. These are captured in the Summary section at the end of the document.

However the core technical debate is around the identification of Source tree in the frame (for unicast and mainly for multicast frames). Note that for SPPBB, unicast frame forwarding is much less of an issue than multicast frame forwarding but there are subtle linkages to the choices of mechanism for multicast.

Ethernet primarily forwards packets on a Virtual Local Area Network (VLAN) active topology identified by the VLAN Identifier (VID) and the destination MAC address (DMAC). Shortest path bridging (SPB) requires the source tree (shortest path tree rooted at the source bridge) to be identified in the Ethernet header, further refining the active topology, so frames are forwarded on this shortest path tree in context of the VLAN and the source tree. Besides using source-tree ID in the Ethernet header for forwarding purposes, this ID can also used in performing the proper ingress check to ensure that only the frames received on the right interface, can be forwarded out of the egress interface(s). This paper examines three options for encoding the source-tree ID in the Ethernet header for Provider Backbone Bridges (PBB) and explains the pros/cons of each solution and the trade off among different solutions. One of the main requirements for any of the proposed solutions is to be able to use the existing 802.1ad bridges as intermediate nodes (Backbone Core Bridges) in a SPPBB network – e.g., no hardware modifications shall be required for BCBs. Both solution 1 and solution 2 described in this paper satisfy this requirement. Solution 3 does not meet this requirement; however, it has other advantages which are worth to be discussed and analyzed here. It is assumed that the reader is familiar with the papers cited in the references.

The fundamental reason for providing SPB is to make better use of diverse topology for example mesh networks. However one enabler for SPB is provided by link state protocols. These link state protocols are also capable of improving the dynamic response, stability of the network and topology distribution. This in turn allows larger and more diverse networks to be addressed by SPB.

Note: The context of the paper does not apply to an 802.1Q network where the backbone MAC (B-MAC) addresses are not present. In these networks only solution 1 using shortest path VIDs with learning turned on is viable.

Table of Contents

VLAN Representation

First we should review a few facts about Virtual Local Area Networks (VLANs) and Shortest Path Bridging (SPB). SPB uses the notion of active topology (a loop-free connectivity in the network) just like RSTP and MSTP. A VLAN is a subset of such active topology. In a typical scenario in RSTP and MSTP, a VLAN is represented by a single VLAN-ID (VID). The main deviation for this simple representation of a VLAN to date is a VLAN which corresponds to an E-TREE where two VIDs are used (one in the downstream direction of the tree and the other is the upstream direction of the tree). The baseline SPB expands on such representation of the VLAN where it provides a loop-free connectivity among a set of edge bridges (BEBs – in PBB terminology) represented by a number of Shortest Path VIDs (each SPVID representing a unidirectional tree sourced at a given BEB). This set of VIDs represents the same VLAN in the context of SPB (e.g., different IDs for the same loop-free VLAN connectivity). Furthermore, this VLAN is also represented by a BASE VID for the purpose of identification of the VLAN by the management controls. This BASE VID is used when frames are allocated to the Common Spanning Tree (CST) for connectivity to bridges outside of the SPB region. The CST is a minimum spanning tree that may also be used for connectivity when the SPB is not operating. For example due to miss-configuration of SPVIDs there could be ambiguity preventing shortest path forwarding. The use of BASE VID for CST in connectivity of the bridges inside the SPB region is a topic not relevant to this paper and thus will not be further discussed.

When SPB is used in context of PBB network, as we will see, solution 2 presented in this paper allow for the representation of the SPB VLAN by a single VID (e.g., using only the base VID to represent the VLAN rather than a set of VIDs); whereas, solution 1 uses a set of SPVIDs for such representation.

Regardless of VLAN representation in the SPB (whether a single VID or a set of VIDs are used), multiple VLANs may be used when it is desired to provide multiple shortest paths in networks where a single shortest path may not provide as high redundancy or topology utilization. In this case, the multiple logical VLANs correspond to multiple active topologies just like MSTP.

Shortest path Trees

Before examining the different solution options, it is helpful to explore a few facts about shortest path trees.

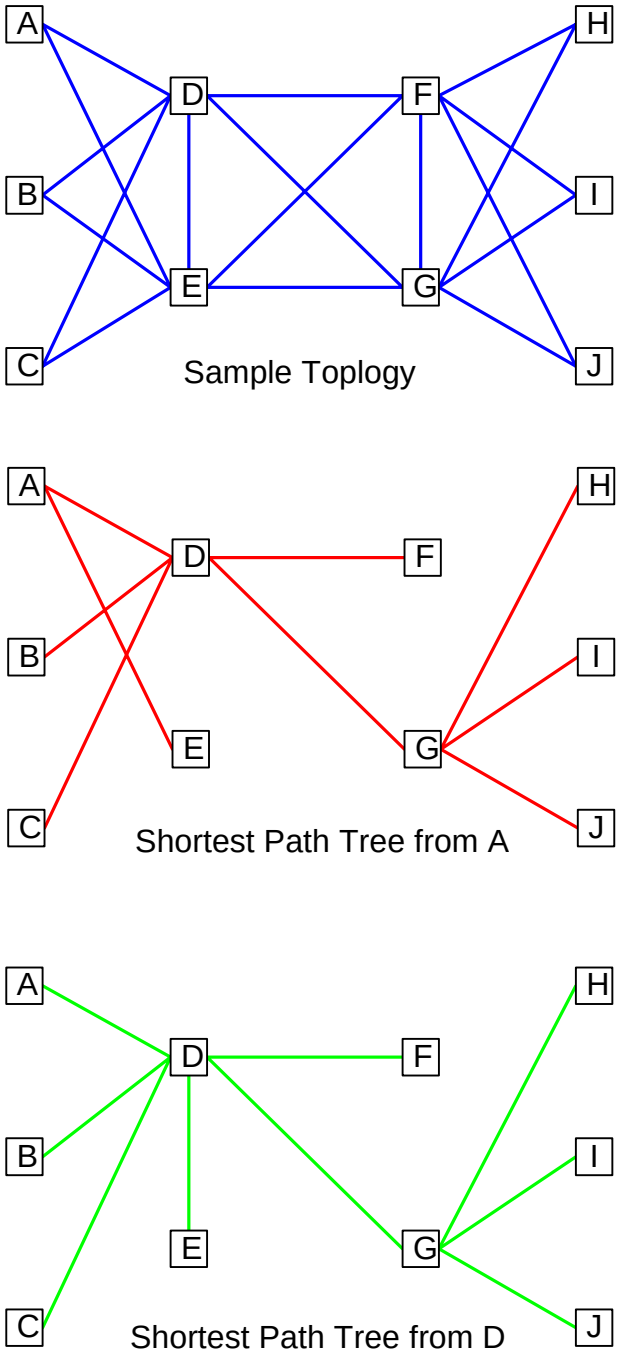


Figure 1 Shortest Path Trees Downstream Congruent

In Figure 1 we have a sample topology where there are two shortest path trees shown rooted at A and D. It is assumed all links have equal weight in this figure.

Congruency Aspects

Congruency requirements have been described in [SPB-2]. Basically, there are two main congruency requirements with respect to the existing bridged network: a) congruency between forward and reverse paths and b) congruency between unicast and multicast data paths in the same direction.

Congruency between forward and reverse paths is required because current bridges perform MAC learning in data-plane and thus in the absence of such congruency, unknown unicast frames can be flooded forever and/or unicast frames may never reach their destination because of the learned path may lead to a dead-end and frames get dropped since the Spanning Tree of the return path may block a given port while the Spanning Tree in the forward path may have the same port in forwarding state. When MAC learning is performed in the control plane, then the congruency requirement between forward and reverse paths may somewhat be relaxed if CFM is adapted; however, as we will see, the main issue in congruency is between unicast and multicast paths in a given direction which once achieved can be extended easily to cover the congruency between the forward and the reverse paths.

Congruency between unicast and multicast paths is required because the following issues may arise in the absence of such congruency:

- Out-of-order delivery of unicast frames
- Lack of unicast path coverage by CFM CC messages
- Black holing of customer data or loop creation in the customer network

For a description of these issues, the reader is referred to [SPB-2]. As it can be seen, the congruency between unicast and multicast paths (in a given direction) is required even if the MAC learning is performed in the control plane (using IGP protocol such as IS-IS).

As it can be seen, these congruency properties are not only essential for learning but are also useful for ingress checking as we will see later. Historically this was more than just learning, the virtual connectivity provided between any two points mimics a physical link where you have symmetrical congruency. Therefore there is minimal impact on Ethernet mechanisms which have been carefully crafted around ensuring no unidirectional failures.

As shown in Figure 1, bridges A and D have multiple permutations of shortest path trees. They must use a deterministic rule to build their trees. The trees that are illustrated have the property that for all shortest path destinations downstream of D, A and D's trees are congruent. We call this attribute downstream congruency. Since there are multiple valid shortest paths, it is possible to have trees rooted on A and D that do not have downstream congruency.

One argument for downstream congruency is that minimum spanning trees created by STP or RSTP are by definition downstream congruent since there is only a single tree.

Given that SPB and SPPBB can honor this requirement we can choose to maintain this property. We will come back to this point when we discuss forwarding options.

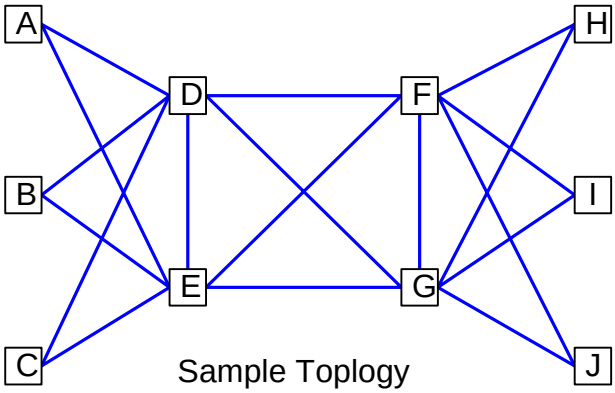
Note that all shortest path trees must also be reverse path congruent. In other words the shortest path from any bridge X to any other bridge Y is simply the reverse path of the path from bridge Y to Bridge X. Again referring to Figure 1 we can see that while there are multiple paths from Bridge A to many other bridges, the reverse path rule will ensure that all trees are reverse path congruent. This is another key factor in the forwarding rules because the source and destination address (SMAC, DMAC) will always be populated in the forwarding nodes. While it is possible to have trees that are not reverse path congruent it is undesirable when maintaining backwards compatibility with Ethernet bridging and functions such as OAM compliance and ingress checks.

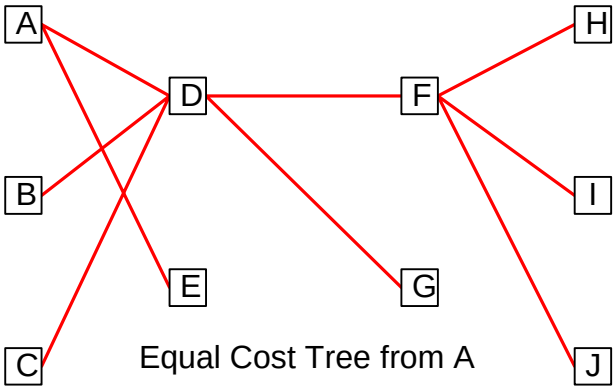
This paper only considers trees that meet both of the above congruency requirements (e.g., unicast & multicast congruency as well as forward and reverse direction). However at this point having trees that are downstream congruent may be desirable in some scenarios but not others as we will see later.

This brings us to an interesting question. Since there are possibly multiple shortest path trees from a source, which one or how many should we choose to build and use for forwarding? It turns out that the rules for generating these shortest path trees have certain properties. One way to choose the shortest path trees is to maximize network coverage. With a single logical VLAN and shortest path bridging it logically follows that you would build trees that are multicast and unicast congruent, reverse path congruent but not downstream congruent in order to maximize the number of links used in the network.

However another way to produces shortest path trees is to use multiple logical VLANs (typically two) and to produce trees that are multicast and unicast congruent, reverse path congruent and downstream congruent. We can term these trees Equal Cost Multiple Tree (ECMT). The downstream congruency is an artifact of the deterministic algorithm within logical VLAN keeping trees from the same VLAN together and it minimizes the number of links used. Naturally the two trees must be driven by different algorithms to minimize the nodes and links in common. In Figure 2 the topology has edge nodes that can have completely disjoint shortest path trees but the core nodes D, E, F, and G only have a single shortest path between them.

In summary, when there are multiple logical VLANs provisioned in the SPB network, then one can either use ECMT or opt not to use it; however, if the SPB network is represented by a single logical VLAN, then ECMT and downstream congruency should be avoided in order to maximize different link utilization in the network.





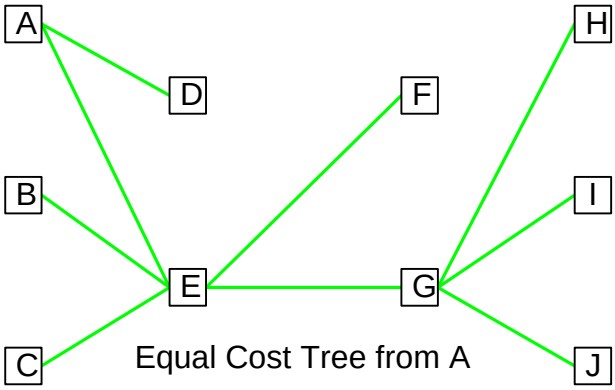


Figure 2 Disjoint Equal Shortest Path Trees

For those familiar with Equal Cost Multiple Path (ECMP) in IP will note there are differences between the ECMT of Ethernet bridging. ECMP selection is performed Hop by Hop. ECMT is performed on a per logical VLAN basis the path is determined by the tree chosen.

Figure 2 illustrates two shortest path trees from Bridge A that are disjoint for every bridge but Bridge E and D (Bridge A's immediate neighbors).

Frame Forwarding

Ethernet Bridging IEEE 802.1 frame forwarding is typically based on forwarding to the destination address and a VLAN based on an active topology. Active topologies in IEEE 802.1 are identified by a Spanning tree where a VLAN can either represent this tree or represent a subset of this tree.

The Spanning tree algorithms [802.1Q] STP, Rapid Spanning tree Protocol (RSTP) and Multiple Spanning Tree Protocol (MSTP), provide a foundation for forwarding by creating a minimum spanning tree for forwarding operations. Minimum spanning trees have the property of having one ingress port and zero or more egress ports.

The forwarding operation is: forward a frame to a destination address (DMAC) in the context of a spanning tree (or subset of) identified by a VID. This operation involves verifying the VID for validity first. The frame may be exclusively forwarded on VID but usually involves VID validation and then forwarding on the DMAC. Exclusively forwarding on VID is for flooding of unknown unicast or broadcast frames. If the DMAC is not known, then Ethernet Bridging will typically invoke the learning process (in data plane). Learning is supported in Shortest Path bridging for conventional Ethernet but when shortest Path bridging is applied to Provider backbone bridging (PBB) the agreed technical decision is to turn off address learning (in data plane) for the Backbone addresses (B-MACs). These B-MAC addresses are all known to the provider backbone bridges and entirely within the Backbone network domain. Learning is not as efficient as distributing the addresses up front using link state. Also by distributing unicast addresses ingress checking based on unicast MAC addresses can be enabled; otherwise, ingress checking must be done based on VLAN ID.

Shortest path bridging is refining the rules for forwarding further by requiring an ingress check be performed on packets as a safeguard for so called "micro loops" or transient loops that may form as well as a safeguard for multicast replication on poorly formed multicast trees. STP and other protocols use port blocking and handshakes between nodes to ensure loop free topology. With link state protocols the usual method of loop mitigation is fast convergence coupled with a packet hop count or time to live (TTL). In Ethernet forwarding there is no TTL field and multicast frame looping cannot be satisfied by TTL so an alternative mechanism the ingress check is proposed [Looping]. The ingress check is an aggressive form of checking that validates the source tree of the frame, and drops it if the frame has arrived on any port other than a port on the source tree. It should be noted that ingress checking is stricter than the hop-count mechanism and it can discard the transient frames during a topology change where they would not be discarded otherwise under the hop-count. Ingress checking minimizes the possibility of frame duplication or misordering. In other words an ingress check will ensure a low probability

of looping frames [Looping]. We will discuss the options for source tree root identification.

Note that the choice of field use for ingress check is dependent on the header encoding. Ingress check may be base on VID, DMAC or SMAC depending on which of these three fields used for encoding of the source tree-id in the frame header.

The PBB Ethernet Header is illustrated in Figure 3. The B-DMAC and B-TAG fields are the fields that Ethernet currently uses to forward frames. As mentioned, SPPBB presupposes turning off learning in PBB networks and using the link state to pre-populate forwarding tables. (Note: this capability is only available for Backbone addresses where the extent of the backbone addresses can be fully controlled and known.)

For review, in normal VLAN forwarding on a minimum spanning tree we need:

- The VLAN of the frame to determine the tree
- The destination of the frame

This forwarding works because in a minimum spanning tree there is only ever one route to the egress point on any VLAN. So the packet may be validated against the VID on ingress and then forwarded based on VID and destination unambiguously. In both unicast and multicast forwarding, the VID identifies the minimum spanning tree (or subset of) and thus the filtering database associated with that tree, and the destination MAC address indicates which branch (or branches in case of multicast) of that tree the frame should get forwarded (or replicated).

In SPPBB, since MAC learning is performed in control plane, the requirement for source-tree identification can somewhat be relaxed for unicast forwarding but not for multicast forwarding. In multicast forwarding, we need to identify (S, G) and the associated filtering database so that the proper frame replication can be done over the egress interfaces associated with the shortest path tree. However, in case of unicast forwarding, the forwarding is done based on destination address (D) and the filtering database for unicast addresses can be shared across many of shortest path trees (no need to have a filtering database per tree for unicast addresses). Furthermore, ingress check can be performed based on either source-tree ID (e.g., soution-1) or based on MAC SA or both. If it is done based on MAC SA, then no source-tree ID is required for unicast forwarding since source-tree ID can be encoded within the MAC SA.

Therefore, one can conclude that the source tree identification in SPPBB can be same as the minimum spanning tree for normal VLAN forwarding if solution 1 is chosen and it can be less restrictive if solution 2 and Solution 3 are chosen.

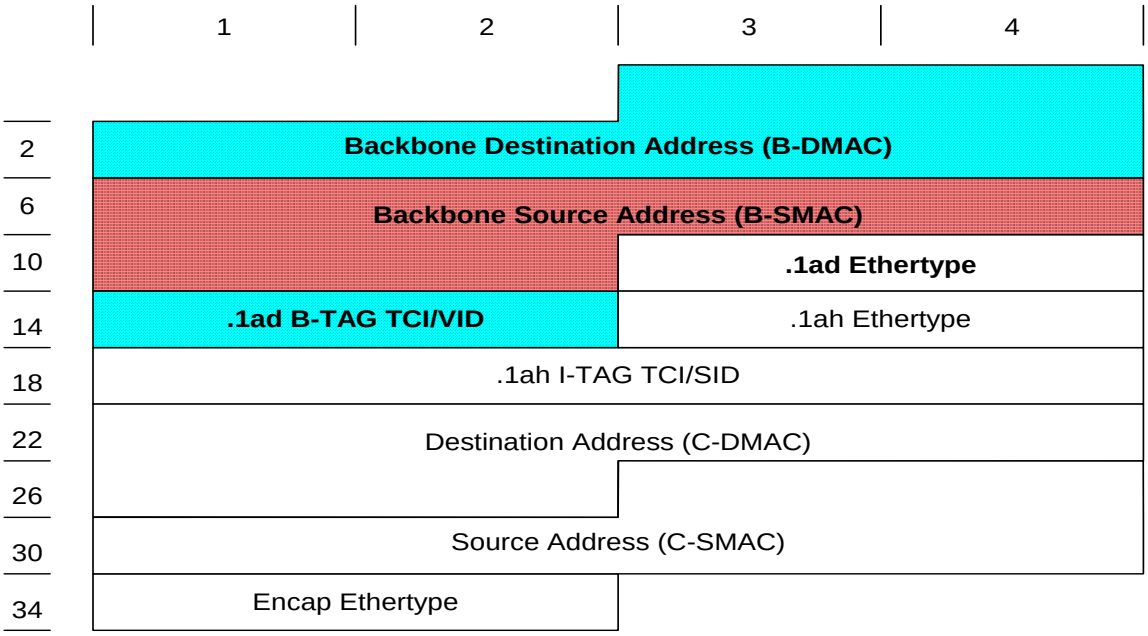


Figure 3 802.1 PBB Header

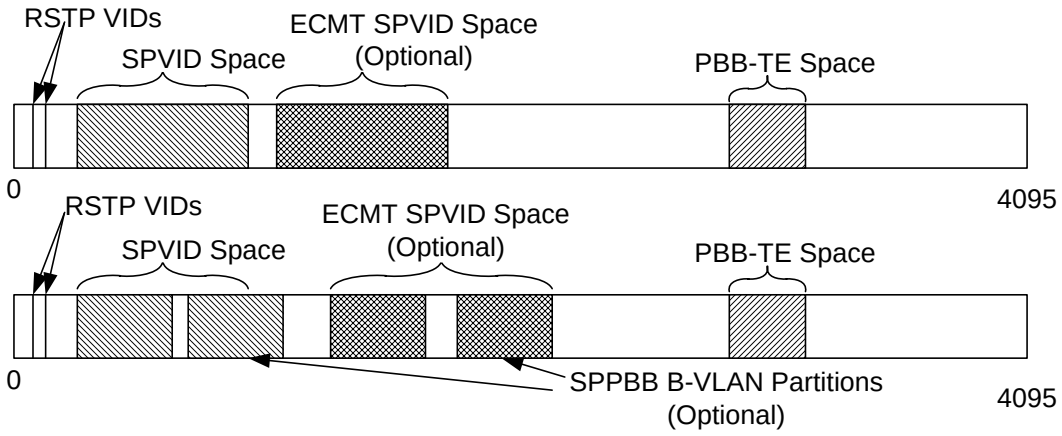
As mentioned earlier, since MAC learning is performed in control plane in shortest path provider backbone bridging (SPPBB), a frame that has no forwarding MAC entry can be dropped since forwarding tables are populated by the link state algorithms prior to forwarding. When a valid forwarding entry is found, the frame can be forwarded along the spanning tree that was determined for that source. In order to determine the source tree root, a mechanism is required to determine the source in the context of a particular VLAN.

VLAN Space Allocation

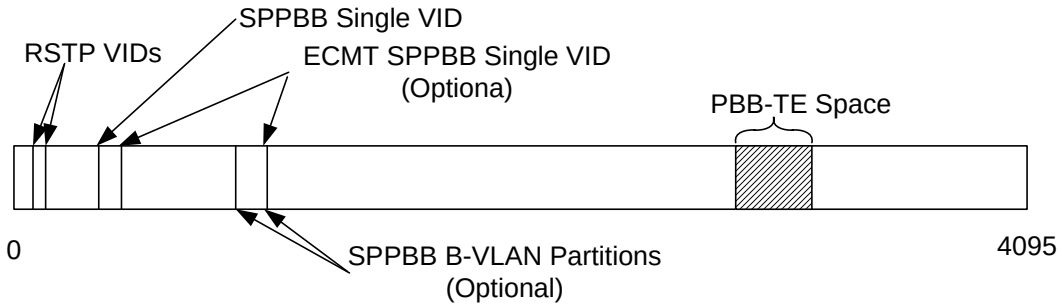
In SPPBB the VLAN may represent a number of functions:

- A minimum spanning tree for the Common Spanning Tree
  One VLAN may be relegated to a CST for some functions.
- A B-VLAN topology
  Network operators can use B-VLAN partitions to control resources of the VLAN to groups of customers. With Services Instances (I-SIDs) this is less of a requirement than in Provider Bridge networks, never the less B-VLANs allow I-SIDs to be mapped to a controlled portion of the network. This does mean allocating VIDs.
- A PBB-TE range for traffic engineered PBB
- An SPVID range or a Single VID as we will see in the following sections.

One of the main drivers of this document is to understand the impact of Source Root Tree identification. Figure 4 illustrates the VID space allocation Tradeoffs.



VID Space Consumption Models for Solution 1



VID Space Consumption for Solution 2 & 3

Figure 4 VID Space Allocation Factors

SPPBB Source Identification Frame Header Encoding Choices

In this paper we have identified the requirements for a source tree identifier in a frame. We now discuss the possible solutions for encoding source tree identification in the Ethernet header. Again it is important to stress the need for source tree identification is primarily for SPPBB multicast forwarding, but SPPBB unicast forwarding should use consistent mechanisms.

Solution 1: SPB VLAN Representation Using Multiple VIDs – Source Tree Encoding using VID

The first solution for source tree identification for Shortest Path Bridging uses a VID to represent simultaneously the VLAN and the Source Bridge or root of the shortest path tree. This solution is discussed in [SPB] and [SPB-P2] and it requires that the source tree to be unidirectional.

The VID field is 12 bits or 4096 VIDs with some VID values being reserved. Shortest path bridging uses a BASE VID and a setoff shortest path VIDs (SPVIDs) to represent the logical VLAN or active topology. The BASE VID is used for the purpose of identification of the VLAN by the management controls. It is also used when frames are allocated to the Common Spanning Tree (CST) for connectivity to bridges outside of the SPB region.

The unicast and multicast forwarding in this solution is very similar to normal bridging where the VID is used to identify the tree and thus the associated filtering database and MAC DA is looked up within this database for forwarding the frames. The VID is also used for performing ingress checking in this solution (e.g., there is no need to perform ingress checking using MAC SA). Since ingress checking is performed using VID, the total number of lookup in this solution is reduced to a single lookup per frame – e.g., a single lookup using VID + MAC DA can perform both ingress checking as well as frame forwarding. In case of multicast forwarding, (S,G) is also identified by the same lookup since VID represents the source tree. Although current 802.1ad bridges used as intermediate bridges (BCBs) can not take advantage of this single-lookup feature, the future bridges can be designed to take advantage of it (and thus reducing the cost or increasing the throughput of the bridge).

Since this solution uses the VID for encoding of the source-tree ID, it can work with normal 802.1ad or 802.1Q bridges as well as low-cost bridges intended for home AVB applications. In other words, this solution does not require that the bridge in the network have PBB capability – it can work with both PBB and non-PBB bridges.

As mentioned earlier, source-tree identification is primarily needed for forwarding of multicast frames where (S,G) for the shortest path tree need to be identified and

subsequently the associated filtering database. For unicast forwarding, there is no need for source-tree identification; however, this info is available with this solution .

The ingress checking for both unicast and multicast frames is performed using source-tree ID (VID) and if a frame arrives at a port which is not on the path toward the root of the tree (e.g., the port is not configured to receive this VID), then that frame gets discarded.

There are two main issues with this solution and they are: a) it limits the total number of bridges in a SPPBB domain to 4K or fewer since each bridge is required to be represented by a VID and b) unicast addresses need to be replicated across multiple filtering databases. The second issue stems from the fact that 802.1ad bridges can operate in either IVL mode or SVL mode with respect to both unicast and multicast forwarding – e.g., it cannot work in IVL mode for multicast forwarding and SVL mode for unicast forwarding. Since multicast forwarding requires IVL mode of operation because each (S,G) needs to have a unique entry in the filtering database, this solution needs to also use IVL mode for unicast forwarding. This means that unicast addresses need to be replicated across many filtering databases and thus wasting TCAM resources.

The typical size of a PBB network is in order of tens or may be hundreds of bridges (but not thousands). If there are thousands of bridges in a single network, then convergence of IS-IS can easily become as issue because contrary to an IP network where a router only needs to calculate a single tree rooted at itself, a SPPBB bridge needs to calculate N number of trees where N is the total number of bridges in a network. Therefore, the first issue may not be the prominent issue here.

The second issue seems to be a bigger of the two since it can consume bridge resources to the point of exhaustion. However, there is a simple remedy for this problem that can be addressed in next hardware revision and that is when a bridge run SPB, then have IVL mode for multicast traffic and SVL mode for unicast traffic. This remedy works for both SPB and SPPBB networks although it is a bit more efficient for SPPBB. With this remedy, known unicast and multicast traffic gets handled the same way for both SPB and SPPBB. However, unknown unicast frames get handled differently. In SPPBB, unknown unicast traffic cannot exist (because of MAC learning is performed in control plane) and thus they get dropped. However, in SPB, unknown unicast traffic is a common fact (because of MAC learning in data plane) and they must be handled accordingly. Since SVL mode is used for unicast traffic, the unknown unicast traffic gets replicated to all the egress interfaces and it then gets filtered via egress filtering for that VID, thus resulting for the replicated traffic only on that source-tree (identified by its VID) to exist the bridge. Therefore, the handling of unknown unicast traffic is less efficient than the handling of multicast traffic in SPB (not in SPPBB); however, the amount of unknown unicast relative to multicast traffic is minuscule and in the noise.

If multiple logical VLANs are required for ECMT, multiple trees may be created each distinguished by a unique VID. Although ECMT is intended to maximize topology coverage in a SPPBB network, its usage is not recommended in this solution because it

results in more VID consumption and thus other means of maximizing topology coverage within a single logical VLAN need to be exercised.

One assignment issue which is common across all the encoding options is how to assign source-tree IDs (or root IDs) uniquely across the SPPBB network among all the bridges. This is basically the same question as – e.g., how to assign unique IDs to source trees in the SPPBB network. Once the tree ID is assigned uniquely, then it can be encoded either into a VID, a DMAC, or a SMAC. Since this issue is in common across all the encoding options, it will not be discussed further since it is of no value in comparison of these encoding options.

Another issue with this encoding mechanism worth mentioning is the handling of some of the CFM procedures such as loopback and linktrace. The current loopback and linktrace procedures expect the same VID to be used in both forward and reverse directions (in both command and reply messages). However, since the VIDs in this approach represent unidirectional trees, different VIDs need to be used in forward and reverse direction. This issue can be addressed easily by modifying CFM messages to incorporate the reverse VID in them. It should be noted that this issue is not just pertinent to this solution but also it is pertinent to E-TREE VLAN in normal 802.1Q or 802.1ad bridges.

Ingress check may use SPVID to identify the source tree root for both multicast and unicast. In SPPBB ingress check could alternatively use SMAC in both unicast and multicast to identify the source since the SMAC and DMAC addresses are pre-populated along the shortest path trees.

Benefits of unidirectional VIDs:

- 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 a several 802.1ah islands operating under the same I-SID space. With this approach, no multicast DB-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

Issues with unidirectional VIDs:

- 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.

- It would require modifications to some of the CFM messages to incorporate the VID in the reverse direction.
- The number of unicast forwarding entries is Order (N^2) each destination needs a representation for each source (SPVID *DMAC).
- SPVID may limit the number of B-VID topologies that could be used for other applications such as PBB-TE.

Solution 2: SPB VLAN Representation Using a Single VID – Source Tree Encoding Using MAC-DA

To keep the concept of a logical VLAN in shortest path provider backbone bridging, another approach is to infer the source tree root of the frame by some other field other than the B-VID. Referring to Figure 3 the B-DMAC and the B-SMAC are the two other fields that can be examined in a valid packet.

This solution was introduced in the Provider Link State Bridging white paper [PLSB] and shows how the shortest path concepts coupled with PBB have additional benefits. In a provider bridge context, the B-MAC for multicast is not necessarily out of the Universal (or Global) address space. This provides an interesting option: by using the locally assigned bit the B-DMAC address can be computed based on other factors. This is even more attractive in link state environment where the computation of addresses can be based on a topology attribute. Figure 5 illustrates a possible encoding of a multicast destination address where a 16 bit Source Nickname field is used to identify sources. The lower 24 bits of the multicast addresses are assigned a Multicast Group Identifier. So in essence the multicast destinations are identified by a source specific destination address. By distributing sources and multicast membership in link state any bridge can create this address when it is on the shortest path tree to a destination.

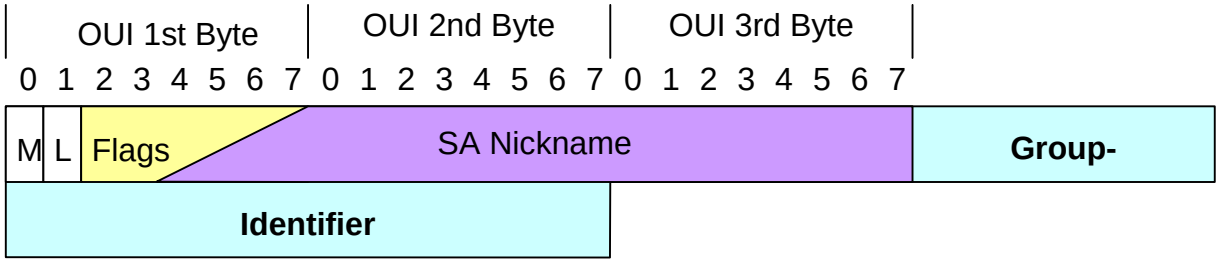


Figure 5: Multicast B-DMAC Encoding

It should be noted that in this solution only DMAC addresses for multicast traffic need to be modified in order to incorporate the source-tree ID. SMAC addresses as well as DMAC addresses for unicast traffic do not require any modifications. This implies that for inter-AS scenarios only multicast DMAC addresses need to be translated which is consistent with PBB operation.

In this encoding option, ingress checking is performed using SMAC for both unicast and multicast traffic. As the forwarding of unicast traffic does not require any source-tree identification, there is no need to incorporate source-tree ID in unicast DMAC filed. However, for multicast traffic, we need to identify a given (S,G) tree and replicate the frames over that tree only. This requires source-tree identification to be encoded as described above in the DMAC. Incorporation of source-tree ID in multicast DMAC ensures consistent operation with respect to existing 802.1ad bridges used as intermediate bridges (BCBs).

Benefits of Source encoded B-DMAC:

- 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.
- 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

Issues of Source encoded B-DMAC:

- This application of the locally assigned address bit must be standardized for multicast addresses. The scope of these addresses is only within the PBB domain.
- It prevents the use of global structured multicast MAC addresses but there are no restrictions on unicast addresses.
- 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.
- Multicast addresses are of the form (S,G) where both S and G are encoded in the DMAC.

Solution 3: SPB VLAN Representation Using a Single VID – Source Tree Encoding Using MAC-SA

This solution was originally included merely for completeness. Some possible advantages of this type of encoding are presented but not quite to the depth of the other solutions. This solution requires changes to Ethernet forwarding that now include a SMAC context for forwarding by looking at the B-SA and therefore are not backwards compatible with Ethernet.

The more obvious choice for source identification is to use the source address itself. Logically this adds a function that has never been used in 802.1 which is the forwarding of a packet with the context of a VID + B-SMAC + B-DMAC for multicast only (unicast forwarding requires ingress checking of SMAC only, as for solution 2).

Source-tree identification using SMAC can be a good solution if the following issues are resolved. First, a single source bridge can be typically represented by many SMACs requiring a hierarchical MAC address representation. Second, the existing intermediate nodes (802.1ad bridges) may not be capable of performing ingress check based on SMAC and thus may require hardware/firmware modifications. The first issue can be addressed by adopting a global frame format as depicted in the following figure:

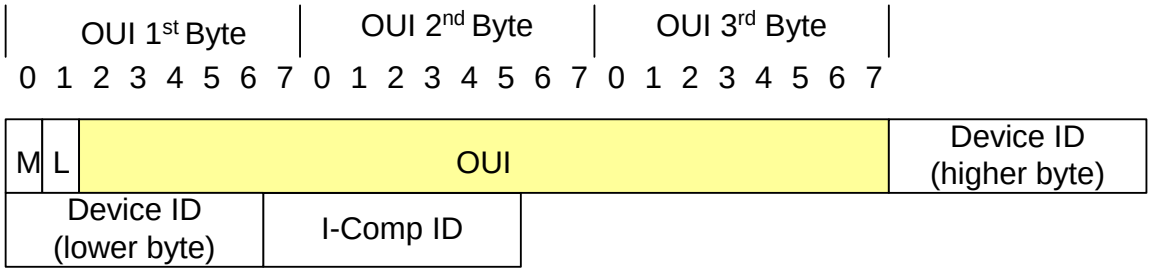


Figure 5 SMAC Encoding

As it can be seen from the figure, the device ID is encoded as a 16-bit unsigned integer after the OUI field. Following the device ID is a 8-bit unsigned integer representing I-Component ID within that device (which can correspond to a device itself or a line card within that device or a port within that device).

The advantage of encoding the source-tree (or device ID) within the MAC SA is that it gives all the advantages of the 2nd approach while enabling the representation of a globally structured MAC address format. It also provides the ability to aggregate CFM messages at the device level since forwarding within a SPPBB network can be performed only based on the 16-bit value device ID.

Benefits of B-SMAC B-DMAC multicast forwarding:

- It uses only a single logical VID for all shortest path trees. Another logical VID may be used for equal cost trees if load spreading is used. By using a single VID for a whole set of shortest path tree, we have preserved the bidirectional nature of VIDs.
- Ingress check is a function of the checking the source address. Since source addresses are populated this operation can be very similar to a source address learning logic by checking of the address is known on the ingress port and discarding the frame if the source address is not populated.
- Unicast forwarding can be shared forwarding based (60 bit lookup).
- Multicast DMAC addresses can be Universal (Global) or locally administered. Multicast entries must be (S,G).

Issues of B-SMAC B-DMAC forwarding:

- Forwarding on SMAC is a new operation.
  The SMAC will take up key space in the Multicast forwarding tables. An effective 108 bit lookup must be performed.
- Multicast addresses are of the form (S,G) where S and G are encoded in the SMAC and DMAC respectively.

Scalability and Performance

When considering scalability several factors come into play to determine overall scalability.

Factors for forwarding entries performance:

- Lookup key size
- Lookup Key Sharing per VLAN
- Number of Source Bridges
- Number of Logical VLANs
- Number of actual VIDs
- Multicast service instances
- Unicast service instances

Figure 6 illustrates the range of lookup parameters and some factors for scalability. As we have seen these factors play in to the scalability. Note that two primary factors for scalability are the number of VIDs and the number of destination addresses.

Unicast addresses are basically tied to the number of external ports in the network. Forwarding options that support a large mesh of unicast addresses will result in source/destination forwarding tables in intermediate nodes. In solutions 2 and 3, this can be reduced by using shared forwarding where the tables are just destination based. In solution 1, shared forwarding can be used but require modifications to 802.1Q bridges to allow SVL mode to be used for unicast frames and IVL mode to be used for multicast frames.

Multicast addresses are related to the number of groups, the size of the groups and the distribution of the groups across the bridges. There is some ability to trade off the granularity of the group with the specifics of the multicast tree by basically doing less specific broadcast with edge based filtering.

Performance factors are related to the number of lookups performed on the header objects and the size of the field. The current Ethernet bridge relay algorithm typically filters on VID and forwards on VID+DMAC a (12+48) bit key. If learning is performed, a common case in conventional bridging, the current Ethernet Bridge must check whether the SMAC is known, and update the forwarding tables if it is not known.

In SPPBB, ingress checks must validate the source tree root of the packet. Depending on the solution chosen the Ingress check is either based on VID validation or VID + SMAC validation. VID based ingress checking has performance equivalent to VID validation and does not require SMAC ingress checking (e.g., does not require the equivalent to the learning operation). The SMAC ingress checking is equivalent to the learning operation but may vary in performance based upon hardware implementation from current learning operations.

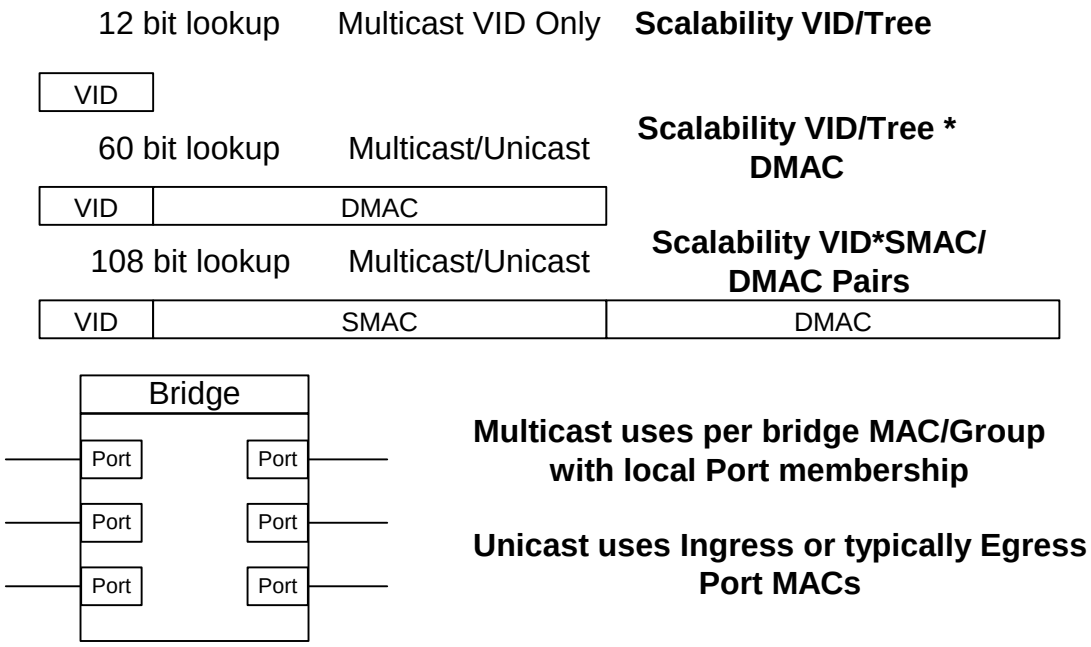


Figure 6 Scalability Factors

Logical Equivalence

Solution 1&2 are functionally equivalent as shown below. The major difference between the two is the choice of putting source information in the VID field or the DMAC. It is

possible that the two solutions could co-exist with very little limitation. Solution 1 is the only option for non PBB environments.

Summary

While the above discussion focused primarily on the encoding of the source tree among the three solutions and the pros/cons of such encoding; with regard to other aspects, there have been many alignments among these three solutions as shown below:

- VLAN Topology — All support shortest path Trees
- VLAN Partitioning — All use a logical B-VLAN
- Link state topology — All use IS-IS
- No Learning — All use IS-IS to populate FIB
- Mesh Networking — All support shortest path trees
- Forwarding: backwards compatibility — All use a VID+DMAC context
- Control plane objects — Similar requirements
- SPT computation — Similar requirements
- Multicast Groups — Support Via IS-IS
- Multicast and Unicast Congruency — Aligned
- Forward & Reverse Path Congruency — Aligned
- Number of Trees for Unicast Forwarding — All use one tree per source BEB
- Number of Trees for Multicast Forwarding — All use one per (S,G)
- Multicast Trees — All use pruning of the broadcast source tree
- Multicast Groups — All can use Groups to represent multiple I-SIDs
- Single path per VID to a destination — Aligned No ECMP
- Ingress Check — All support ingress check
- **Source Tree Root Identification** — **Different - Main Issue**

The following table lists some of the scalability factors.

Source Root Tree Identifier	Solution 1: SPVID	Solution 2: Logical VID with Source encoded DMAC	Solution 3: VID+SMAC +DMAC
VID Usage	(1 per Bridge) x (# of ECMT/BASE VID)	(1 per BASE VID) x (# of ECMT/BASE VID)	(1 per BASE VID) x (# of ECMT/BASE VID)
Unicast Forwarding	VID+DMAC	VID+DMAC	VID+DMAC
Multicast Forwarding	VID+DMAC	VID+DMAC	VID+SMAC+DMAC
Unicast Forwarding Information Base(FIB) Size	1 entry per # of Unicast Destination x BASE VIDs x # of SPVIDs	1 entry per # of Unicast Destinations x # BASE VIDs	1 entry per # of Unicast Destinations x # BASE VIDs
Multicast FIB size	1 entry per Source Tree /Multicast DMAC	1 entry per Source Tree /Multicast DMAC	1 entry per Source Tree / Multicast DMAC
Maximum Flat Network	4000 Bridges/ ((# of ECMT/BASE VID) * (# of BASE VID))	10,000+ Bridges Limited only by FIB entries and Link State	10,000+ Bridges Limited only by FIB entries and Link State
# of Active topologies	Low Each Base VID Group reduces the number of SPVIDs	High Each BASE VID consumes 1 VID Comparable to B-VID	High Each BASE VID consumes 1 VID Comparable to B-VID
Ingress Check	SPVID or SMAC	SMAC	SMAC
# of Lookups (ingress check + forwarding)	1 (0+1)	2 (1+1)	2 (1+1)
Global representation of multicast B-MACs	Yes	No	Yes
CFM Aggregation	No	No	Yes

References:

[802.1Q] IEEE Standard 802.1Q, "Virtual Bridged Local Area Networks", 2005

[PBB] Paul Bottorff, Steve Haddock, editors, “IEEE 802.1ah - Provider Backbone Bridges”, Draft 3.7, August 2007, work in progress.

[SPB] Mick Seaman, editor, “IEEE 802.1aq – Shortest Path Bridging”, Draft 0.3, May 9 2006, work in progress.

[Looping] Mick Seaman, "Exact Hop Count", White paper; December 2006, http://www.ieee802.org/1/files/public/docs2006/aq-seaman-exact-hop-count-1206-01.pdf

[PLSB] Don Fedyk, Paul Bottorff, "Provider link State Bridging", White Paper, January 2007. http://www.ieee802.org/1/files/public/docs2007/aq-fedyk-provider-link-state-bridging-0107-01.pdf

[SPB-P1] Don Fedyk, "802.1aq Shortest path Bridging Design Implications", Presentation to 802.1 Interwork Task Group, July 2007 http://www.ieee802.org/1/files/public/docs2007/aq-fedyk-design-implications-0707-v1.0.pdf

[SPB-P2] Ali Sajassi, “802.1aq: Link State Protocol & Loop Mitigation Options”, Presentation to 802.1 Interwork Task Group, May 2005 http://www.ieee802.org/1/files/public/docs2007/aq-sajassi-link-state-protocol-0705.pdf

[SPB-P3] Ali Sajassi, “802.1aq: Link State Protocol – Part II”, Presentation to 802.1 Interwork Task Group, July 2005 http://www.ieee802.org/1/files/public/docs2007/aq-sajassi-lsp-part-II-0707-v01.pdf

Authors

Don Fedyk, dwfedyk@nortel.com
Ali Sajassi, sajassi@cisco.com