CHAPTER 1

IEEE 802.1aq in a Nutshell: Antecedents and Technology

The Enterprise Local Area Network (LAN) is the traditional Ethernet domain. However, Ethernet has throughout its history widened the range of applications and markets that it could address. Now it is increasingly being equipped to address the provider space, which has significantly different requirements, notably the capability to virtualize large numbers of services to run on common infrastructure. These requirements were the initial motivation for IEEE 802.1aq—Shortest Path Bridging (henceforth SPB).

SPB: ANTECEDENTS AND PRINCIPLES OF NETWORK OPERATION

Summary of Ethernet Connectivity Models

Ethernet was invented to deliver LANs, offering "plug and play" networking, and required no configuration in its original form. Addresses are burned into endpoints at manufacture and are not under the control

802.1aq Shortest Path Bridging Design and Evolution: The Architect's Perspective, First Edition. David Allan and Nigel Bragg.
© 2012 the Institute of Electrical and Electronics Engineers. Published 2012 by John Wiley & Sons, Inc.

of the network. The LAN was a passive medium (coax cable), and a collision detection mechanism was used to arbitrate access to this single shared medium by multiple endpoints. Broadcast was the only native connectivity type, with endpoints being responsible for filtering frames which were not addressed to them. Over time, the requirement emerged to scale Ethernets beyond the 6000-foot limit imposed by the collision detection mechanism. This resulted in the development of bridging, and with it the need to discover the location of endpoints across the bridged network.

This was arguably the only major architectural discontinuity in Ethernet's history, the transition from a LAN segment implemented as a passive shared medium to an actively switched network. This transition was achieved while preserving unaltered the service offered to clients, but it required a completely new network technology, the learning bridge. To allow this perfect emulation of a passive shared medium, the routing system adopted by bridged Ethernet then, and still specified, is flood-and-learn; frames with destination media access control (MAC) addresses unknown to intermediate switches are flooded, and the correct port to use for forwarding subsequent unicast traffic back to the source of the flooded frame is found from the source address by reverse path learning. To permit such broadcast mechanisms without frame looping and network meltdown, the active topology must be highly constrained and offer symmetric connectivity between any two points, the common case being a simple spanning tree.

The moment multiple paths between switching points are installed in this bridged model, whether deliberately for resiliency or acciden-

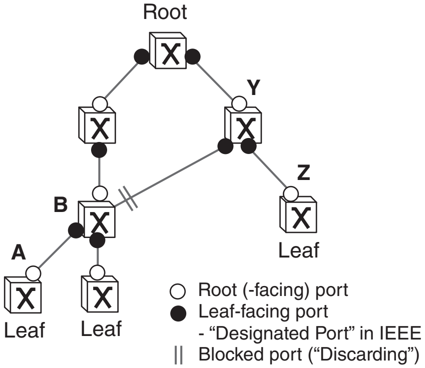

Figure 1.1 A simple spanning tree.

tally by misconnection, a loop is created, which results in network meltdown when broadcast is used. The Spanning Tree Protocol (STP) was developed to disable all redundant paths in a bridged network and create an active topology which is a simple spanning tree (with only one path between any pair of endpoints), and which therefore appears as an exact replica of the coaxial shared medium.

Figure 1.1 illustrates the salient attributes of spanning trees. The tree itself is a directed graph from the root node, which is typically administratively determined, but it is important to realize that the forwarding path thereby established is bidirectional. This means that "go" and "return" paths between any two endpoints must be congruent, which is fundamental to the traditional Ethernet "flood and learn" data path routing process.

The first time a frame, destined for an endpoint attached to Z (say), is sent by a source attached to A, it is flooded by A as "unknown." Copies of the frame traverses the entire tree, and its port of arrival on intermediate bridges allows them to learn how to reach the original source, with no other knowledge of the network at all; on a simply connected tree, a reply can only be delivered by returning it through the port through which the original message arrived. The same mechanism allows the reply from the endpoint attached to Z to teach the intermediate bridges which port to use to reach Z on subsequent occasions.

These mechanisms work functionally and robustly, but have undesirable consequences:

- all redundant links, representing real dollar investment, are turned off; in Figure 1.1, the link between B and Y must be turned off to prevent the formation of the obvious loop.
- as a consequence, traffic routing is often suboptimal, in particular for traffic between leaves having disjoint paths to the root; in Figure 1.1, traffic between A and Z is not able to take the "shortest path."
- the spanning tree offers simply connected connectivity to the set of endpoints served and hence is a single point of failure; the need to guarantee lack of loops at all times requires **all** connectivity on a spanning tree to be disabled after **any** topology change until the new tree has converged. Originally this "shutdown" period typically lasted tens of seconds; this has been improved, but recovery dynamics are still regarded as unacceptable.

SPB introduces link state routing to Ethernet to replace the distance vector algorithm underlying STP, and uses sets of shortest path trees in lieu of a single or small number of spanning trees. This addresses both issues cited above:

- with full topology knowledge, link state allows the control plane to construct loop-free shortest path trees, with no need to disable any data plane connectivity;
- the use of per-source shortest path trees means that connectivity unaffected by a topology change is uninterrupted;
- link state routing inherently has much better convergence properties than distance vector, and SPB has further improved these with speed-up techniques which exploit Ethernet's innate multicast properties.

This replacement of STP by something substantially superior is a general "good" which applies to Ethernet networking in both enterprise and provider space.

Introduction to Virtualization Support in Ethernet

The other key requirement of Ethernet networking, which is increasingly shared by enterprise applications as well as providers, is virtualization, which is the ability to support multiple independent LAN segments on the same physical infrastructure. SPB did not originate the technology to do this, but directly supports earlier IEEE Standards (Provider Bridging and Provider Backbone Bridging) which defined the hierarchical data path constructs to support virtualization.

To provide a summary of virtualization support by Ethernet, Figure 1.2 shows evolution of the increasingly rich header formats which have been defined. In this, a hierarchical layering is implied by prefixes associated with the well-known terms MAC (often used as shorthand for MAC address) and VID (virtual LAN [VLAN] identifier). The prefix "C" refers to customer address information, the prefix "S" refers to provider imposed tags (VIDs) in a Q-in-Q network, and the prefix "B" refers to backbone address information in a MAC-in-MAC network.

A major contribution of SPB is to offer a replacement for spanning tree that is capable of fully utilizing much more richly connected

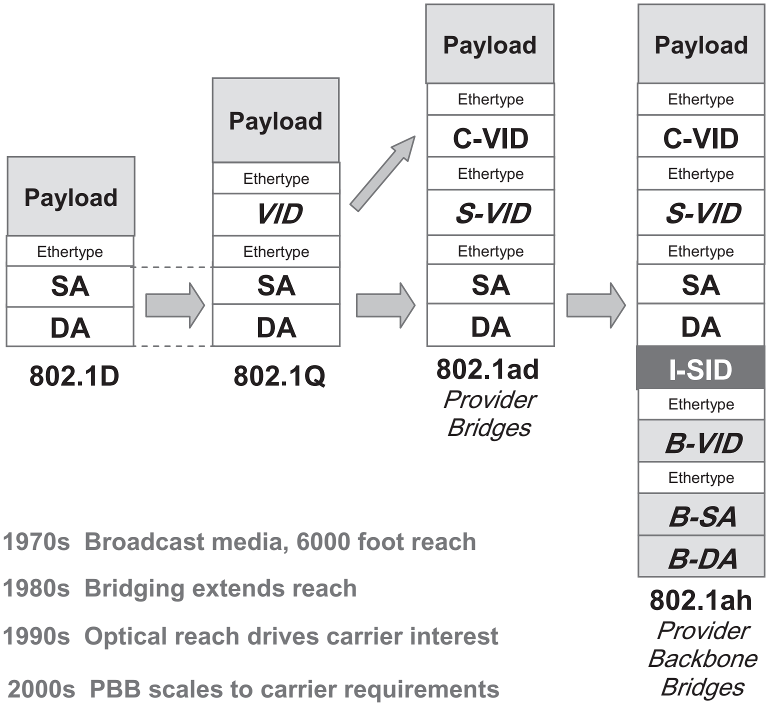

Figure 1.2 The evolution of Ethernet stacking.

topologies. SPB is an umbrella term covering two modes of operation:

- SPBV is VLAN based and builds upon the IEEE 802.1ad (Q-in-Q) tagging structure of Provider Bridges to construct shortest path trees each defined by a different VID,
- SPBM in which the shortest path trees are MAC based (the B-MAC space of Provider Backbone Bridges). VLANs are used to delineate multipath variations.

At a 50,000-foot level, both SPB modes use very similar operations and have very similar overall properties, the primary differences emerging as a consequence of the scaling limitations of the SPBV data plane, which are not shared by SPBM.

The key antecedents for SPBV are:

- shared VLAN learning, whereby MAC addresses learned in one VLAN populate a shared forwarding table for a set of VLANs; this development came with the specification of the ability to support multiple instances of spanning tree in a network;
- the concept of unidirectional VLANs or asymmetric VLANs [SPB].

These two collectively permit a properly constructed mesh of shortest path trees constructed from unidirectional VLANs to employ traditional flooding and learning outside a spanning tree context.

There are two key antecedents of SPBM which foreshadow the techniques it uses:

- Provider Backbone Bridging (IEEE 802.1ah PBB) introduced true hierarchy to Ethernet for the first time, with customer Ethernet traffic using what are referred to as C-MAC addresses being encapsulated in Backbone MAC (B-MAC) addresses across the backbone. This has a number of benefits; the key ones to be aware of now are that the hierarchy directly supports virtualization, and that **all MAC addresses in the backbone are known to and in control of the network operator;** C-MACs are encapsulated and therefore hidden, and B-MACs are all associated with switches in the PBB network itself. This offers a significant degree of summarization of state across the backbone.
- PBB has a comprehensive architectural model, which defines nodal roles in relation to the backbone network (known as a Provider Bridged Backbone Network or PBBN). These are the Backbone Edge Bridge (BEB), which is a node that has both UNIs and NNIs, and the Backbone Core Bridge (BCB) which is purely a transit device at the backbone layer.
- PBB-Traffic Engineering (802.1Qay PBB-TE) exploited this complete knowledge of backbone addressing and topology to permit the disabling of Ethernet's native routing system—flooding and learning. Instead, forwarding tables were **explicitly populated by management or a control plane.**

SPB also explicitly configures forwarding tables, but uses a different control regime, and we now introduce this.

Introduction to Path Computation in SPB

SPB takes a radically different approach to the construction of connectivity compared with spanning tree, but with the result consistent with Ethernet principles. SPBV constructs shortest path forwarding trees between all Provider Bridges in an SPBV domain using shortest path VIDs (SPVIDs) to identify each tree. It is required that the go and return paths between any two bridges, identified by their respective SPVIDs, share a common route in order that source learning works across the core. This permits the "flood and learn" paradigm of bridging to be retained while keeping endpoint state out of the control plane. SPBM constructs shortest path forwarding trees between all BEBs in the network using the combination of B-VIDs and MAC addresses. The MAC learning process in the B-MAC layer is adapted to become a frame-by-frame policing of loop freeness. Symmetrical metrics are used to ensure unicast/multicast congruency and bidirectional fate sharing, both highly desirable properties for Ethernet services. In both cases, the information to derive the forwarding databases is distributed by a link state routing system.

Shortest path forwarding within the Ethernet architecture is achievable because it is possible to fully connect a network with shortest path trees such that there is bidirectional symmetry of the forwarding path between any two points in the network. MAC (SPBM) and VID (SPBV) entries in the forwarding tables are populated by the control plane. This requires placing the responsibility for maintaining a loop-free active topology on handshaking within a link state control plane, and moving away from Ethernet's traditional reliance on a strictly maintained and simply connected spanning tree in the data plane.

Furthermore, it is both possible and practical to condense all SPB control and configuration into a single control protocol: Intermediate System to Intermediate System (IS-IS), which is fundamentally a robust means of synchronizing a common repository of information across multiple platforms. This consolidation is possible because the VID (SPBV), also the Provider B-MAC, B-VID, and Service Identifier information in the form of the I-SID (SPBM) is all global to the network, and so link local forwarding state (e.g., Frame Relay data link connection identifiers, or MPLS labels) is not required for SPB. In other words, the SPB control plane has no need to describe the modification of identifiers within link state control packets crossing the network, because this identifier information is invariant across the network, which is the exact corollary to the fact that in the data plane, Ethernet frames transit the network unmodified; neither requires personalization. Consequently, small extensions to the IS-IS protocol permit control plane flooding of the required VID, B-MAC, and I-SID information within the network.

Connectivity is constructed using a distributed routing system where each node independently computes the local filtering database (FDB), used for the actual forwarding of frames, from the information in the routing system database. The necessary personalization exists only in the form of each node's local view of its position in the network which is extracted from a common information repository during the FDB generation process.

A converged network can have numerous fully connected multipath solutions implemented in the data plane; for SPBV, this requires consuming a VID per node per solution in the network, and for SPBM, one solution can be instantiated per B-VID. So while in the case of SPBV the size of the network directly affects the number of potential multipath solutions that can be deployed, for SPBM it is independent of network size, and the design limit is based on the limitations of the B-VID code space, with in theory support for 4094 multipath solutions before the identifier space is exhausted.

The configuration of a single fully connected solution in a converged network will typically have a multipoint-to-point (mp2p) unicast tree **to** each node in the network (shown as solid black lines to node "A" in Fig. 1.3), and a congruent point-to-multipoint (p2mp) broadcast tree **from** each node to its peers (the pecked lines from node "A" in Fig. 1.3), the latter performing what [Metcalfe] referred to as "reverse path forwarding." These are constructed such that the point-to-point (p2p) path between any two points in the network in a given multipath solution is symmetric and congruent in both directions, and this is true for both unicast and multicast. The tree rooted on node "B" (shown in dotted lines in Fig. 1.3) shows that its shortest path connectivity is different from other trees, but the connectivity between "A" and "B" always uses the reverse of the path from "B" to "A." For SPBM, each p2mp broadcast tree is the prototype for construction of "per I-SID" (per service) multicast trees pruned to connect only BEBs participating in a specific I-SID. The set of multicast trees built to support a specific I-SID offer a perfect virtualized emulation of a traditional Ethernet LAN segment.

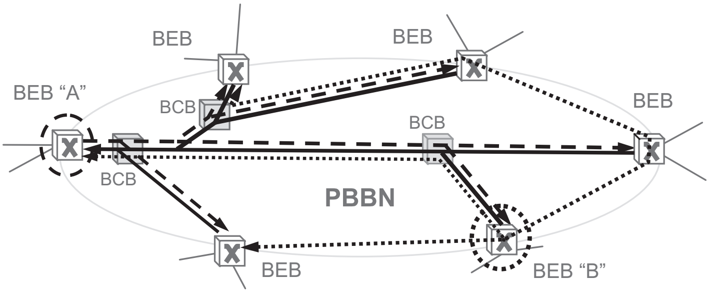

Figure 1.3 IEEE 802.1aq technology—data plane connectivity for BEB "A."

The SPBV Data Plane

Provider Bridging was the first product of the endeavor to adapt Ethernet technology to carrier needs. Provider Bridging specified the imposition of an additional "outer" VLAN tag to the Ethernet frame, known as the S-tag, which permitted a provider to both isolate a customer in a provider network, and also to control subsetting of the network connectivity (the traditional function of a VLAN), in order to properly implement a service instance as a bridged closed user group (see the stack shown in Fig. 1.2).

Embodied in 802.1ad is the ability to translate VLAN tags at provider boundaries to ensure providers can independently administer their own tag spaces, and so avoid a reassignment in one domain propagating into other domains. SPBV also uses this capability. It had never been part of Ethernet before 802.1ad, because a VLAN describes a network-wide topology, and moving between VLANs requires routing, not bridging. It is important to understand this very specific meaning of the VLAN and its tag, which is quite different from the link-local "identifier" used by MPLS, and Asynchronous Transfer Mode before it, where label swapping on every hop is a fundamental part of forwarding. The Ethernet VLAN tag translation function is normally a symmetric function where tag "A" is translated to tag "B" in one direction, and tag "B" is translated to tag "A" in the other.

SPBV implements a VLAN using a set of unidirectional shortest path VIDs (known as SPVIDs), each being used by a different Provider Bridge to mark frames which it transmits into the SPBV network. It is necessary to have an identifier to refer to the complete set of SPVIDs implementing a VLAN, and this is known as the "Base VID" in IEEE documentation. The Base VID is the VLAN that a set of SPVIDs implements.

The tag translation functionality of 802.1ad needs to be modified for SPBV. A VLAN tag received at the ingress to an SPBV network will map to a single SPVID associated with the ingress Provider Bridge, but the reverse function needs to translate the complete set of N SPVIDs associated with the VLAN, one from each Provider Bridge, back to the single S- or C-tag value for egress from the SPBV network. This has necessitated the extension of the symmetrical tag translation concept above for use by SPBV. This defines an N to 1 tag mapping on egress from an SPBV domain, but continues to exclude tag swapping as a switching function (i.e., there is still no 1 to N mapping needed because the mapping on ingress to the domain is 1 to 1).

An SPVID is unidirectional, and the set of SPVIDs that implements a VLAN operates in shared VLAN learning (SVL) mode. Within a bridge, a VLAN, and its tag, are associated with the use of a single FDB to control frame forwarding on that VLAN. With shared VLAN learning, multiple VLANs are assigned to the same FDB. As a consequence, a MAC address learned when received as a source address on one VLAN is used when received as a destination address on any VLAN to determine the forwarding action.

In the specific case of SPBV, a source MAC in a given frame, tagged with the SPVID of the ingress SPBV provider bridge, is learned by all SPBV bridges transited by the frame as applicable to the complete set of SPVIDs associated with the VLAN the MAC arrived on, so that a frame destined for that original source may be forwarded irrespective of the bridge (hence SPVID) which sent it.

Existing MAC registration protocols for multicast groups may interoperate with an SPBV environment, and registrations received at the edge of an SPBV region are advertised throughout the region using IS-IS.

The SPBM Data Plane

Provider Backbone Bridging (IEEE 802.1ah PBB) was the culmination of the evolution of the Ethernet forwarding path, allowing for a full encapsulation of the customer functions of topology and service identifying frames. SPBM inherits this forwarding path unaltered. Both PBB and SPBM use an 802.1Q standard header and an S-VLAN

Ethertype, but unlike Provider Bridging, separate the service identifier from the backbone VLAN (B-VLAN) and instantiate it completely independently as the I-SID (see Fig. 1.2). This is important since the number of VLAN topologies is typically a scaling constraint for Ethernet (only 4094 VIDs are available), and so when the VID is overloaded and used as a service identifier as well, this severely impacts the number of services a Provider Bridged Network can support.

The separation of VID and service ID permits the services to scale independently of topology; the B-VID is then delegated exclusively to the role of engineering the network. SPBM also uses the term Base VID (as above) to refer to the VID identifying the VLAN, but unlike the case of SPBV where an SPVID must be used to identify the source bridge, SPBM can use the edge bridge B-MAC address for this purpose. This is because the domain of the IS-IS control plane is fully congruent with the set of endpoints in the backbone. Consequently, SPBM can fully mesh the network with a single VID, and so there is a 1:1 correspondence between the Base VID and the SPBM B-VID.

The I-SID is a service identifier which is unique and consistent within a provider network. The binding of a particular I-SID to a set of BEB customer network ports uniquely identifies a community of interest, which is implemented as a virtual switched broadcast domain between those ports, over which customer transparent bridging operates. I-SIDs are normally associated with a single B-VID.

Customer Ethernet traffic is adapted onto an SPBM network in the same manner as used in 802.1ah PBB. A customer's Ethernet frame arrives at a BEB at the edge of the SPBM network, and is mapped to the customer I-component and I-SID associated with the customer tag or port. Associated with the I-component is a table, exactly analogous to the FDB of a physical bridge, which records the set customer MAC addresses received together with the B-MAC address of the remote BEB which encapsulated and sent them. This is exactly the normal reverse path learning mechanism associated with bridging, except that C-MAC addresses are here associated with the B-MAC address of the BEB via which the C-MAC can be reached, rather than a physical port. Thus, the backbone MAC simply becomes a named interface in a large distributed bridge.

When the customer frame's destination C-MAC cannot be resolved to a B-MAC, or it is a broadcast or multicast frame, then the I-component will resolve the address to a Group (multicast) B-MAC associated with the local I-component, which identifies the specific shortest path multicast tree over the backbone for the combination of that source and I-SID.[1] Forwarding of the frame addressed in this way floods the frame to the other BEBs that have registered interest in receiving that I-SID. The peer BEBs learn the customer source C-MAC to ingress BEB B-MAC binding, analogous to MAC learning today but using B-MAC named "ports" rather than physical ones. When a response is elicited from the customer destination, the initial ingress BEB learns the binding of C-MAC to far end BEB B-MAC from this response and populates the I-component table accordingly, whereupon subsequent traffic between that C-MAC pair uses unicast communication over the backbone.

The complete encapsulation provides for a comprehensive customer–provider demarcation point. The service provider network only transports frames in a provider frame format containing provider administered identifiers. This allows the service provider to separate the topologies used by different customers, or aggregations of customers, by controlling the mapping of I-SIDs to different B-VLANs. Many customers can be supported on a single B-VLAN. It also isolates the behavior of incompetent or malicious customers from the core of the network.

This service identifier thus allows for a greater degree of flexibility in managing services than hitherto, by allowing their complete independence from the topology.

The other advantage of encapsulation is that customer addresses and customer MAC learning are isolated to the provider edge, with the adaptation function providing the mapping between the customer MAC space and the provider MAC space. As the number of BEBs is orders of magnitude lower than the number of customer MAC endpoints supported by the PBBN, the overall scalability of bridging increases by a corresponding amount. Scalability can now be global as interconnected sets of C-MAC addresses are held only at the edge of the network, and moreover, only at those BEBs which have registered an interest in the specific service.

[1] This is where SPB deviates slightly from PBB. Because PBB is based on spanning tree, the forwarding tree is common to all BEBs, and PBB can use a common multicast address for an I-SID that is used by every source hosting an instance of that I-SID. SPBM is required to use a unique multicast address per source per I-SID since a unique MAC-based tree per source is needed as a consequence of shortest path forwarding.

Furthermore, this encapsulation has the merit that operations, administration, and maintenance (OAM) procedures are significantly simplified, as the provider edge can now be instrumented independent of the customer addresses. Finally, this separation allows the control plane functions of the carrier to be completely independent of the customer, and vice versa. In particular, there is no need for the carrier to peer with the control plane of all of his customers; the carrier is just providing a completely isolated multipoint-to-multipoint (mp2mp) LAN segment to the customer, and the customer may run over that what he chooses.

SPBM treats unicast B-MAC addresses as falling into two classes as far as backbone control plane operation is concerned. These are

- Nodal MACs,
- Port MACs.

Nodal B-MACs are the SPBM equivalent of the “loopback address” of an Internet Protocol (IP) router, and are the way in which the IS-IS instance on an SPBM node address their peers on other nodes. They may also be used for addressing user data frames if the receiving node can determine how to process the frame from the I-SID only.

However, the PBB forwarding model which SPBM inherited allows multiple granularities of B-MAC addressing in the forwarding path:

- at the nodal level, as above,
- also at the card level, the processing subsystem level, and the customer backbone port (CBP) level.

The reason for this is to allow implementations to be optimized:

- nodal level addressing allows the greatest scalability, but does require per I-SID virtual switches to be implemented on the node NNIs,
- the more granular addressing options allow the NNIs of a node to identify the target virtual switch on the basis of B-MAC alone.

There are two important points to make about these two address classes:

1. Port B-MACs play no part in topology determination or path calculation, which use only the nodal addresses. Port B-MACs are associated with their nodal B-MAC only at the time at which forwarding tables are determined.
2. All SPBM nodes automatically install nodal B-MACs for all other nodes on all B-VID planes. This provides fully connected internode connectivity at all times by default, for example, for use by OAM. To enhance scalability, port B-MACs are installed using the same criteria as per service multicast forwarding; in other words, a node only installs the port B-MACs associated with a service if it lies on the shortest path between two nodes which host endpoints of that service.

All unicast B-MAC addresses and I-SIDs are known to and distributed by IS-IS. Instead of distributing the multicast addresses in SPBM, they are constructed locally by creating a unique source-specific multicast address according to a "well-known" algorithm. A 20-bit identifier called the shortest path source ID (SPSourceID) identifies the source bridge for multicast forwarding. A Group MAC address is formed by concatenating the SPSourceID with the 24-bit I-SID value. The SPSourceID is unique in the network and therefore confers uniqueness on the algorithmically constructed multicast address.

Ethernet's support of up to 4094 VLANs permits multiple sets of equal cost trees (ECTs) to be implemented for both SPBV and SPBM in order to support multipath forwarding over multiple fully connected planes. An "SPT set" corresponds to an individual instance of a single plane fully connecting the network. For SPBV, multiple SPVIDs are used in the construction of each SPT set. For SPBM, an SPT set is delineated by a B-VID. Multipath is only really useful if there is some degree of path diversity between SPT sets, hence an SPT set is typically associated with a particular ECT algorithm for path generation.

The final aspect of both the SPBV and SPBM data planes is the data plane OAM. Ethernet OAM (the IEEE 802.1ag and Y.1731 tool suites) all operate entirely in the data plane, because in bridged Ethernet the routing system is "flood and learn," and there is no control plane. Since SPB makes no changes to the Ethernet data path or the semantics of the data plane identifiers, the entire OAM tool suite can be inherited unaltered.

SIDEBAR: *The Metro Ethernet Forum (MEF) Service Models and Interfaces*

The basic MEF service set describes three connectivity models in the specification MEF 6.1. These are E-LINE, E-LAN, and E-TREE as well as their virtualized (or tagged) equivalents:

- E-LINE corresponds to a p2p Ethernet tunnel,
- E-LAN is an mp2mp LAN segment.
- E-TREE is a split horizon client–server variation on the LAN segment model, which is particularly useful for backhaul and content distribution applications. In an E-TREE, leaves can only communicate with roots, while roots can communicate with both leaves and other roots.

The MEF definitions go significantly further than simply discussing the connectivity primitives, casting these definitions specifically in terms IEEE 802.1ad, and then continuing further to define an entire service architecture including interfaces and instrumentation around these definitions. We have found it useful to consider the 802.1ah and 802.1aq I-SID instantiations of these services to be exact matches of their 802.1ad equivalents, and we can extend the umbrella of these terms directly into the SPB space. The MEF, however, has not defined interfaces in terms other than that of 802.1ad.

The IEEE has been studious in ensuring backward compatibility during every step of Ethernet's journey. The PBB (and therefore SPBM) network model maps the IEEE 802.1ad S-tagged service to the I-SID while preserving all of the attributes of connectivity.

SPB TECHNOLOGY: THE CONTROL PLANE

The IS-IS Routing System Requires Modest Enhancements

The IS-IS link state routing system is put to work to control SPB configuration. IS-IS is uniquely suited to this task due to its robust, standards compliant implementation and many years of live deployment.

IS-IS, the base protocol used by SPB, is commonly associated with IP, but is in fact not dependent on IP at all. IS-IS is a pure Layer 2 protocol, and is capable of discovering a network Layer 2 topology through the use of both a Hello protocol to its immediate neighbors, and by using a flooding protocol (link state updates) to all other nodes in the network. The Hello protocol is used to learn the identifiers (MAC addresses) of the nodes immediately adjacent to a node. The flooding protocol is used to advertise information throughout the IS-IS domain, about a node's immediate neighbors and, in the case of SPBM, about attached service endpoints (I-components corresponding to E-LINE, E-LAN, and E-TREE service instances).

IS-IS in effect provides the distributed database upon which each SPB node executes computations. SPB can therefore be thought of as a sophisticated computation that takes network topology and service information endpoint provided by IS-IS as input, and produces FDBs as an output.

The key factor that allows the collapse of all requisite functionality into a single control protocol is that the Ethernet data plane is fully self-describing, and Ethernet frames transit the network unmodified. The importance of this cannot be overemphasized. The addressing and service identifiers are globally unique network-wide. This property eliminates the need for signaling or any form of per node personalization of the data as an additional convergence step, which is a major advance. Signaling only becomes necessary when forwarding state is locally unique, since local-to-local relationships (such as label switching) must be signaled along every path. By contrast, with SPBM's globally unique MAC/VID addresses, any topology change flooded as a single database update provides all nodes in the network with sufficient information to compute the new network configuration.

The elimination of signaling and the integration of service knowledge into a single control plane radically simplifies the control structure, collapses the number of steps to network convergence, and eliminates race conditions between control protocols.

The next sections introduce the information model used by SPB and summarize the extensions to IS-IS required.

Visual Model of Control Plane Information

These next sections introduce the new information items needed for IS-IS for SPB.

The new items associated with a node are modest in number. Referring to the figure above, the nodal nickname, known formally as

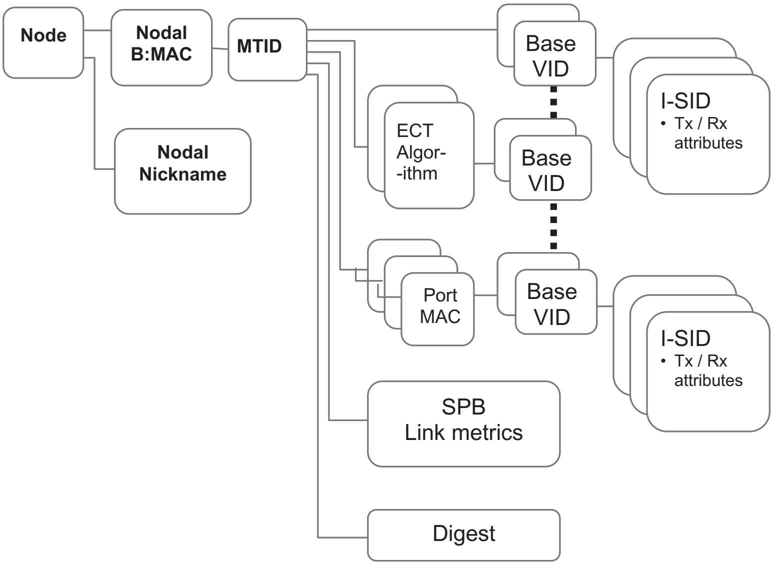

Figure 1.4 New information items in IS-IS for SPBM.

the SPSourceID, is the 20-bit value used to construct the service-specific (S,G) multicast addresses. SPB has its "own" link metric field, to avoid any interaction with other IS-IS applications. The digest is a compact topology summary, used to determine whether or not nodes share an identical topology view, which is a key part of the synchronization process used to guarantee loop-free forwarding at all times.

The remaining items are the nodal unicast MAC addresses, B-VIDs, and the services associated with each. The nodal B-MAC is the SPBM equivalent of the loopback address, and need be the only externally visible address in an SPBM domain. However, PBB permitted different granularities of B-MACs, to allow implementation trade-offs to be made. Multiple B-VIDs for load spreading and traffic engineering may be associated with SPBM operation, and hence the services (I-SIDs) and port B-MACs may only be associated with a single B-VID.

The information model for SPBV is a significantly simpler subset of the one for SPBM. It is presented in Chapter 3, in the more detailed treatment of the control plane ("Visual Model of Control Plane Information," p. 89).

Link State Packet (LSP) Extensions for Link State Bridging

Link state bridging introduces no new protocol data units (PDUs) to IS-IS and adds only new type-length value (TLV) fields and sub-TLVs to the existing IS-IS PDUs. These are briefly and informally enumerated now, with fuller details of each type and its parameters deferred until later ("New IS-IS TLVs for Link State Bridging," p. 90):

(a) ***The Multitopology Aware Port Capability (MT-PORT-CAP) TLV.*** It differentiates topology instances in Hello PDUs. Each IS-IS topology allows only one metric per link; multitopology (MT) allows the use of different IS-IS metric sets running on the same topology if this is desired for manipulation of preferred traffic paths.

- this carries an MT identifier (for possible use in future), and
- an overload bit specifically for use by link state bridging, used to indicate whether the bridge can be used for transit, following the analogy of the generic IS-IS overload bit.

(b) ***SPB MCID Sub-TLV.*** This sub-TLV is added to an IS-IS Hello (IIH) PDU to communicate the multiple spanning tree configuration identifier (MCID) for a bridge. This digest is used to determine when adjacent bridge configurations are synchronized. The data used to generate the MCID is the allocation of VIDs to the various protocols used by the bridge, which is populated by configuration, and the digest is based on a cryptographic hash of these allocations. Adjacent SPB bridges may only use the link between them for SPB traffic if their digests are identical. Two MCIDs are carried to allow transitions between different but nonconflicting configurations.

The important information elements are:

- The MCID and the auxiliary MCID. The complete MCID identifies an SPT region, and its computation is defined in [SPB].

(c) ***SPB Digest Sub-TLV.*** This TLV is added to an IIH PDU to indicate the current topology digest value. Matching digests indicate (with extremely high probability) that the topology view between two bridges is synchronized, and this is used to control the updating of multicast forwarding information. Digest construction is considered later in the control plane description, under “Agreement Digest Construction details” (p. 115).

During the propagation of LSPs the Agreement Digest may vary between neighbors until the key topology information in the LSPs is synchronized. The digest is therefore a summarized means of determining agreement on database consistency between nodes, and may hence be used to infer that the nodes agree on the distance to all multicast roots. The SPB Digest sub-TLV contains the following key information:

- A (2 bits) The *Agreement Number* 0–3 (a rolling count sequence number), which aligns with the *Agreement Number* concept fully described in [SPB], used to guard against control packet loss.
- D (2 bits) The *Discarded Agreement Number* 0–3 which aligns with the *Agreement Number* concept of [SPB].
- Agreement Digest. This digest is used to determine when IS-IS is synchronized between neighbors, and comprises a hash computed over the set of all SPB adjacencies (all edges) in all SPB MT instances. This reflects the fact that all SPB nodes in a region must have identical VID allocations, and so all SPB MT instances will contain the same set of nodes.

(d) ***The Multitopology Aware Capability TLV.*** It differentiates topology instances for other SPB TLVs.

(e) ***SPB Base VLAN-Identifiers Sub-TLV.*** This sub-TLV is added to an IIH PDU to indicate the mappings between ECT algorithms and Base VIDs. This information should be the same on all bridges. Discrepancies between neighbors with respect to this sub-TLV are temporarily allowed during upgrades (e.g., during the assignment of new ECT algorithms to Base VIDs), but all active Base VIDs, as declared by the state of the Use-flag below, must agree and use the same ECT-ALGORITHM.

The key information element is a list of ECT-VID tuples, each comprising

- The ECT-ALGORITHM (4 bytes), which declares that the advertised algorithm is being used on the associated Base VID.

- The Base VID that is associated with the SPT Set defined by the ECT-ALGORITHM which supports a single VLAN over the SPT region.
- A Use-flag, which is set if this bridge, or any bridge in the SPB region, is currently using this ECT-ALGORITHM and Base VID. This is formed from the logical OR of the U-bits (found in the *SPB Instance Sub-TLV* below), and is used to ensure orderly upgrade when new Base VIDs are introduced.

(f) ***SPB Instance Sub-TLV.*** The SPB Instance sub-TLV announces the SPSourceID for this node/topology instance. This is the 20-bit value used for formation of multicast DA addresses for frames originating from this node/instance. The SPSourceID occupies the upper 20 bits of the multicast DA together with 4 other bits (see the SPBM multicast DA address format). This sub-TLV is carried within the MT-Capability TLV in the fragment ZERO LSP.

This TLV carries several information elements used for compatibility with bridges running STP, and these are not enumerated here. The important information elements from the point of view of SPB comprise the following:

- Bridge Priority is a 16-bit value which together with the low 6 bytes of the System ID form the spanning tree compatible Bridge Identifier. This Bridge Identifier is a unique value which is used in SPB by the ECT tie-breaking algorithms.
- The SPSourceID is a 20-bit value used to construct multicast DAs for SPBM multicast frames originating from the node which originated the LSP containing this TLV.
- A list of ECT-VID tuples. Each ECT-VID tuple defines one VLAN, and comprises the ECT-ALGORITHM and Base VID information given earlier, under the SPB Base VLAN-Identifiers sub-TLV, and adds a declaration of whether any I-SIDs are assigned to this Base VID at this node (the U-bit). Each ECT-VID tuple also declares the SPVID used by this bridge to identify it as the root of a shortest path tree when operating in SPBV mode.

(g) ***SPB Instance Opaque ECT-ALGORITHM Sub-TLV.***

(h) ***SPB Adjacency Opaque ECT-ALGORITHM Sub-TLV.*** There are multiple ECT algorithms already defined for SPB; however, additional algorithms may be defined in the future. These algorithms will use this optional TLV to define new algorithm tie-breaking data. There are two broad classes of algorithm, one which uses nodal data to break ties and one which uses link data to break ties, and so as a result two identically formatted TLVs are defined to associate opaque data with either a node or an adjacency.

(i) ***SPB Link Metric Sub-TLV.*** The SPB Link Metric sub-TLV occurs within the Extended Reachability TLV or the MT Intermediate System TLV.

The important information elements are

- SPB-LINK-METRIC indicates the administrative cost or weight of using this link as a 24-bit unsigned number. Smaller numbers indicate lower weights and are more likely to carry traffic. Only one metric is allowed per topology instance per link.
- Sub-TLVs can include an opaque ECT Data sub-TLV, whose first 32 bits are the ECT-ALGORITHM to which this data applies. This sub-TLV carries opaque data for the purpose of extending ECT behavior in the future.

(j) ***SPBM Service Identifier and Unicast Address Sub-TLV.*** The SPBM service identifier and unicast address sub-TLV is used to declare service group membership on the originating node and/or to advertise an additional (port) B-MAC address by which the I-components supporting the declared service instances may be reached. The SPBM service identifier sub-TLV is carried within the MT capability TLV.

The information elements are

- A single B-MAC address, which is a unicast address of this node. It may be either the nodal address, or it may address a port or any other level of granularity relative to the node.
- The Base VID (and hence the ECT-ALGORITHM) to which the following list of service identifiers are assigned.
- A list of service identifiers: ISID #1 . . . #N are 24-bit service group membership identifiers. Each I-SID has a transmit (T)

and receive (R) bit which indicates if the membership is as a transmitter/receiver or both (with both bits set). In the case where the transmit (T) and receive (R) bits are both zero, the I-SID is ignored for the purposes of multicast computation, but the unicast B-MAC address must be processed. In this scenario, edge based replication of broadcast, multicast, and unknown frames replaces the use of an (S,G) multicast distribution tree.

The SPBM service identifier sub-TLV is carried within the MT capability TLV and can occur multiple times in any LSP fragment.

(k) ***SPBV MAC Address Sub-TLV.*** The SPBV MAC address (SPBV-MAC-ADDR) sub-TLV is not used by SPBM, only by SPBV. It contains the following information elements:

- SR bits (2 bits), which derive from Multiple MAC Registration Protocol [MMRP], specifying (typically at port level), the service requirements external to the SPBV region applicable to the following set of group addresses.
- SPVID (12 bits); the SPVID and, by association, the Base VID and the ECT-ALGORITHM and SPT set that the MAC addresses defined below will use.
- A list of Group MAC addresses which declare this bridge as part of the multicast interest for these addresses, with bits to indicate if membership is as a transmitter, a receiver, or both. Pruned multicast trees can be constructed by populating FDB entries for the subset of the shortest path tree(s) that connect the bridges supporting that MAC address as a receiver. This replaces the function of the 802.1ak [MMRP] for SPTs within an SPBV network, and allows the semantics of MMRP messages received at the edge of an SPBV region to be flooded across it.

SPB TECHNOLOGY: PATH COMPUTATION

To this point, the SPB data path and the extensions to IS-IS needed to support them have been the main focus. However, several other issues still needed to be resolved, largely related to the challenge of how to provide a node with the data it needs to compute its forwarding state. The following section also highlights how SPB computes forwarding state for only a subset of all destinations so that the familiar E-LINE, E-LAN, and E-TREE services may be rapidly created in very large quantities.

Computing Forwarding State

SPB nodes learn the topology of the network in a standard IS-IS link state fashion, and once each node has learned the topology then the shortest paths for unicast and multicast traffic are determined by simple shortest path (Dijkstra) computations against the data distributed by IS-IS.

SPBV scopes multicast (for an entire VLAN) using SPVIDs, while SPBM scopes each multicast receiver set (per source per I-SID) using service-specific multicast addresses, each within a single B-VID if multiple paths are being used. The decision process for computing multicast SPB forwarding state for both modes of operation is fairly straightforwardly described. Every node asks itself a simple question: "Am I on the unique shortest path between a given pair of nodes and do those nodes participate in a common service?" This requires that some variation of the "all-pairs shortest path" algorithm is run against the link state database. When an SPBV node finds itself on the shortest path between two bridges for a given VLAN it installs those bridges' SPVIDs associated with that VLAN on the appropriate ports in its FDB.

When an SPBM node finds itself on the shortest path between any two BEBs for a given SPT set/B-VID, it checks the transmit/receive attributes of I-SIDs assigned to that B-VID on those BEBs, and if it is on the shortest path between a transmitter and a receiver on an I-SID, the node installs any associated unicast port MACs and locally constructed multicast MAC addresses in the local FDB. In this manner, only bridges involved in the forwarding of traffic for a service will ever install forwarding state for that service. When all nodes in a given path have completed the computation and installed forwarding state, a given path will be complete end-to-end.

The routing system is "single touch" for service addition and removal; only the node that is joining or leaving needs to be configured with the service change. All other nodes will be informed by flooding in IS-IS, and the multicast trees and unicast forwarding paths will be adjusted accordingly to keep the routing optimal. It is an important property of shortest path trees that neither addition nor removal of a node alters the tree to other nodes, so changes to service membership do not disrupt unaffected nodes.

Per-Service-Instance Routing and Forwarding

Computation of IP forwarding tables traditionally requires only a single shortest path calculation, with the computing node placed at the root, to determine the next hop to the set of destinations. SPB requires a node to compute whether it has a transit role for traffic between all possible pairs of nodes in the network, and therefore requires the computation of "all-pairs shortest paths." Although this is computationally intensive, some two orders of magnitude more performance is now available in embedded processors compared to when shortest path calculation was first deployed. Furthermore, the modern trend in processor architecture is to move to multiple cores, and the N × Dijkstra calculations to perform "all-pairs shortest paths" for an N node network naturally partition into coarsely parallel threads on as many cores as are available.

With its use of the SPVID as a network-wide source node identifier in the data plane, SPBV builds a complete mesh of broadcast trees, one per node per SPT set.

SPBV has the same limitations as QinQ in that the SPVID set is overloaded to be both a topology and a service instance. An immediate consequence is that the scalability of services is drastically constrained due to the VID consumption needed to construct basic connectivity. SPVIDs are unidirectional and this does permit the construction of MEF services in an SPBV network. The Base VID defines the service association, so that E-LINE and E-LAN services each map to a single Base VID that will have two or more SPVIDs associated with it, equal to the number of Provider Bridges participating in the VLAN. E-TREE requires the use of two Base VIDs, both associated with the same ECT algorithm such that congruence is preserved in order for shared VLAN learning to work. One Base VID defines the set of leaf to root paths, and the other Base VID the set of root to leaf paths.

SPBM can exploit the "all-pairs shortest paths" computation more fully, by building per service multicast trees which are each a strict subset of the multicast tree rooted on the node hosting an instance of the service.

After SPBM nodes complete their "all-pairs shortest paths" calculation, if two nodes require just a simple E-LINE service, the computation will result in the installation forwarding state on all nodes between the two nodes on the shortest path. Essentially, SPBM will create a p2p connection for that service.

In SPBM, if a third member (node) is now added to the service, transforming it from an E-LINE into an E-LAN, SPBM will automatically compute and create forwarding state for this service instance from each member to the other two members along shortest paths.

The E-TREE service deserves special discussion because SPBM solves this in a very simple manner. When a node advertises that it has a member of a service instance, it indicates whether that member will be a transmit-only, a receive-only, or a transmit–receive member of that service. This allows "split-horizon" behaviors to be created.

An SPBM E-TREE service instance is therefore formed by using two I-SIDs in a direct analogy to QinQ's asymmetric VID. On one I-SID, transmit-only "spoke" members can send only to "hub" members which are receive-only. On the other I-SID, "hub" members are receive + transmit (so that they can communicate with each other as well as with spokes), and all "spoke" members are receive-only. The SPBM computation algorithm takes the transmit/receive attribute into consideration and uses it to create the unidirectional state as required between the members, thus ensuring that they cannot violate transmit- or receive-only constraints placed on them by the operator, and that bandwidth consumption is minimized. Very simply, when a node finds itself on the shortest path between two nodes with an I-SID in common it will check the transmit/receive attributes for both ends for each direction. If there is a transmit attribute at one end and a receive attribute at the other, the destination multicast MAC for the transmitter is constructed and installed in the FDB. At the edge of the network, the I-components perform the functional equivalent of "shared VLAN learning" for the mapping customer MAC addresses to backbone MAC addresses in order for the bridging aspects to operate properly.

A topic of concern in the industry is the notion that frequently a BEB or MPLS/VPLS-PE will host a root instance and separate and distinct leaf instance(s) for a given E-TREE. This can be a frequent occurrence in networks with a significant backhaul component which also has service grooming capability. This is actually a more subtle problem than the description above would indicate, as the choice to colocate the root and leaf instantiation on a common subsystem is both an implementation issue and/or an operational practice problem. When this scenario occurs for SPBM, the virtual switch instance (VSI) hosting the combined root and leaf functionality will simply advertise transmit and receive interest in both I-SIDs. The internal implementation will determine the steering of frames to and from the root and leaf ports on the basis of the I-SID received for egress traffic, or inferred from port/VID attributes for ingress traffic, and the rest of the SPBM network will simply fill in the proper connectivity to the other roots and leaves. Where the node implements the VSIs in different subsystems, and the implementation cannot "hide" that fact, the node will be obliged to advertise different B-MACs (identified as "port MACs" earlier) in order to correctly steer frames within the BEB.

A consequence of the generalized provisioning and fulfillment model is that SPBM allows single-point or one-touch provisioning—the "Holy Grail" so to speak of multicast/E-LAN service solutions. An operator may add a new member to an E-LAN service by configuring only that new member, without disruption to the existing members and with a fulfillment time corresponding roughly to the network convergence time. The distributed nature of SPBM computations and the piggybacking of service information in the same protocol as is used to distribute topology mean that SPBM can perform all of its functions without additional provisioning or protocols.

There are of course numerous other permutations possible with these service attributes. For example, a unidirectional E-LINE is just two nodes, where one advertises transmit-only membership while the other advertises receive-only membership. Likewise, one can create more elaborate communities with, say, two transmitters and multiple receivers to allow for redundant head-end transmitters. The mechanism employed by SPBM to form service instances is simultaneously elegant, simple and powerful, and highly scalable. This is fundamentally because a calculation is far simpler and faster to perform than signaling or other message-based solutions to create an individual service instance.

How Symmetry and Congruence Are Preserved

The challenges of using a simple shortest path computation involve ensuring that both unicast and multicast traffic are kept on the same shortest path, and that the chosen path is the same in the forward and reverse directions for both unicast and multicast traffic, even when there are multiple equal cost candidate paths available. In essence, the connectivity across a network between any two points in a converged network should behave like a bidirectional p2p link.

This absolute symmetry (which is an inherent and desirable property of STP) is very important because without it, many of the Ethernet OA&M mechanisms lose their value. Further, the overlaying of client bridging on this infrastructure avoids misordering of frames and race conditions between unicast and flooded unknowns. If traffic is spread around, this would no longer be true. There are other important benefits to this symmetry and determinism, discussed below.

The solution to this challenge is to use a shortest path tree not only for the unicast routes, but also for the multicast routing from a given source. The end result is that each node in the network has its own source tree, which originates from itself and reaches all other nodes. When equal paths on any tree are resolved by the deterministic tie-breaking algorithm described below, all nodes will then choose the identical source tree rooted on any node. Now, instead of knowing about just one spanning tree, as is the case with the existing Ethernet STP, in SPB each node knows about one shortest path tree from every node in the network (in the multicast world, this type of per-source tree is referred to as an (S,G) tree, whereas a single tree for all members of the community (e.g., as constructed by the STP protocol) is referred to as a (*,G) tree. It should be noted in the case of a (*,G) tree that Ethernet split horizon forwarding ensures a sender does not get a copy of a multicast frame it has sent to the group looped back to it, a property which is essential for the prevention of loops. IEEE 802.1aq preserves this property.

Using these trees, every transit node in the network can easily forward unicast traffic along the shortest path simply by hop-by-hop destination lookup, and every node can multicast or broadcast traffic along the same route as the unicast traffic as long as it knows which node originated the multicast or broadcast frame.

Tiebreaking

Frequently, the shortest path between a source and destination in a network is not unique. There may, in fact, be dozens of equivalent shortest paths between a source and destination. SPB requires that for a given SPT set, every node agrees upon the same one of these paths and that determination must be made by each node without reference to other nodes, because no signaling is used. This requirement is met by a symmetric tie-breaking algorithm which, when executed by every node in the network whenever offered a choice of shortest paths, still results in a network-wide consistent decision as to which end-to-end path is chosen. The determinism of SPB has the added benefit of allowing accurate prediction of exactly where traffic will go, using for example an offline network planning tool.

The requirements on the distributed tie-breaking algorithm can be reduced to needing independence of the computation order, and independence of the network position of the computation. Each bridge has a unique Bridge ID. A path ID is specified as the lexicographically sorted list of the Bridge IDs which the path traverses, which is therefore also guaranteed to be unique. This satisfies the requirements for distributed tiebreaking, because all nodes find the same paths between any two endpoints, and the sorting process relies only on the values of the Bridge IDs in the path, which is not dependent on the order of computation. Thus, all nodes implementing the same logic choose the same path from the available options, for example, the one having the lowest path ID after ordering the Bridge IDs from lowest to highest in the path ID.

This algorithm has a further benefit from a computational perspective. The sorting algorithm results in the property that any segment of a shortest path is also itself a shortest path. As a consequence, as soon as multiple shortest paths forming a segment of a longer path have been resolved, all the state associated with the rejected paths can be discarded because it is known that it will never be required again. This simplifies the computation, since the amount of state to be carried forward as a Dijksta calculation progresses across the network can be continually pruned.

Finally, it is easy to see that there is a dual of the algorithm above, in which the selected path is the one having the highest path ID following highest to lowest ordering of Bridge IDs in the path ID. Although not guaranteed to find a diverse path if one exists, this technique is weighted toward doing so; as such it forms the first example of the techniques for load spreading across multiple equal cost paths which are introduced below and discussed at more length in “Load Spreading: Equal Cost Trees” (p. 119).

Exploiting Multiple Paths in SPB

Shortest path forwarding enables an inherent improvement in utilization of a densely meshed network, because all links can be used, and none need be blocked for loop prevention. It is possible to get even better utilization by allowing the simultaneous use of multiple equal cost shortest paths. The IEEE 802.1aq standard initially supports up to 16 different shortest paths between any pair of endpoints and provides an extension mechanism to permit future enhancements to be supported. This is achieved by manipulation of the tie-breaking criterion used to select between multiple equal cost shortest paths.

The previous section on Tiebreaking (p. 27) showed that consistent distributed path computation can only be achieved if all bridges make the same path choices, and a deterministic algorithm to do this was introduced. This can be extended further, and by using a set of globally defined transformations of the Bridge ID prior to sorting to form the path ID for each forwarding plane, different path sets are selected and each set is associated with a single VLAN (forwarding plane), and hence a set of SPVIDs for SPBV, and a single B-VID identical to the Base VID for SPBM. IEEE 802.1aq specifies 16 Bridge ID transformation methods, using the XOR of the Bridge ID with a “well-known” set of masks, and in addition makes possible the definition and application of further tie-breaking methods in the future.

The ECTs have unique attributes. First, the path congruence property means that IEEE 802.1aq actually supports equal cost routing for multicast and broadcast traffic as well as unicast. Second, since these are **end-to-end** paths, and not a hop-by-hop spreading function; assignment of traffic to a path is done once, at the ingress to the network, and so the operator can avoid random assignment, and instead place the traffic based on estimates of loading. This can be viewed as a low-overhead type of traffic engineering, in which services are not individually microengineered, but are assigned to a specific forwarding plane.

SPB TECHNOLOGY: LOOP AVOIDANCE

The SPB Approach: How It Works

Moving from spanning tree to a distributed routing system and mesh connectivity enables a mechanism to address transient loops at a much finer granularity than port blocking. For unicast forwarding, a routing loop is at best an inconvenience and at worst a chronic drain on bandwidth. For multicast forwarding, looping can be catastrophic, especially if a loop feeds back into another loop, resulting in an exponential increase in the bandwidth consumed in the network, in turn causing nearly instantaneous network "meltdown."

The combination of bidirectional symmetry and unicast/multicast path congruency between any two SPB nodes means that the FDB already has sufficient information to suppress most loops. In a stable and converged SPBV network, a frame from a given root can arrive only at ports which are on the shortest path to that root, which are also the only valid ports of receipt for the SPVID associated with the root. Similarly for SPBM, a frame **from** a given MAC address should only arrive at a port which is also on the shortest path **to** that same MAC address. A frame from a given root arriving at an unexpected port is an indication of a problem and potentially a loop. For SPBV, port membership of the VLAN (i.e., SPVID) in question determines whether the frame is valid. SPBM requires a simple modification to Ethernet source learning to convert it into a "reverse path forwarding check" (RPFC). RPFC simply checks the port of arrival of a frame against the expected egress port for that frame's source MAC address; frames arriving on an unexpected interface are silently discarded.

It can be shown that although the addition of VID or MAC-based RPFC substantially improves protection against loops in a routed environment, it cannot guarantee to eliminate a loop under some pathological conditions. It is nonetheless worth pointing out that even if such a loop does form, RPFC ensures that it must be "closed," preventing any new frames entering the loop, and thus guaranteeing that an exponential increase in the number of looping frames cannot occur. This is because RPFC allows any node to receive frames from only a single ingress port, so flows from two or more directions can never merge.

Nonetheless, it was desired that a technique offering absolute assurance against loop formation should be available in SPB. Such a technique has been specified, which uses the IS-IS topology database at a bridge to identify potentially hazardous changes, and to trigger topology database synchronization with neighbors before forwarding state associated with such hazards is installed.

A high level summary of this approach is that when an SPBV node receives a topology change, it

1. computes its unicast topology and FDB;
2. when it then determines that the shortest path distance to a root node has changed, so that there is a possibility a loop could form, it blocks the SPVID(s) associated with that node on all its ports;
3. it unblocks the SPVID entries related to changed trees (removed at step 2 above) on each port only when it knows that its control plane is synchronized with the neighbor connected to that port.

Similarly for SPBM, when an SPBM node receives a topology change, it

1. computes its unicast topology and FDB;
2. when it then determines that the shortest path distance to the root of a multicast tree has changed, so that it is possible a loop could form, it removes the multicast FDB entries related that to tree at the same time as it installs the updated unicast FDB entries;
3. the node can then install multicast entries for trees which have changed, but which are considered "safe" (because the shortest path distance to the root of that tree is unchanged), without reestablishing topology agreement with its neighbors, as the existing relationship has not changed;
4. the node installs the multicast FDB entries related to changed trees (removed at step 2 above) only when it knows that its control plane and unicast FDB state is based on a view of topology which is synchronized with its neighbors.

It should be noted that this process is less intrusive on the network in SPBM due to its use of per service multicast (rather than broadcast) trees, and due to its ability to distinguish unicast from multicast MAC forwarding, and so to give unicast different treatment.

The test above, for **any** change of distance to the root of the tree, is more aggressive than is strictly necessary, but it still ensures that trees unaffected by a fault carry on forwarding. This test has the major attraction that it is a nodal test, and is trivial to implement, because it is a simple per tree comparison of "before" and "after" distances to the root, and does not require per port calculation.

The optimal test is more complex to implement, but the actual requirements for handshaking with neighbors are still fairly straightforward:

- If a node believes that it has moved closer to a given root, it needs to handshake with its (new) parent node on that tree before installing the affected multicast entries. This ensures that the node's new parent is guaranteed to be closer to the root than the node itself;
- If a node believes that it has moved further from a given root, it needs to handshake with any child nodes on that tree whose last known (synchronized) distance from the root was closer than this node now believes itself to be. Only then may multicast be re-enabled on the port connecting to a child node. This ensures that the children still remain further from the root than this node.

This approach has a number of desirable properties. First, we maintain uninterrupted connectivity for multicast trees unaffected by the topology change, which exploits a key inherent property of link state protocols. Second, synchronization of multicast updates does not need to be ordered from the root; nodes can safely reinstall affected state as soon as they are synchronized with relevant peers, because if a peer has not achieved the required synchronization further up the tree, its own lack of installed multicast state "protects" the downstream nodes. Finally, the delay which synchronization would normally be expected to incur is largely eliminated, as the required handshaking with peer nodes can be done in parallel with the computation of the multicast FDB.

Synchronization with neighbors involves the exchange of a digest of the current IS-IS database, in order that both nodes can agree on the network topology they have used when computing loop-free forwarding paths (by removal when required of multicast state that was at risk of causing a loop). The principle is that if their exchanged digests match exactly, then they must also both agree on the distance to all roots. The digest is constructed in a way that maximizes how authoritative the comparison is, and also minimizes the overall computation. This is possible because the requirement is simply to determine whether two nodes agree on their topology model; if they differ, they differ, and there is no need for the digest to allow differences to be resolved, because a node determines what multicast state should be removed by the differences between its **current** view of the topology, and the topology view it held when it **was** last synchronized.

Summary of Topology Digest Construction

The requirements which must be met by the Topology Agreement Digest are:

- to summarize the key elements of the IS-IS link state database in a manner which has an infinitesimal probability that two nodes with differing databases will generate the same Digest;
- to have a very low incremental computation overhead because in general, link failure and repair are isolated events, and so it is very desirable that a single event should not require complete recomputation.

To achieve this, the Topology Agreement Digest field comprises six elements:

- the Digest Format Identifier
- the Digest Format Capabilities
- the Digest Convention Identifier
- the Digest Convention Capabilities
- the Digest Edge Count
- the Computed Topology Digest

The first four fields are provided to preserve extensibility, allowing development of alternative digests in the future if required, and will not be discussed further here.

The Digest Edge Count is a 2-byte unsigned integer. Its purpose is to offer a summarization which is simple to compute and powerful in detecting many simple topology mismatches. In the light of the use of a strong hash for computation of the Computed Topology Digest, the Edge Count can be seen as a historical hangover from a time when a simpler multiplicative hash was envisaged.

This value is the sum modulo 2^{16} of all edges in the SPB region. Each p2p physical link is counted as two edges, corresponding to its advertisement by IS-IS in an LSP flooded from either end of the link.

The overall procedure for constructing the final component, the Computed Topology Digest, is to:

- form a signature including every edge in the topology by computing the MD5 hash (RFC 1321) of the significant parameters of the edge, as defined below;
- compute the Digest as the arithmetic sum of all edges in the topology.

Although MD5 is widely reported to be cryptographically compromised, this is not relevant in this application because there is no motivation for an attack. What is required is a function exhibiting good avalanche properties such that signatures with potentially very similar input parameters have an infinitesimal probability of collision.

This strategy also allows the Computed Digest to be incrementally computed when the topology changes, by subtracting the signatures of vanished edges from the Digest and adding the signatures of new edges.

The input message to the MD5 hash for each edge is constructed by concatenating the following fields in order:

1. the Bridge Identifiers (Bridge Priority || Bridge SysID) of the two bridges advertising the edge, with the numerically larger Bridge ID first;
2. one 3-tuple for each MTID declared in IS-IS. The 3-tuples are declared in descending order of MTID value, with the largest MTID declared first.

Each 3-tuple is constructed by concatenating the following fields in this order:

- MTID value || Link Metric of higher Bridge ID || Link Metric of lower Bridge ID

If an edge is not present in a topology, its SPB Link Metric is set to zero in that topology.

The value of the Computed Topology Digest is the arithmetic sum of all of the signatures returned by presenting every edge message to MD5, treating each signature as an unsigned 16-byte integer and accumulating into a 20-byte integer. Every physical link is seen as two edges, one advertised in an LSP by each bridge comprising the adjacency, and formally the Computed Topology Digest includes both.

SUMMARY

In summary, SPB applies state-of-the-art link state routing to Ethernet forwarding, with the intent of providing a robust and efficient any-to-any infrastructure. In its SPBM incarnation, this is used to support client-server hierarchy to deliver perfect virtualization of traditional enterprise Ethernet, in which virtual LAN segments defined by the I-SID replace physical facilities dedicated to each LAN.

So far we have discussed how SPB functions as it is currently specified. In the next sections we will explore some of the background behind the journey to this point, and provide some insight into the design decisions described to date.