Spanning Trees and IEEE 802.3ah EPONs

Norman Finn, Cisco Systems

1.0 Introduction

The purpose of this document is to explain the issues that arise when IEEE 802.1 bridges, running the Spanning Tree Protocol, are connected to an IEEE 802.3ah Ethernet Passive Optical Network. Section 2.0 on page 1 describes EPONs and the emulation of point-to-point and shared LANs for the benefit of 802.1. Section 3.0 on page 2 provides an introduction to the IEEE 802.1D Spanning Tree Protocol for 802.3. It is a somewhat simplified model, for clarity, and includes some concrete examples. Section 4.0 on page 7 then explains the problems that IEEE 802.1D Spanning Tree Protocols have with EPONs. Conclusions are presented in Section 5.0 on page 12.

2.0 Ethernet Passive Optical Networks (for 802.1)

The IEEE 802.3ah (EFM, Ethernet in the First Mile) task group is examining Ethernet Passive Optical Networks, or EPONs. These devices use optical fibers to connect one OLT (Optical Line Termination) to some number (say, 8 or 128) of ONUs (Optical Network Units). Any signal transmitted by the OLT passes through passive optical elements which split (replicate) the signal and deliver the same thing to all of the ONUs. Upstream, the ONUs transmit one at a time. Each ONU's transmissions are received by the OLT, but not by any other ONU. This type of LAN may be called a "point-to-multipoint LAN". It operates point-to-point in the ONU-OLT direction, and point-to-multipoint in the OLT-ONU direction.

To date, all *other* 802.3 LANs have had the characteristic that any frame transmitted by any station on the LAN would, subject to the bit error rate, be received by all other stations on that same LAN. The special case of only two stations on a LAN is a "point-to-point" LAN. Where three or more stations are connected, it may be called a "multipoint-to-multipoint" LAN. The corresponding IEEE 802.3 names are "point-to-point LAN" and "shared LAN". The standard IEEE 802.1D, which specifies the operation of a bridge, takes only these two LAN types into account.

Recognizing this fact, 802.3ah has voted to include a mode whereby frames in both directions are tagged with an ONU identifier. At the OLT is an array of virtual MACs, one for each corresponding ONU MAC. (It has not yet been determined whether or not one ONU may incorporate multiple MACs.) The ONU and OLT MACs are paired, associated by means of the ONU identifier tag, and so emulate a bundle of point-to-point LANs. In the OLT-ONU direction, the upper layers' choice of virtual MAC determines the ONU ID. Each ONU filters ONU IDs, and passes only its own. By assigning each OLT MAC to a separate port of an IEEE 802.1D bridge, the IEEE 802.3ah EPON becomes perfectly compliant with IEEE 802.1D's expectations.

Similarly, the 802.3ah LAN can use the ONU identifier to emulate a shared LAN. In this mode, every frame transmitted by an ONU is marked with that ONU's ID. In addition to being passed up the stack when received by the OLT, every frame is reflected back towards the ONUs. The ONU that originated the frame recognizes its own ONU ID, and discards the frame. All other ONUs

pass it up their stacks. Frames transmitted by the layers above the OLT are marked with the OLT's "ONU ID", so that all ONUs pass it.

It is important to note that, to meet the definition of a shared LAN expected by an IEEE 802.1 bridge, the OLT must reflect *all* frames received from any ONU, not just multicast or broadcast frames, and not just unicast frames intended for another ONU. Bridges depend on the fact that, on a shared LAN, every station sees every transmitted frame.

If the ONU ID space is larger than that required to enumerate the ONUs, then the shared LAN emulation can be more capable, and hence, more complicated. For example, one could group a specific subset of ONUs together with a specific OLT virtual MAC (or even with no OLT virtual MAC) into one shared LAN, and emulate several such shared or point-to-point LANs on a single EPON. By such means, or by the addition of a mode bit, the LAN may emulate both point-to-point and shared LANs at the same time. At this time, IEEE 802.3ah has discussed all of these options, but has not decided to incorporate shared LAN emulation into the standard.

3.0 Simplified Spanning Tree Model (for 802.3)

In order to explain the issues problems associated with applying the IEEE 802.1D Spanning Tree Protocol (STP) to IEEE 802.3ah EPONs, we introduce a somewhat simplified view of STP, and apply it to a specific network as an example.

IEEE 802.1D assumes a network consisting of a number of point-to-point and shared LANs connecting some number of bridges and end stations. End stations originate and receive data frames. Bridges relay those frames from LAN to LAN in order to deliver each data frame to *at least* that station (or those stations) to which it is addressed. To the extent possible, bridges also attempt to conserve bandwidth by not transmitting any data frame onto any LAN unless it is necessary to reach the end station(s) to which it is addressed. When in doubt, a bridge prefers connectivity to bandwidth conservation, and errs on the side of transmitting the frame.

In order to accommodate failures, either of LANs or of bridges, most bridged LAN networks incorporate a degree of redundancy. That is, there are multiple physical paths available between the source and the destination stations for most data frames. If every bridge merely flooded all received data frames on all LANs, these redundant paths would result in each data frame being endlessly duplicated. To avoid such a "meltdown", an IEEE 802.1D bridge constructs a spanning tree over the physical topology.

The spanning tree is a logical topology constructed over the available physical topology, and changes as necessary to adapt to changes in the physical topology. This logical topology differs from the physical topology in that bridges decide to "block" certain physical ports, and refrain from transmitting or receiving data frames on those ports. The spanning tree is thus a subset of the physical topology. A spanning tree is "spanning" in the sense that every station on every LAN can reach every other station. It is a "tree" in the sense that there is exactly one path along the spanning tree (that is, through the non-blocked ports) that connects any two LANs or stations.

3.1 Bridge Protocol Data Units (BPDUs)

In the simplified spanning tree model of this paper, each bridge has a globally unique Bridge ID. Bridge IDs are 8 bytes in length in the real protocol; in this document, we will use small integers.

In the process of creating a spanning tree over a network, a Root Bridge is elected. This is the bridge which has the lowest numerical value of its Bridge ID.

The spanning tree also incorporates the idea of a “path cost”. Each bridge associates with each of its ports a cost to receive a frame on that port. These costs default to a value which is inversely proportional to the bit rate of the LAN. The total cost for a given path from bridge to bridge is equal to the sum of the costs for receiving the frame at each bridge.

The bridges construct the spanning tree by transmitting and receiving Bridge Protocol Data Units (BPDUs). There are three numbers in each BPDU which are of import in the simplified model of this document. These three numbers are arranged in a fixed order of significance into a priority vector, so that comparing priority vectors is mathematically equivalent to comparing integers. The three numbers are:

1. The Bridge ID that the transmitting bridge thinks belongs to the Root Bridge of the network.
2. The total Root Path Cost along the path from the Root Bridge to the transmitting bridge.
3. The Bridge ID of the transmitting bridge.

When comparing priority vectors, the numerically smaller vector is “better”. Since all Bridge IDs are unique, it is impossible for two bridges to build priority vectors that are equal. In the notation of this document, {r5, c110, i6} denotes a priority vector with a Root Bridge ID of 5, a Root Path Cost from the Root Bridge of 110, and a Bridge ID of 6. That is, bridge 6 thinks that it costs 110 units to get from the Root Bridge, 5, to bridge 6.

The exchange of BPDUs accomplishes three tasks simultaneously. These three tasks, taken together, ensure that the spanning tree is both “spanning” and a “tree”. They are:

1. Elect a Root Bridge.
2. For each LAN, elect exactly one Designated Bridge.
3. For each bridge except the Root Bridge, select exactly one port to link the bridge to a LAN that lies in the direction of the Root Bridge.

3.2 Bridge Info

Every bridge maintains a “Bridge Info” variable, which is a priority vector. This priority vector is transmitted in any BPDU that the bridge emits.

A bridge’s Bridge ID is configured. It is changed only by reconfiguring the bridge, a process which we will not discuss. The Bridge Info is derived from the Port Info variables (see Section 3.3), and reflects the best information about the Root Bridge that the bridge has learned from all its ports.

When initialized, the Bridge Info of a bridge with Bridge ID *b* contains a priority vector of {r*b*, c0, i*b*}, indicating that this bridge believes itself to be the Root Bridge of the spanning tree.

3.3 Port Info and Designated Bridge Election

Every bridge maintains a “Port Info” variable for each of its ports. The Port Info includes:

1. a priority vector, which represents the best information that the bridge has learned from, or is transmitting to, the port;
2. a Port Role, which is either Designated, Root, or Alternate;
3. a Port State, which is either Forwarding, Listening, or Blocked;
4. a Port Cost, which is the cost of receiving a frame on that port (a configured value); and
5. various timers.

The Port State determines the forwarding of data frames on the port. In the Forwarding state, the bridge is receiving, relaying, and transmitting data frames, as well as BPDUs, to and from this port. In the Blocked or Listening states, the bridge may be transmitting and receiving BPDUs, but allows no data frames either in or out of the port. The Listening state is a timed state used to transition from the Blocked state to the Forwarding state. It is required, because a too-sudden transition to Forwarding might cause the duplication or out-of-order delivery of data frames.

When initialized, the Port Info of a bridge with Bridge ID *b* contains:

1. a priority vector of {r*b*, c0, i*b*}, indicating that this bridge is the Root Bridge;
2. a Port Role of Designated, to be consistent with being the Root Bridge; and
3. a Port State of Listening, in preparation for becoming Forwarding, later.

As long as a port is in the Designated Port Role, the bridge periodically transmits BPDUs containing the priority vector from the bridge's Bridge Info. Each time a BPDU is received, the bridge compares the received priority vector to the port's priority vector. If the received vector is worse (numerically higher) than the Port Info priority vector, then the received BPDU is discarded. If the received vector is better, then it is recorded in the Port Info, and the Port Role is changed to Root or Alternate, as described in Section 3.5. (If recorded, further action may also be taken; see Section 3.4 and Section 3.5.)

In this manner, for any LAN connected to one or more bridges, the one bridge with the best priority vector becomes the LAN's Designated Bridge. That one bridge continues to transmit periodic BPDUs. The other bridges' ports on that LAN, upon receiving the Designated Bridge's BPDUs, take the Root or Alternate roles, and stop sending BPDUs.

As each BPDU is received, a timer is started. If no further BPDUs are received before the timer expires, then the Port Info priority vector is replaced with the Bridge Info priority vector, and the port reverts to the Designated role. The bridge(s) remaining on the LAN can then elect a new Designated Bridge.

A bridge port in the Designated role is never in the Port State of Blocked; it is either in the Forwarding state, or is in the Listening state on its way to the Forwarding state.

3.4 Root Bridge Election

As mentioned above, each time a bridge receives a BPDU with a priority vector better (numerically lower) than the one in the Port Info, it records that new priority vector in the Port Info. is In addition, the Root Path Cost from the received vector is incremented by the port's configured Port Cost, and the resultant priority vector is compared to the priority vector in the Bridge Info. If the new priority vector is better, then the new Root Bridge ID and Root Path Cost replace the ones in the bridge's Bridge Info. In addition, the new Bridge Info priority vector is compared to that in the

Port Info of every port on the bridge, and those whose information is replaced take the Designated role.

Thus, the best Root Bridge ID and Root Path Cost that a bridge receives are remembered in its Bridge Info. Since the Root Bridge ID is the most significant portion of the priority vector, and since the Bridge Info drives all BPDUs transmitted by the bridge, the best Root Bridge ID is propagated bridge by bridge throughout the network. The bridge with the lowest Bridge ID is elected Root Bridge, and its Bridge ID is known to every bridge in the network.

If the Root Bridge dies, it stops sending BPDUs. After a while, the bridges attached to the Root Bridge time out their Port Info, and each claims to be the Root Bridge. (Additional information is carried in the BPDUs so that the timers in all bridges expire at more-or-less the same time, rather than sequentially from the root out.)

3.5 Root Port Selection

On the Root Bridge itself, all ports are Designated ports. On any other bridge, at least one port is not a Designated port. Every bridge selects exactly one port which is not a Designated port, and gives it the Port Role Root. The Root port selected is the one with the best (numerically lowest) priority vector after adding the Port Cost to the received priority vectors. All remaining ports take the Alternate role. All ports in the Alternate role are placed in the Blocked Port State. The Root port is kept in the Forwarding state (or the Listening state heading to the Forwarding state).

3.6 Example 1: Point-to-point LANs

In this example, we will assume that all Port Costs are configured to be 10. In Figure 1, we show the Bridge Info priority vectors next to the bridge's name, and the Port Info priority vectors next to the port(s). Outside the box representing the bridge (usually), for each port, we show the port's Port Role (**D**esignated, **R**oot, or **A**lternate) and Port State (**B**locked, **L**istening, or **F**orwarding).

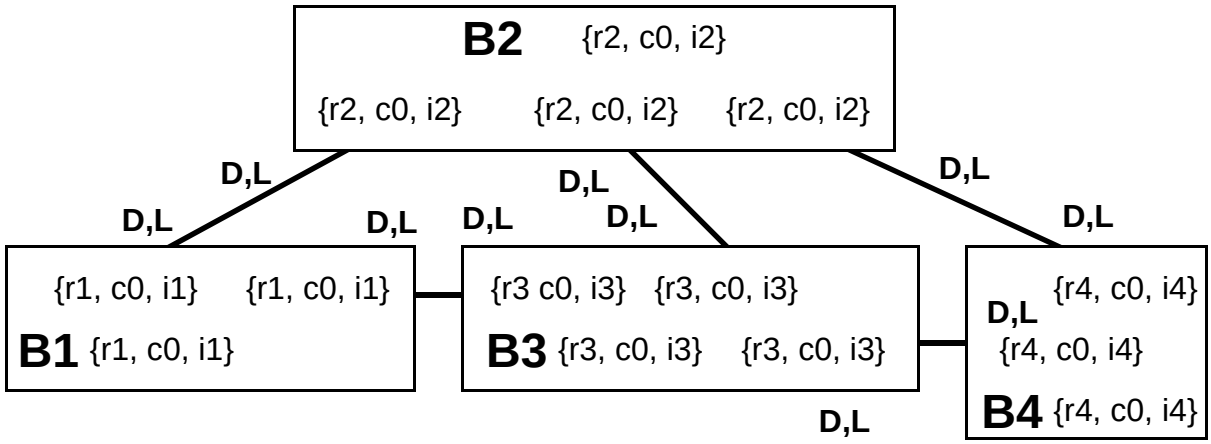

FIGURE 1. Point-to-Point LANs: Initialized state

Imagine that all four bridges transmit their BPDUs at once. The state after the first exchange of BPDUs, but before any bridge has compared the received BPDUs to its Bridge Info, is shown in

Figure 2. On each LAN, the better claim has prevailed. The losing bridge ports' Port Role and Port State cannot be determined until the Bridge Info and the other ports' Port Info is examined.

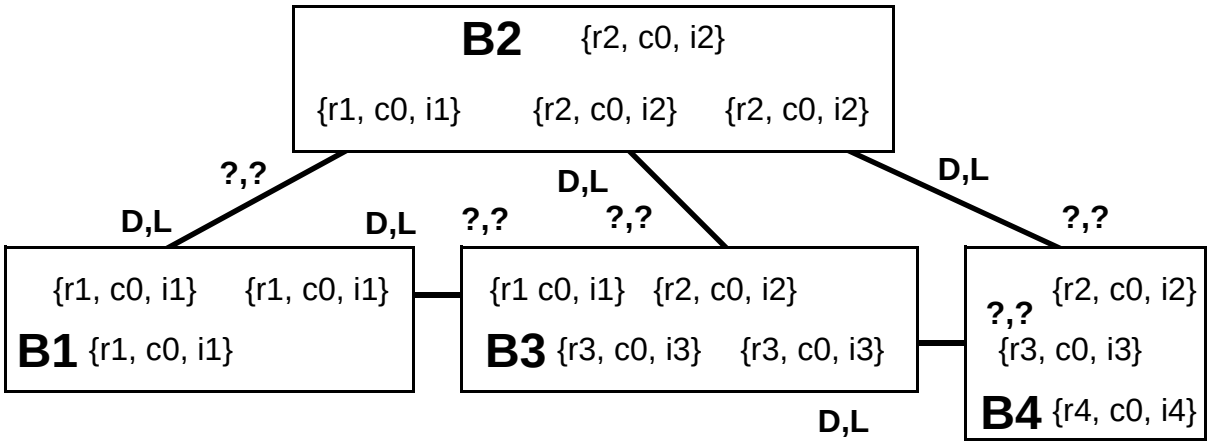

FIGURE 2. Point-to-Point LANs: After 1st BPDU exchange, before reconciliation

In Figure 3, we see the result of the first BPDU exchange. Notice that the Root Path Cost is 0 on B2's port from B1, but that port's Port Cost is added to the received value before it is propagated to the Bridge Info and to the other ports on B2. B3 has overridden the information received from B2 with the better information received from B1. Similarly, B4 preferred B2 to B3 as the root.

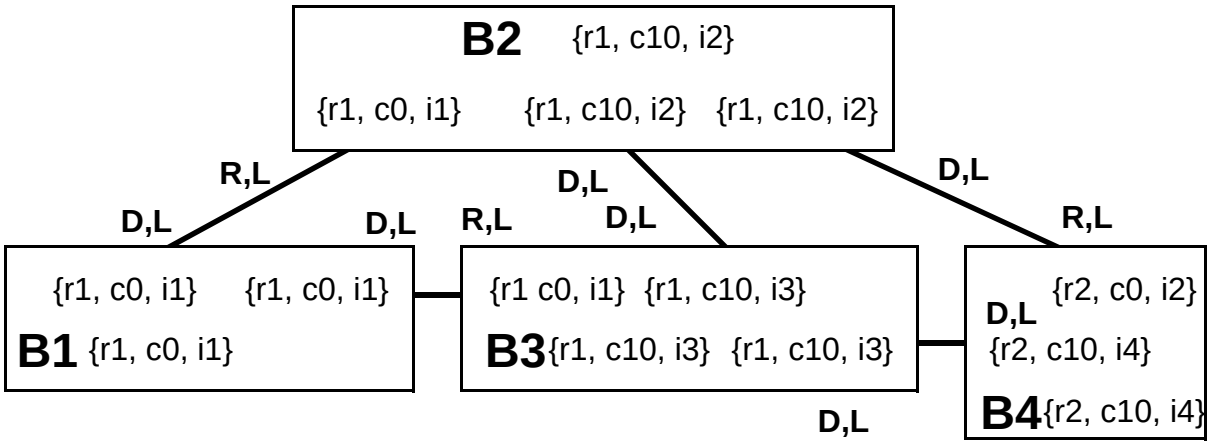

FIGURE 3. Point-to-Point LANs: After 1st BPDU exchange and reconciliation

After the second BPDU exchange, B2 and B3 come to an agreement on who is the Designated Bridge, and B4 has learned about B1. B3 selects B1, rather than B2, as its Root port, because the port towards B1 has the lower Root Path Cost (0 + 10 instead of 10 + 10). B4 has selected its port to B2 as root over the port to B3 based on the Bridge IDs of the senders; both ports are distance (10 + 10) from the Root Bridge. After the Listening states have transitioned to Forwarding, the final spanning tree is shown in Figure 4. Blocked ports are marked with a red bar for clarity.

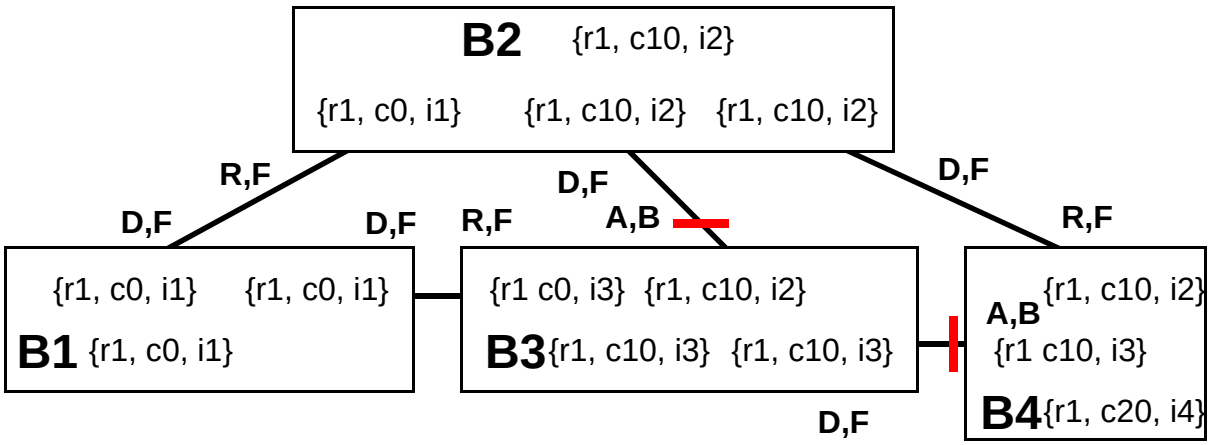

FIGURE 4. Point-to-Point LANs: Final state

4.0 Point-to-multipoint LANs

Let us delete some of the LANs in the example of Section 3.6 and add an EPON and a shared LAN. In Figure 5, we see the initial state. All Port Costs are again 10.

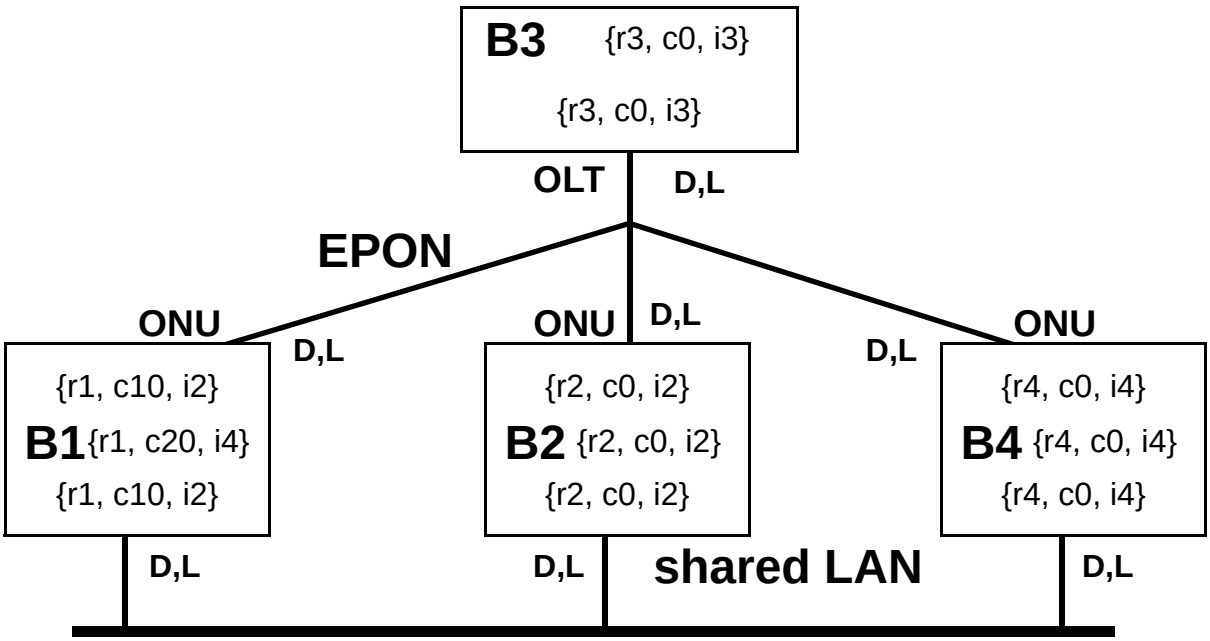

FIGURE 5. Point-to-Multipoint LANs: Initial state

Figure 6 shows the same example after the first BPDU exchange, before reconciling the information received on the various ports. Bridge B1 won the contest on the shared LAN. Bridge B3 received BPDUs from all three ONUs and bridge B1 won. Bridges B1, B2, and B4 heard only from B3. The OLT does not reflect B1's BPDUs; the EPON is *not* emulating a shared LAN.

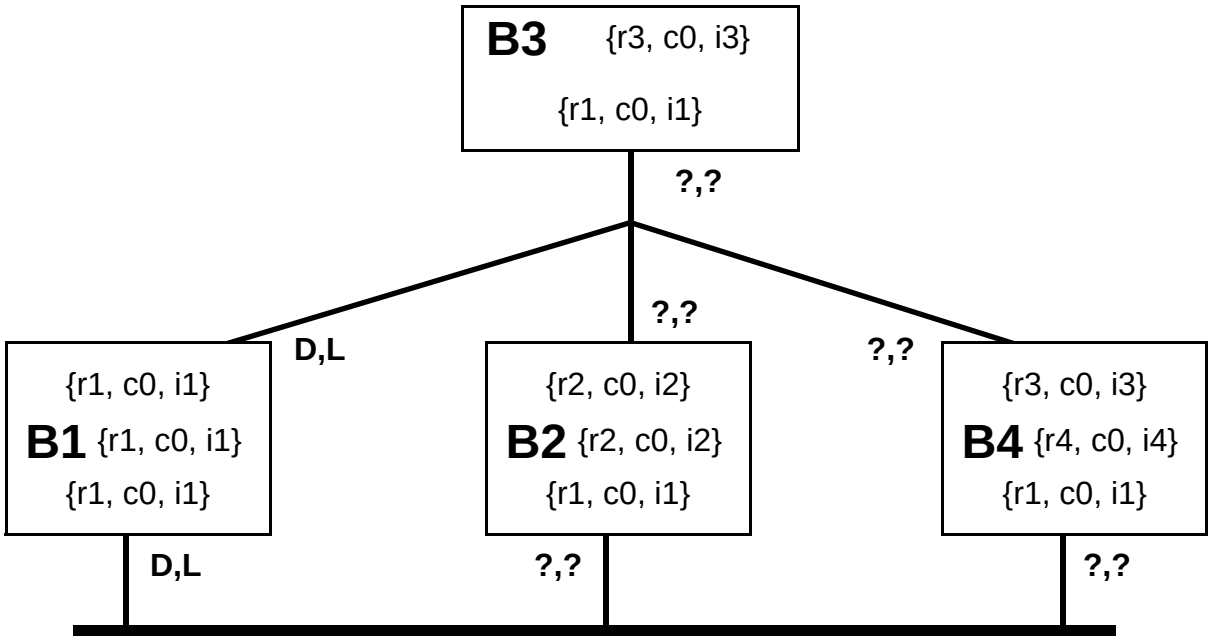

FIGURE 6. Point-to-Multipoint LANs: After BPDU exchange, before reconciliation

In Figure 7, we see the reconciled results of the first BPDU exchange, which is also the final state of the network. Bridges B2 and B3 have made their EPON ports Designated because the vector received directly from B1 was better than that in the first BPDU received from B3.

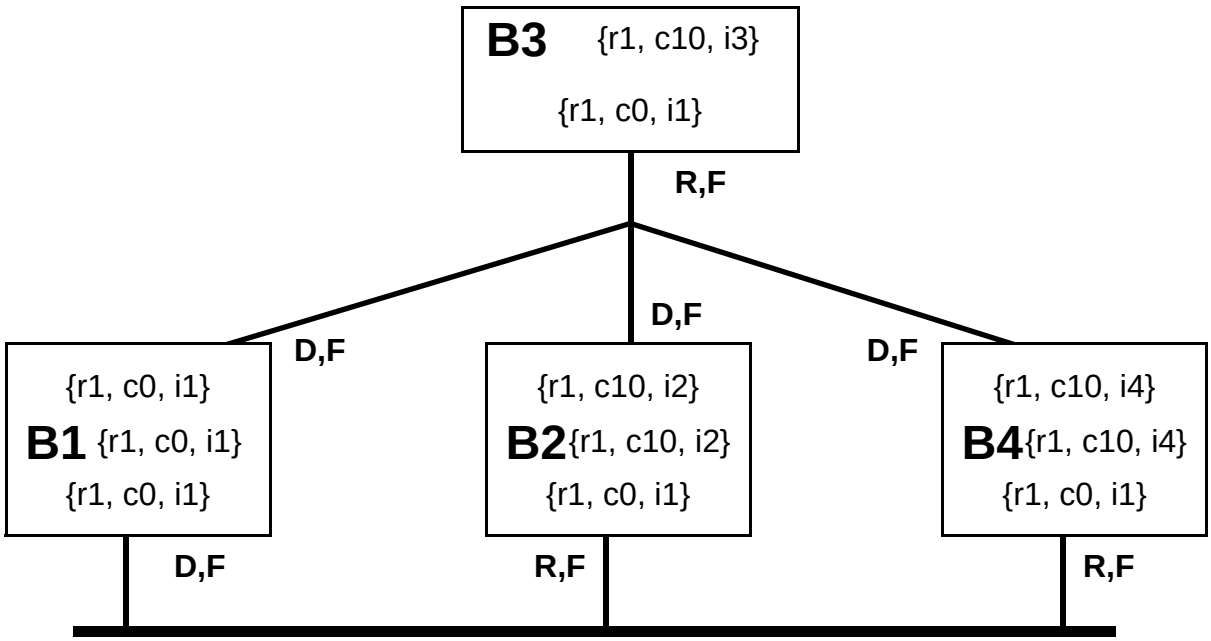

FIGURE 7. Point-to-Multipoint LANs: After BPDU exchange and reconciliation

In Figure 7, all ports are Forwarding, and none are Blocked. The reason is that B2 and B4 cannot see B1's BPDUs. B3 has stopped transmitting BPDUs, because its OLT port is not Designated.

Suppose we ask, “What would happen if a station behind B3 transmitted a broadcast frame?” The first part of the result is shown in Figure 8, and we can see already that this scenario is not going to end well. So far, Y has received one copy of the broadcast.

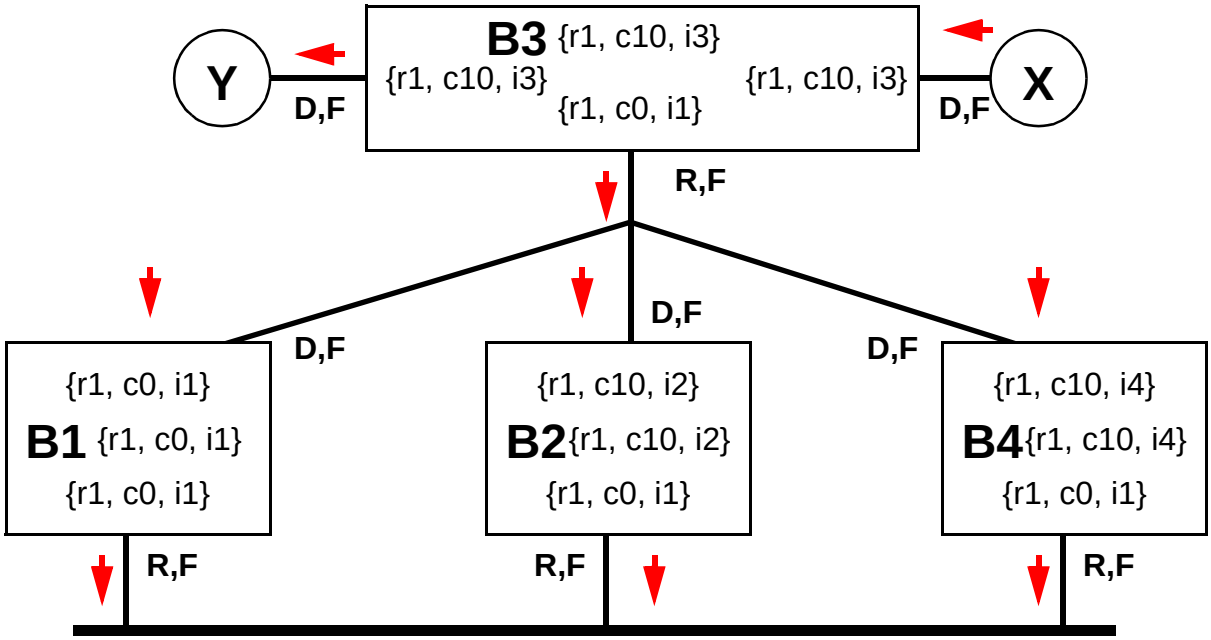

FIGURE 8. Point-to-Multipoint LANs: X sends a broadcast; First wave

In Figure 9, we see the next stage. Each bridge B1, B2, and B4 relays the broadcast to the shared LAN. Each therefore receives two copies of the frame, and relays them back up the EPON. Station Y winds up receiving seven copes of the frame, and X six. The reader will also note that, after sequence, all of the bridges have learned that Station X lives on the shared LAN! This is because the last frame *from* Station X was received from that direction.

The replication of this frame is a mini-storm. The replicated frames are not, themselves, replicated. This is because the asymmetrical characteristics of the EPON that create no blocked ports also prevent the endless circulation of frames. However, the reader will agree that delivering n*(n-1) extra frames, where n is the number of ONUs, is sub-optimal.

It is enlightening, perhaps, to see the final results if we vary the hardware of this scenario. In Figure 10, we replace the EPON with a set of point-to-point LANs. Here, we see that B2/B3 and B4/B3 select the Distributed Bridge of each LAN based on Bridge ID, because they are equidistant from the Root Bridge B1, and choose different directions for the settlement. This scenario works correctly; every LAN sees exactly one copy of the broadcast frame.

In Figure 11, we see the results with two shared LANs. Of interest in this diagram is an ambiguity in our simplified STP model. Bridges B2 and B4 select among their non-Designated ports using a method not yet mentioned; they use the configured Port IDs of B1’s ports to decide which shared LAN port to accept as the Root port, and which as the Alternate port. This scenario also works correctly; the broadcasts are discarded when they hit a Blocked bridge port.

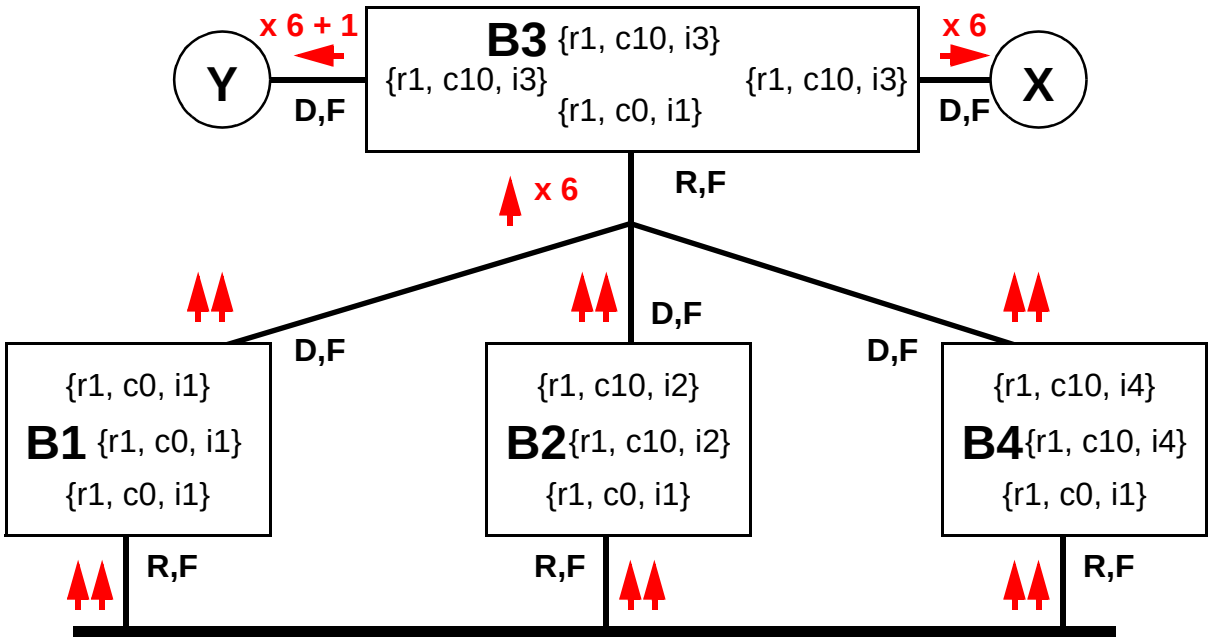

FIGURE 9. Point-to-Multipoint LANs: X sends a broadcast; Last wave

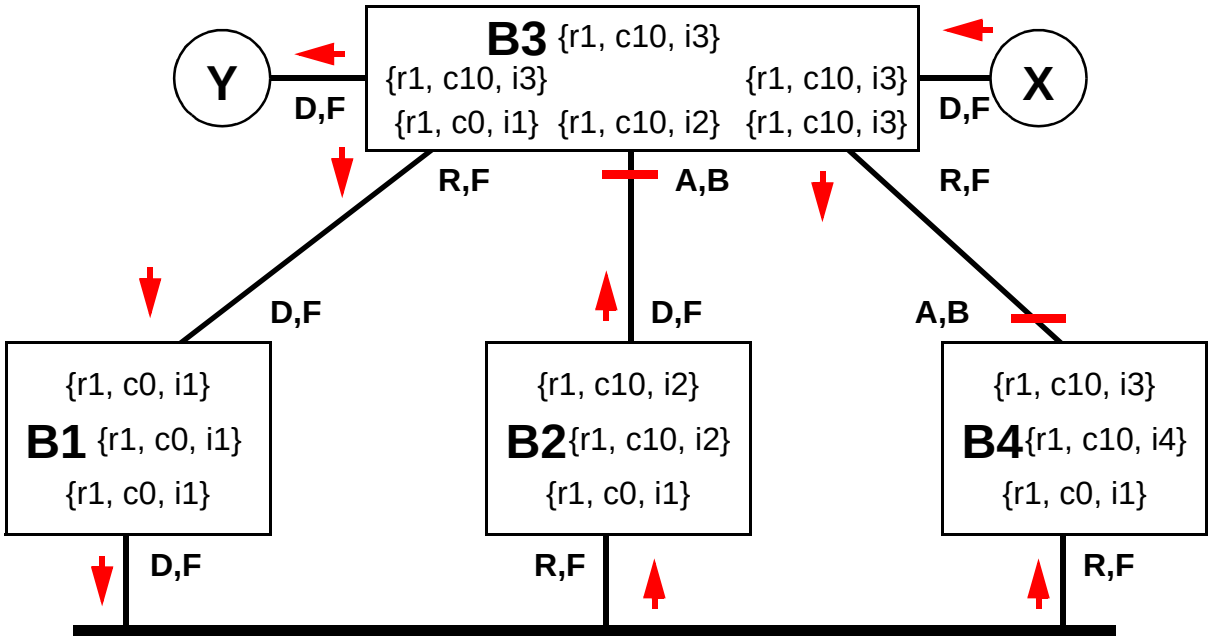

FIGURE 10. Point-to-Point and Shared LANs: X sends a broadcast

Figure 12 is slightly more interesting. We have disconnected B4 from the shared LAN and connected B1 and B3 via a point-to-point LAN, separate from the EPON. B2 and B4 remain connected to the EPON. This might be separate hardware, or this might be done by a choice of emulation modes. Of interest in this diagram is the fact that B4 believes that it is the Root Bridge on the EPON, even though it received a better claim from B3. This happens because, after B3

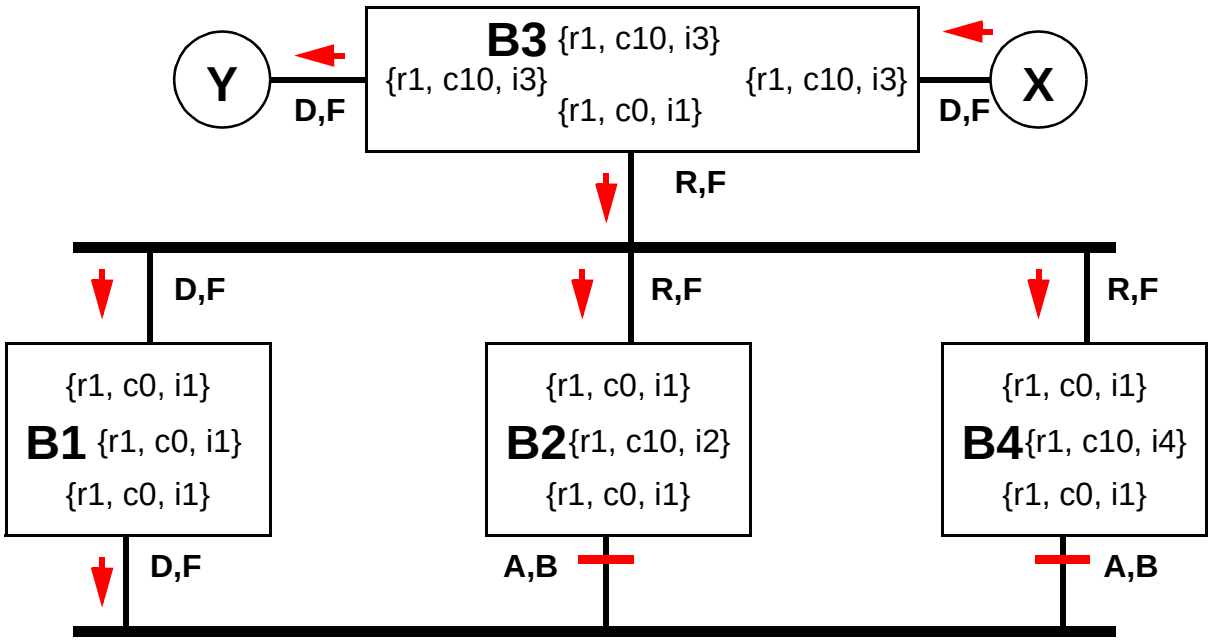

FIGURE 11. Multiple Shared LANs: X sends a broadcast

hears B2's BPDUs on the EPON, it makes the OLT an Alternate port, and stops transmitting BPDUs. Therefore, B2 and B4 hear no more BPDUs from B3. After B3's BPDU information times out in B4, it becomes the Root Bridge. As a result, B4 (and station Z) do not receive the broadcast at all!

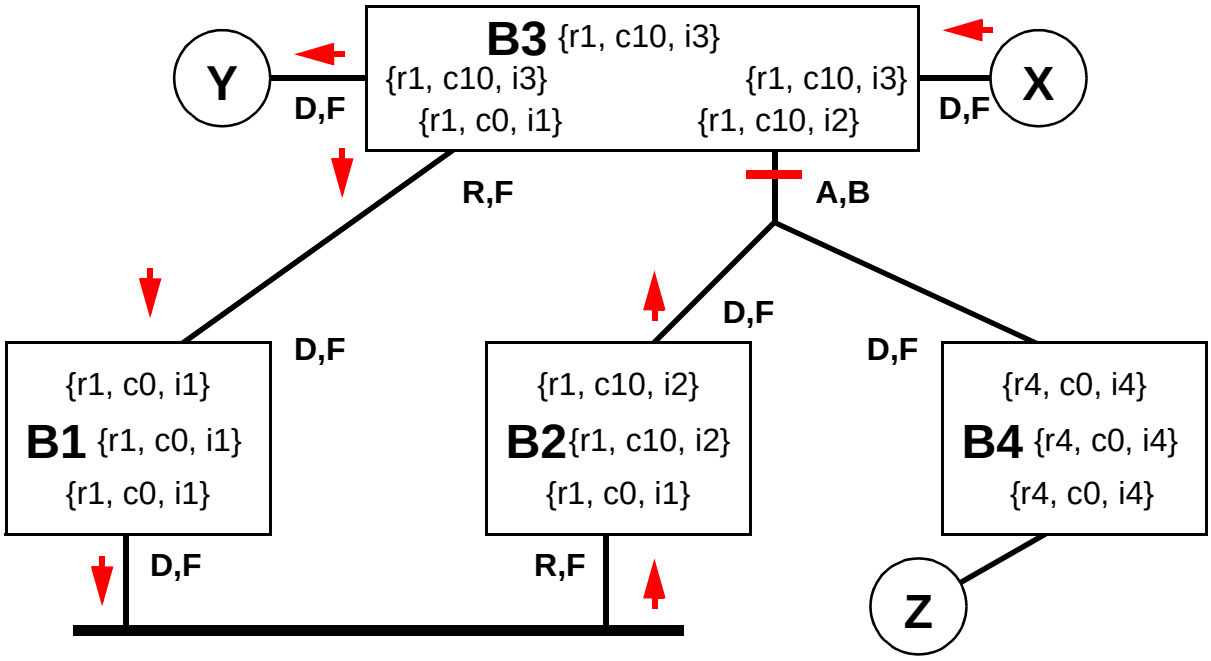

FIGURE 12. All LAN Types: X sends a broadcast

5.0 Conclusions

Clearly, EPON “native” mode, the point-to-multipoint LAN, does not work with the existing Spanning Tree Protocol. Using STP on a point-to-multipoint LAN can result either in delivering multiple copies of data frames (Figure 9, on page 10) or no data frames (Figure 12, on page 11) to portions of the bridged network.