IEEE 802.1 Bridging and IEEE 802.17 Steering

Norman Finn, Cisco Systems

1.0 Point-to-point Links

Over the last few years, it has become common to run more than one spanning tree among a set of bridges. System administrators typically assign different VLANs to different spanning trees. Such usage has been non-standard. However, 802.1 is now working on a new document, IEEE 802.1S, which will standardize this common practice.

To illustrate one usage, consider Figure 1, where we have four bridges (B1 through B4) running two VLANs, “red” and “blue”, and two spanning trees, one for each VLAN. There are two end stations (X and Y) connected via these bridges.

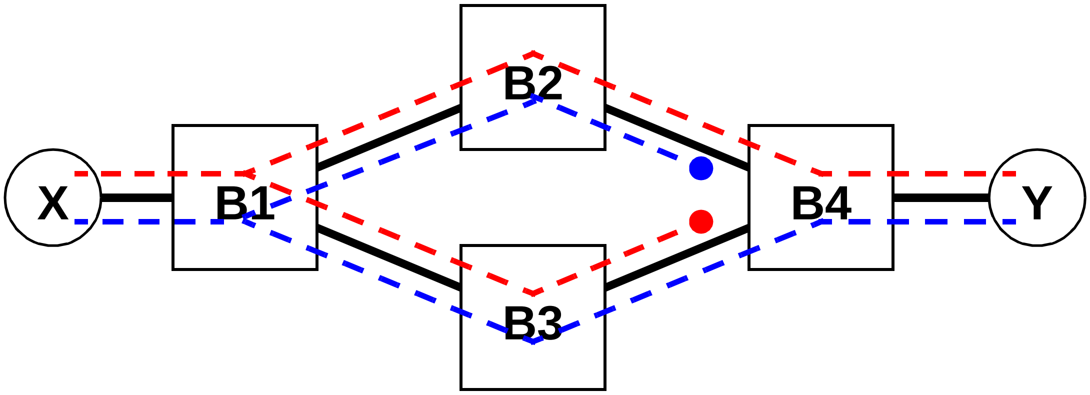

FIGURE 1. Point-to-point links, Separate Databases and Spanning Trees

The solid black lines represent physical wires. The dotted colored lines represent the virtual paths, red and blue, available to the two VLANs. You will notice that B4 blocks the blue VLAN’s spanning tree coming from B2, and that B4 blocks the red spanning tree coming from B3. Thus, the path between X and Y on the red VLAN passes through B2, while the path between X and Y on the blue VLAN passes through B3. We are assuming, here, that both X and Y have a presence on both VLANs. For example, they might both be routers.

Consider bridge B1, both when it is learning about station Y, and when it is making decisions about how to direct frames towards station Y. It will receive frames from station Y on the red VLAN from B2, and on the blue VLAN from B3. When directing frames to station Y on the red VLAN, it must forward them to B2, whereas it must forward blue frames to B3. Thus, bridge B1 must maintain two independent MAC address databases, one for the red and one for the blue VLANs. (Equivalently, it must maintain a single database using the pair {VLAN, MAC address} rather than simply {MAC address}, as the key.)

What would happen if B1 were to (erroneously) use a single database for both VLANs? Suppose it has a blue frame to deliver to Y. If the last frame it received from Y was blue (and necessarily, from B3), then it will direct the frame towards B3, and all will be well. If, however, it last received a red frame from Y, (necessarily from B2), then it will send that blue frame towards B2. This blue frame will be discarded, because the blue spanning tree is blocked by B4; it will never reach Y. If we assume a constant bidirectional stream of data on both VLANs, as would be the case if X and Y were routers operating on two VLANs, then we would expect roughly half of all frames to be misdirected and lost. Depending on the details of transmission order, a bridge making this mistake could cause either no frames, or almost all frames, to be blackholed.

Now consider a second case, in Figure 2. In this case, there are still two VLANs, but now there is only a single spanning tree, and the two VLANs share a common database. In this network, all frames transmitted by Y to X use the red VLAN, while all frames transmitted by X to Y use the blue VLAN. (In Figure 2, B4 blocks its link to B2 in order to break the loop.)

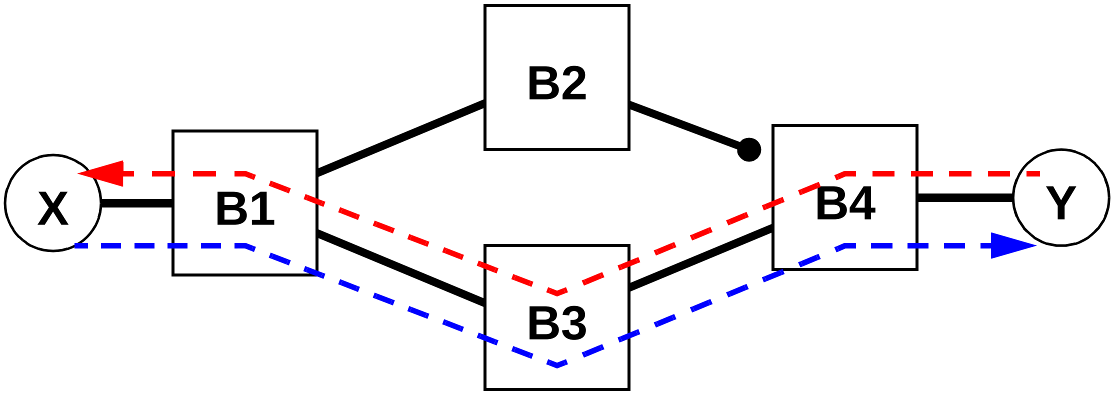

FIGURE 2. Point-to-point links, Single Database and Spanning Tree

Again, consider B1. Since all Y-to-X frames use the red VLAN and all X-to-Y frames use the blue, B1 must be configured to place both VLANs in the same MAC address database. It must ignore the VLAN-ID, and use only the MAC address, as the key to this database. It learns from the red frames that Y's MAC address lies in the direction of B3, and forwards blue frames destined to Y on that basis.

If the bridges (erroneously) keep their red and blue databases separate, then they will never learn Y's MAC address on the blue VLAN, because Y never transmits on the blue VLAN. All bridges will always flood X's frames to Y on all links. Symmetrically, all of Y's frames to X will be flooded everywhere, because X never transmits on the red VLAN. Bandwidth is wasted.

The example of Figure 2 is perfectly allowed by the current published standards. That of Figure 1 is quite common in today's practice, and will be made standard by IEEE 802.1S.

The way that a typical existing bridge implements the ability to use either separate or combined databases is to organize its MAC address database using keys consisting of the pair, {FID, MAC address}, where "FID" is a "Filtering database ID". Typically, the FID is obtained from a 4k x 12 bit table which translates VLAN ID to FID. Less-capable mapping techniques are allowed, although a less-capable bridge may not be able to operate in networks in which a bridge with the full table could operate. In the example of Figure 1, all bridges' translation tables would map the red and blue VLANs to different FIDs, while in the example of Figure 2, all would map both VLANs to the same FID.

2.0 IEEE 802.17 Rings

Let us now transfer the example of Figure 1 to the world of IEEE 802.17, replacing the two point-to-point links B1-B2 and B1-B3 with an IEEE 802.17 ring which emulates a shared-medium Ethernet. (See Figure 3.) While we're at it, let us remove bridge B1, placing the router (or host) X directly on the ring. Assuming that X wants to steer frames in the optimum direction around the ring, it is X's, B2's, and B3's abilities to steer frames correctly that is in question, rather than the bridges' abilities to use forward along point-to-point links.

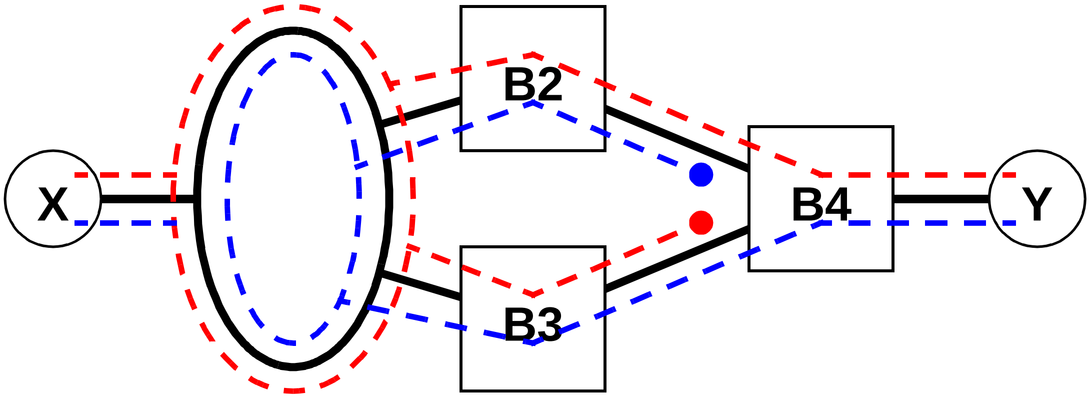

FIGURE 3. IEEE 802.17 link, Separate Databases and Spanning Trees

The first thing to notice is that, as far as bridges B2 and B3 are concerned, they are connected to each other, and to X, via the ring ports. We also notice that X does not run any spanning tree algorithm; it is a router.

However, just like B1 in Figure 1, X must maintain separate databases for the red and blue VLANs. On the red VLAN, Y's frames come from B2, and should be steered towards, B2. On the blue VLAN, they come from, and should be steered towards, B3.

What happens if X keeps a single database instead of two? That depends on whether the ring is whole, or whether it has broken between B2 and B3. If the ring is unbroken, then as for Figure 1, the erroneous single database causes (typically) half of the frames sent by X to be sent in the wrong direction. Unlike the Figure 1 case, in Figure 3 these misdirected frames are not lost. However, they may well be delivered out of sequence, especially if the data rates are high, and if there are additional stations on the ring between B2 and B3. Frequent misordering of frames causes certain Layer 2 protocols to fail completely. It also can cause IP traffic to take sub-optimal (software) paths in a receiver, rather than the optimal (hardware) paths used by properly sequenced traffic. Admittedly, Layer 2 protocols which require ordering are becoming increasingly scarce, but the frequent misordering of IP traffic can cause significant loss of bandwidth or significant increase in software overhead, especially on high-bandwidth data streams within an enterprise network.

Of course, if the ring is broken due to a malfunction, as in Figure 4, then correct steering is required to reach the destination, and X's failure to have separate red and blue databases will cause the network to suffer exactly the same fate of blackholing large numbers of frames that was seen in Figure 1, when B1 did not use separate databases.

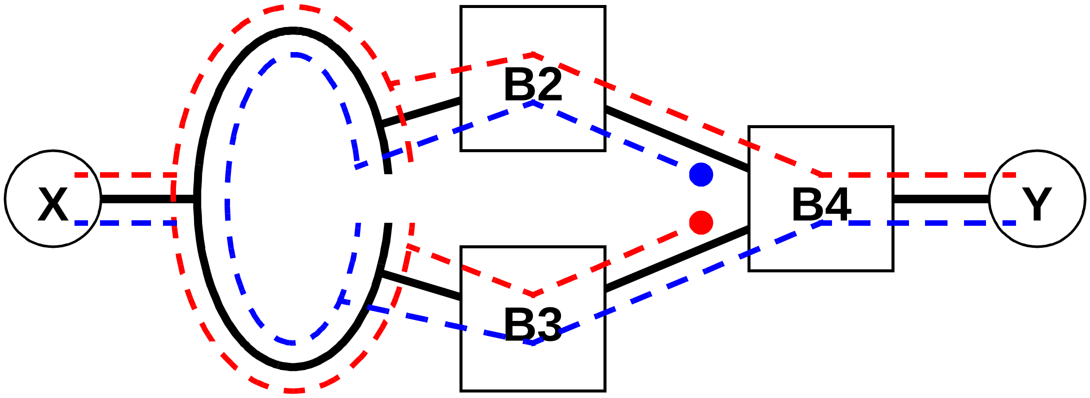

FIGURE 4. Broken IEEE 802.17 link, Separate Databases and Spanning Trees

In Figure 5, we see the case from Figure 2 transferred to a ring. The penalty for X erroneously keeping the VLAN databases separate is still considerable extra flooding. As before, X will flood all frames if it keeps separate databases, as it never learns Y's address on the blue VLAN, but since it is flooding, B2 will see all of the X-to-Y frames. Since B3 is properly configured to always steer towards X, B2 never sees any Y-sourced frames, and so never learns Y's MAC address. It therefore floods all X-to-Y frames to all of its ports. At least, any additional bridges on the ring between X and B3 would be immune from X's misconfiguration.

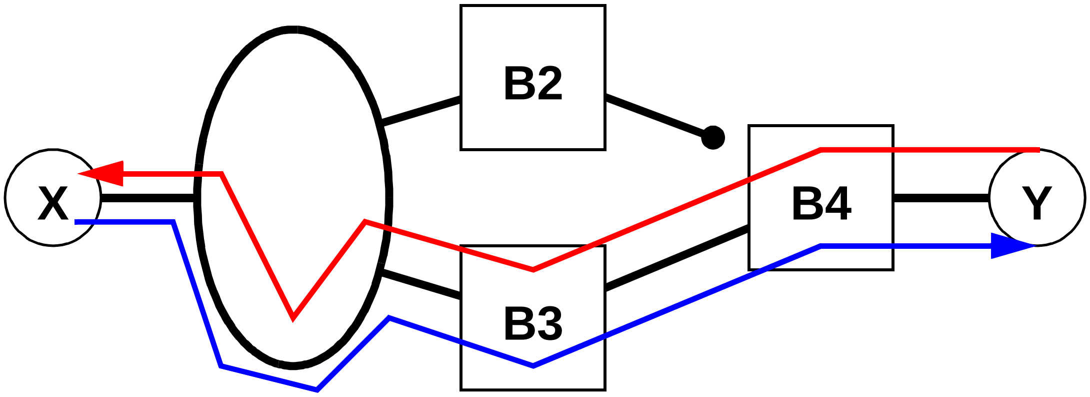

FIGURE 5. IEEE 802.17 link, Single Database and Spanning Tree

If the ring is broken between B2 and B3, the situation is as in Figure 4, because no frames were flowing across that part of the ring. X's flooding still annoys those who connect to B2.

3.0 Spanning Tree Topology Changes

Another imposition is made upon X due to the fact that X is attempting to steer frames in the presence of bridges. Suppose we have a network similar to that in Figure 5, but let us ignore VLANs, and limit ourselves to a single spanning tree. B4 blocks the link to B2 in order to open the physical loop (B2, B3, B4). The network has been running for some time. Now, the link between B3 and B4 breaks. The result is shown in Figure 6; B4 has unblocked the B2-B4 link to restore connectivity.

X had been steering frames for Y towards B3. Suddenly — IEEE 802.1D can now reconverge in milliseconds after a link failure — X must steer towards B2, instead of B3, to reach Y. If it is X's turn to speak at the time the network reconfigures, then Y will not send a frame for X to learn, and X will continue blackholing the frames for Y until its steering information times out.

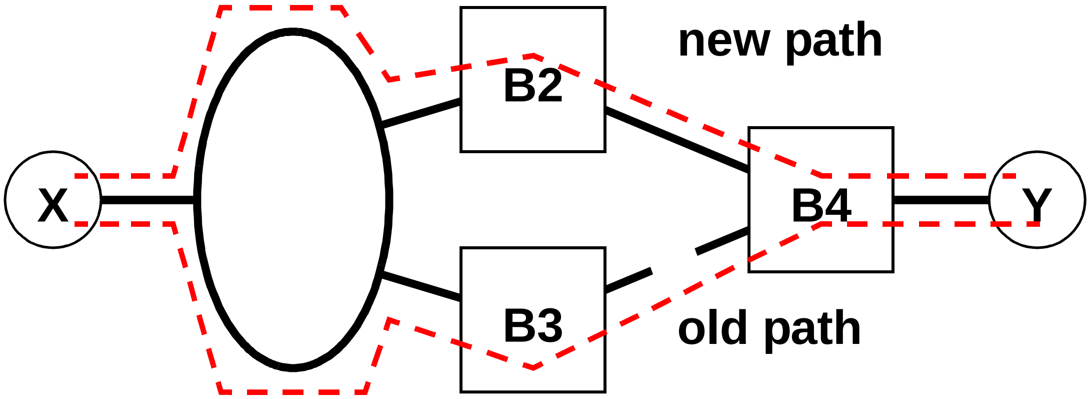

FIGURE 6. IEEE 802.17 link, Spanning Tree Topology Change

In order to avoid this situation in the equivalent Ethernet topologies, bridges transmit Topology Change Notices to tell each other that certain MAC addresses must be forgotten. Traffic destined for those addresses is then flooded. Connectivity is thus established immediately after the topology is repaired. Flooding is quickly suppressed as the flooded frames' MAC addresses are relearned, perhaps on different links than before the break.

In order to avoid an unnecessary loss of connectivity after the bridges reconfigure the network in Figure 6, X must take advantage of the Topology Change Notices transmitted on the ring by B2 and/or B3. When notified, X must take the appropriate actions to "forget" learned MAC addresses in a manner very similar to those taken by a bridge. X does not need to transmit Topology Change Notices. It will then flood those forgotten addresses until it can relearn there new locations.

It is a question for IEEE 802.17, whether to have X listen for Topology Change Notices, or whether the bridges (or the bridges' ring ports) should detect the transmission of Topology Change Notices, and send some sort of special IEEE 802.17 control packet to notify X. Considering the number of variants of current and future IEEE 802.1 standard bridging protocols, not to mention the number of non-standard spanning tree protocols in use, the latter choice might be the better one. The bridges and/or their ring ports are required to have knowledge of these variants and their packet formats; a router (X) and/or its ring port may be harder to teach. Of course, if the bridge notifies its port of a TCN transmission, rather than letting the port detect the transmission automatically, then the API between the port and the bridge must accommodate the notification.

4.0 Symmetrical Steering and Flooding

An interesting problem arises from Figure 5, if X and B3 reach other asymmetrically when steering or flooding. In Figure 7, X and B3 both steer towards each other in the counterclockwise direction. This would, perhaps, be more probable if there were an additional ring station between X and B3. Note that the problem of separate or combined databases is irrelevant; all frames in this example are assumed to be on the same (red) VLAN. The problem is that B2 never sees frames from which to learn X's MAC address. As a consequence, it floods all of the Y-to-X frames to all of its ports. Similarly, any bridges between X and B3 would flood all of their X-to-Y frames to all of their ports, having never seen a frame with Y's source MAC address.

We may conclude, therefore, a restriction on steering and flooding on 802.17 rings. For any given pair of stations, traffic in both directions between those two stations must use the same "side" of the ring. That is, one must forward clockwise, and the other counterclockwise, around the ring.

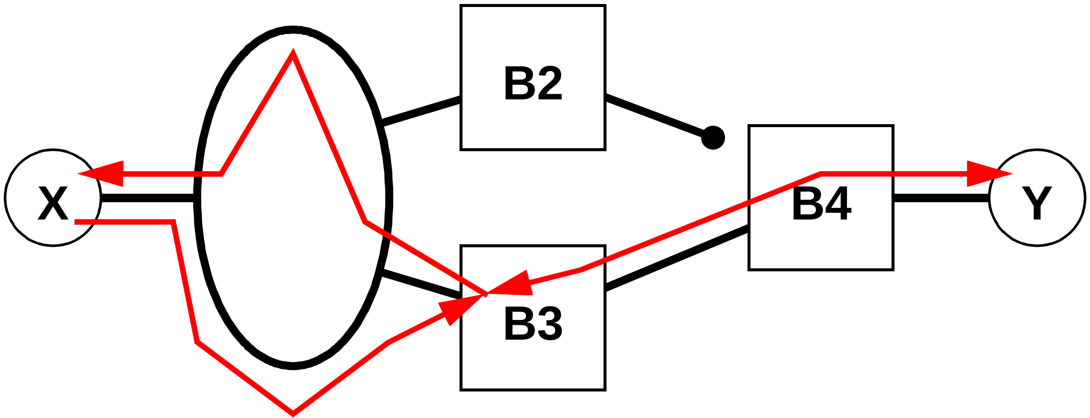

FIGURE 7. IEEE 802.17 link, Asymmetric Steering

5.0 Summary of Requirements for Flooding Ports

If a station's port floods all frames, instead of steering any, then it can ignore all of the above requirements. If all ports flood, and do not steer, then all ports learn as much as possible, no frames are blackholed, and no frames are flooded needlessly beyond the ring, through the connected bridges.

6.0 Summary of Requirements for Steering Ports

If a station's port on a ring is learning information which relates MAC addresses with specific directions around the ring, and using that information to steer some frames in the right direction, then that port must perform many of the functions of an IEEE 802.1 bridge (or a non-standard bridge, if a non-standard spanning tree algorithm is used). These requirements are independent of whether the port is attached to a bridge or not; they are required of all stations' steering ports. A bridge, of course, may find it easier to incorporate the additional ring information into its databases than a host computer finds it to create those databases. The functions required include:

1. (Section 2.0) A port must key its MAC address database on {FID, MAC address} pairs, where FID is a number derived from the VLAN-ID. Its VLAN-to-FID mapping facility must be configured identically to that of any bridges on the same ring.

2. (Section 3.0) A port must be made aware when any bridge on the ring transmits a Topology Change Notice, and take appropriate action to unlearn some or all of the MAC address information it has learned.
3. (Section 4.0) The path used by any ring station A to reach some other ring station B must pass through the exactly the same intermediate stations as the path used by station B to reach station A. This requirement does not distinguish between flooded and steered frames.

Issues related to interactions between a standard bridge's databases and its ring port's databases have not been addressed, here.

7.0 Mixing Flooding Ports and Steering Ports

Consider Figure 7, on page 6. Suppose that X performs steering, but B3 floods all frames to the ring without learning ring directions. Any bridges between X and B3 will be ok, but B2 will never see any X-to-Y frames, so it will flood all Y-to-X frames to all of its ports. Similarly, if B3 steers and X floods, bridges between X and B3 would perform unnecessary flooding.

Therefore, on any ring which contains bridges, the mixing of flooding ports and steering ports is not practical.