Multiple Spanning Trees in 802.1Q

Norman Finn, Cisco Systems

1.0 Introduction

A number of means for adapting the spanning tree algorithm of 802.1D to the virtual LANs of 802.1Q have been discussed. At least three means have been proposed for employing one or more spanning trees in a single installation:

1. One spanning tree encompassing all physical links;
2. Any number of parallel spanning trees, up to the point of one spanning tree per VLAN; and
3. One super spanning tree for the "backbone" that connects all switches, with one or more sub-spanning trees for individual VLANs.

With a suitable choice of rules for the interaction between 802.1Q tags and 802.1D Bridge Protocol Data Units (BPDUs), the first two of these choices can be shown to be equivalent. No new protocols need be invented.

There are many ways of implementing multiple spanning trees in the context of 802.1Q. Section 2.0 gives the definition of "multiple spanning trees" used in this proposal. Section 3.0 lists the specific advantages that may be obtained if multiple spanning trees are allowed. Section 4.0 proposes a specific set of rules for the interaction between 802.1D and 802.1Q. Section 5.0 demonstrates how these rules allow a continuum of implementations and installations that range from one spanning tree per network to one spanning tree per VLAN. Section 6.0 discusses the costs of standardizing and implementing the rules of Section 4.0, and the work items for rev 2 of 802.1Q that these rules suggest.

2.0 Definition of Multiple Parallel Spanning Trees

The central idea driving this contribution is that the ability to operate multiple spanning trees in parallel offers advantages which outweigh a very small cost to the complexity of the standard.

An example of multiple parallel spanning trees is shown in Figure 1. In this diagram, three switches A, B, and C are connected via four physical links 1, 2, and 3. Two VLANs, "red" and "blue", are configured. The red VLAN is blocked at switch B's port to physical link 1, and the blue VLAN is blocked at switch C's port to physical link 3. For traffic on the red VLAN to get from switch A to switch B, it must first traverse switch C. Blue traffic, however, can move directly from A to B. Similarly, red traffic can flow directly between A and C, while blue traffic must traverse switch B.

If one spanning tree were used on the physical links between switches, 802.1D would block all traffic at one physical port to one switch. Traffic between any two switches would take the same path, regardless of which VLAN it is using. This is the essential difference between one and more than one spanning tree; whether the port blocked by 802.1D is a physical port to a physical link, or a logical port to a VLAN.

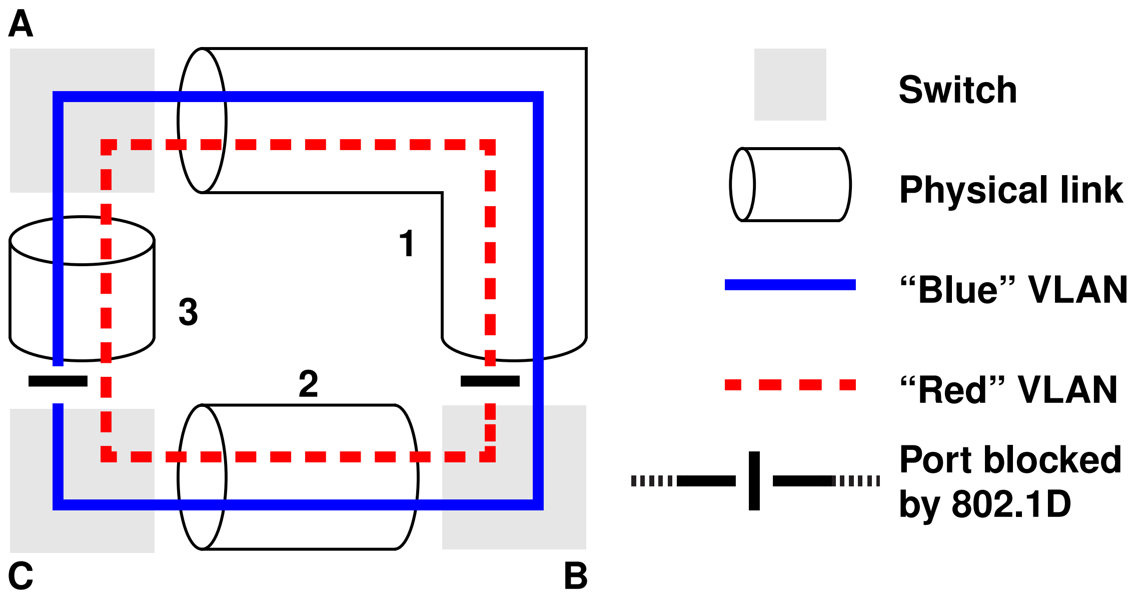

FIGURE 1. Example of Multiple Spanning Trees

3.0 Advantages of Multiple Spanning Trees

Before discussing how to make multiple spanning trees work, let us look at the reasons why they might be useful. The following examples are not intended to prove that multiple spanning trees *must* be used; a single spanning tree is certainly the right choice for many implementations and/or particular installations of 802.1Q. They do, however, provide very good reasons why multiple spanning trees should be *supported* by 802.1Q.

3.1 Simple Load Sharing

Figure 2 illustrates a simple case where multiple spanning trees provide relief from a common difficulty. Two switches A and B are connected via two physical links 1 and 2. Two VLANs, “red” and “blue”, are configured to be carried on both physical links.

If one single spanning tree is used, then one of the physical links must be unused for data traffic; it carries only 802.1D BPDUs. It serves as a hot standby for the other physical link. If two spanning trees are used, the spanning trees for the red and blue VLANs can be configured so that one VLAN uses one physical link, and the other VLAN uses the second physical link. If either physical link fails, spanning tree will cause the other link will carry both VLANs.

There certainly exist other means of solving this problem. For example, an inverse multiplexing protocol could be used to meld links 1 and 2 into a single logical link, and share the traffic load between the two physical links. Presumably, a load sharing algorithm based on some principle other than VLANs could be more equitable than the multiple spanning tree approach. However, there is no standard algorithm for this purpose, and none has been proposed for 802.1Q version 1. Should an inverse multiplexing protocol be adopted, it would serve the multiple spanning tree

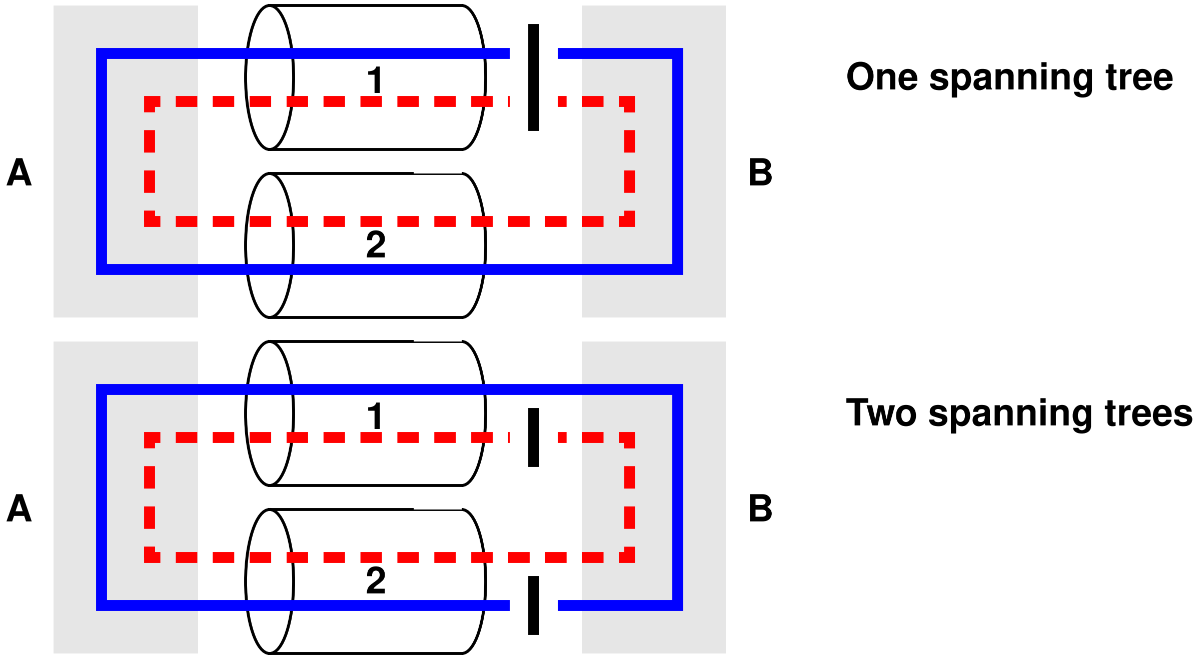

FIGURE 2. Simple Load Sharing

model as well as any other. In the meantime, multiple spanning trees can provide one solution to this problem.

3.2 Multi-path Load Sharing

Figure 3 illustrates a type of load-sharing problem that an inverse multiplexing protocol cannot solve. With three switches and links and three VLANs, each physical link can carry two of the three VLANs. While this a simplistic example, one can imagine the utility of this mechanism in more complex networks, especially in cases where many VLANs have a presence only in certain parts of the network.

3.3 Path Optimization

In Figure 4, we see two switches connected to each other, via link 1, and to other switches in a VLAN cloud. Connected to both switches is a single LAN segment, link 2, to which is attached file server F. Let us suppose that the VLAN to which this file server is connected is used primarily to connect that file server to numerous clients behind both of the two switches. With one spanning tree, the loop {switch A, link 1, switch B, link 2} must be broken by cutting one of the switches' ports to link 2, let us say, B's. Traffic from the file server to clients behind switch B must first flow through switch A and link 1 before reaching switch B. (The alternative, cutting the backbone between A and B, would usually be even worse.)

If link 2 is a 10 Mb Ethernet and link 1 a 1 Gb Ethernet, this is not a serious problem. If both are 100 Mbit Ethernets in computer room, the problem becomes more important.

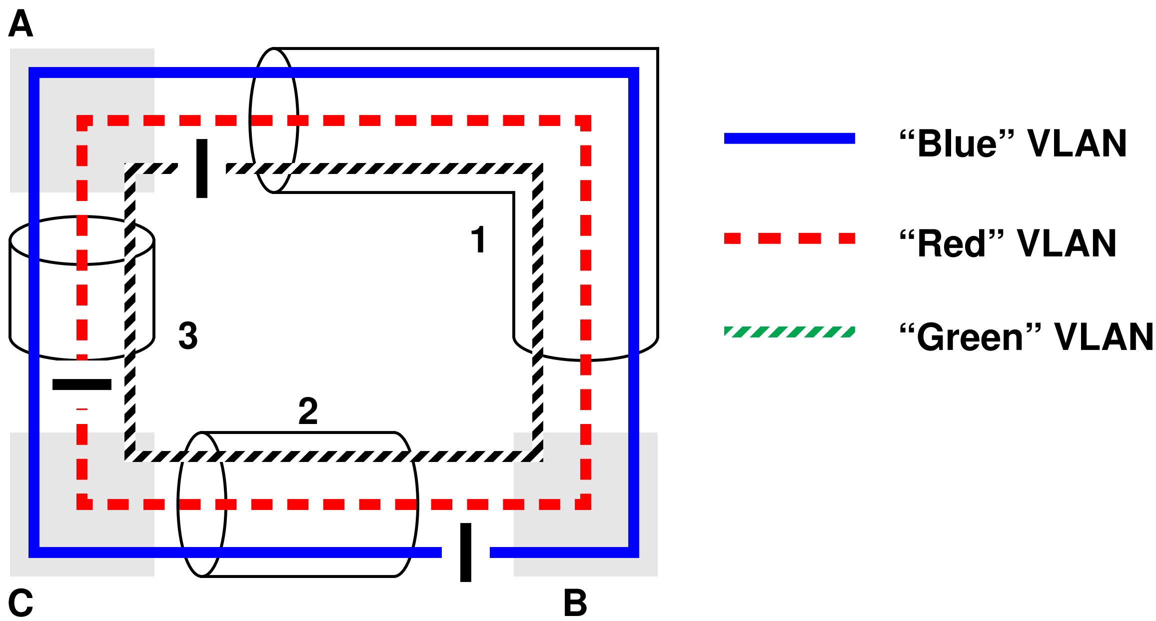

FIGURE 3. Multi-path Load Sharing

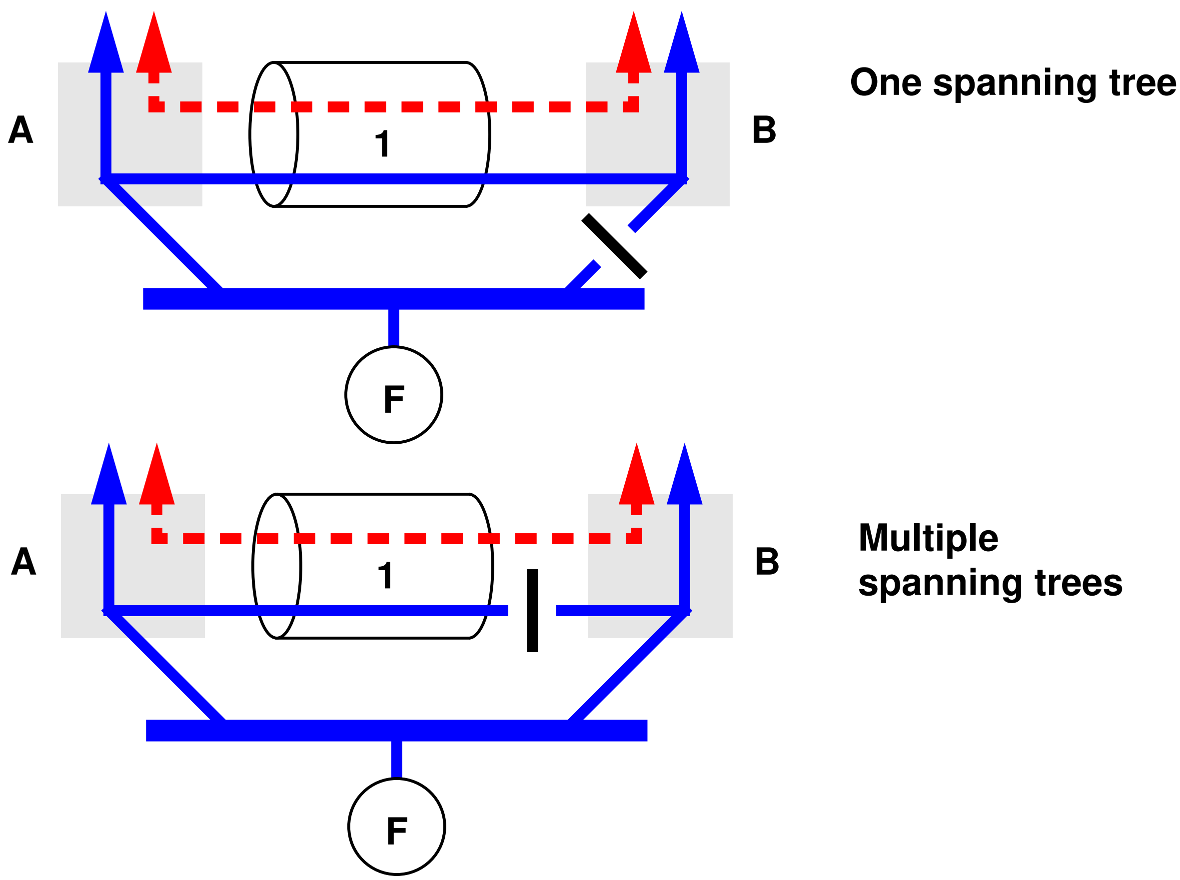

FIGURE 4. Path Optimization

3.4 Extending VLANs to Non-Trunk-Capable Switches

There are at least three reasons why an existing switch might be able to participate in VLANs with a software upgrade, but not be able to support an 802.1Q trunk:

1. The switch might require a hardware modification to its ports to support 802.1Q frame formats.
2. The switch might be incapable of supporting multiple VLANs on one physical port at all.
3. The switch might be incapable of supporting "baby giant" packets made longer than the physical link maximum by the length of the 802.1Q tag, in an installation where it is impractical to restrict the packet sizes sent by endstations.

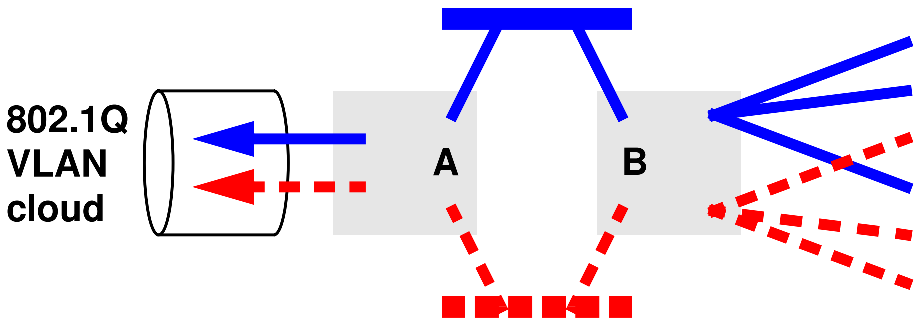

FIGURE 5. Extending VLANs to Non-Trunk-Capable Switches

In Figure 5, we see an example of a switch B connected by two simple 100 Mb Ethernet links to an 802.1Q-capable switch A that is part of a VLAN cloud. Switch B can, with a software upgrade, support a number of VLANs equal to the number of physical connections it has to switch A, but it cannot support an 802.1Q-tagged link to A. In this case, if a single spanning tree is used, then one of the two VLANs "red" or "blue" must be disconnected to break the loop {switch A, blue link, switch B, red link}. If the red and blue VLANs are on separate spanning trees, then both can operate and be part of the VLAN configuration.

3.5 Accidental Interruption of Backbone Connectivity

In Figure 6, we see two 802.1Q switches A and B both connected to a VLAN cloud. Suppose two untagged ports, one from each switch, are connected by an 802.1D bridge X that is ignorant of 802.1Q. One would hope that the loop in the blue VLAN would be broken where it presumably should be broken, at one of the switches' ports to that blue VLAN, or perhaps at one of bridge X's ports. However, depending on the various 802.1D adjustable parameters programmed into the switches and bridge X, the loop might be broken on the VLAN cloud side of the bridges.

In the two examples in Figure 6, we see the result of an unfortunate blockage. If the blue VLAN shares a spanning tree with all other VLANs, the backbone is broken, and all VLANs partitioned. If the blue VLAN is on a separate spanning tree, then the worst that can happen is that all blue traffic is funneled through bridge X. The path through the blue VLAN may be sub-optimal, but connectivity is maintained.

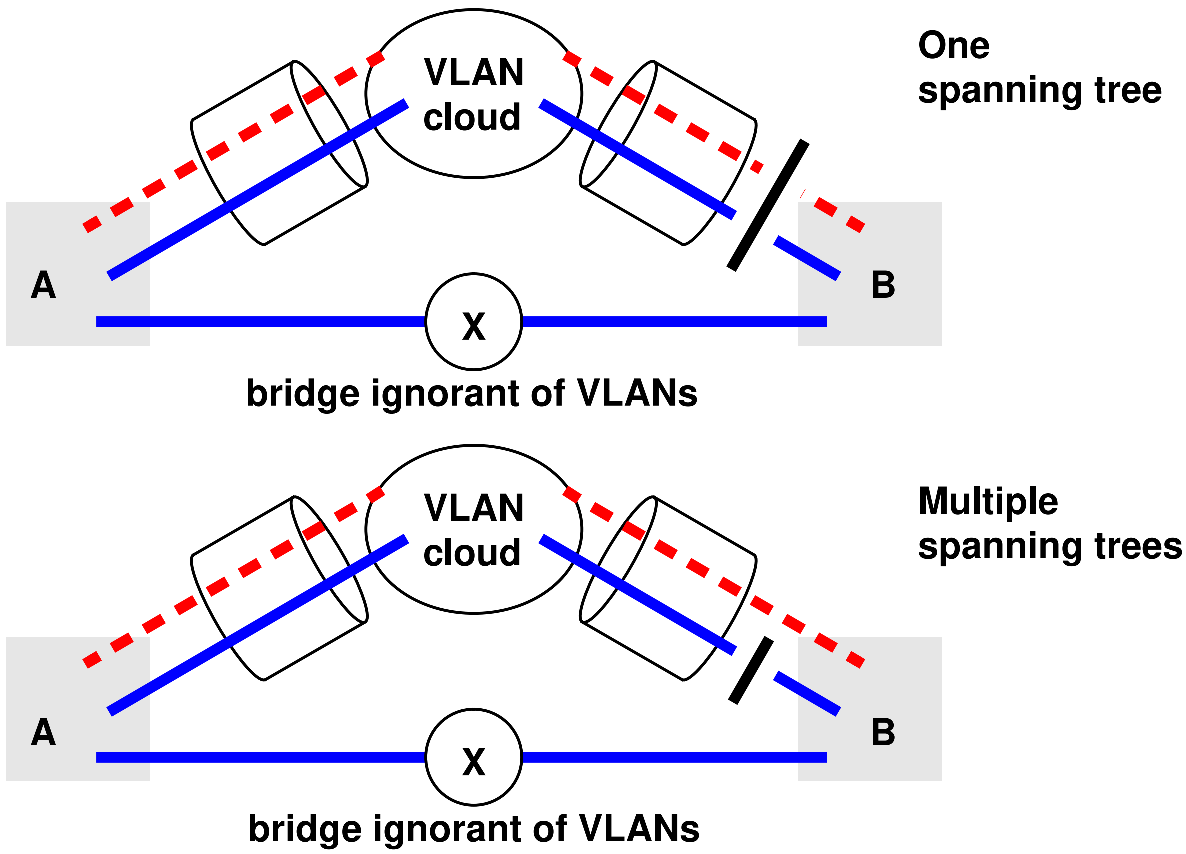

FIGURE 6. Accidental Interruption of Backbone Connectivity

3.6 Security versus Connectivity

For any number of reasons, it may be necessary to disallow a particular VLAN from flowing through a certain switch or across a certain physical link. Reasons might include:

1. Security: One might not want to pass the Human Resources net through Engineering.
2. Volatility: One might not want to pass production networks through a development laboratory, though connectivity to the laboratory is necessary.

As a trivial example, Figure 7 shows two switches, A and B, connected by two physical links 1 and 2. The red VLAN is restricted to physical link 1, and the blue to physical link 2. If one spanning tree is employed, then one of the two VLANs must be disconnected. If the two VLANs are in different spanning trees, both may remain connected.

3.7 Isolation from Spanning Tree Reconfiguration

Whenever the gain or loss of a link, port, or bridge causes a change in the configuration of a spanning tree, some disruption of service is possible. In the case of the loss of a component, some interruption is extremely likely. If all switches and all VLANs are running one spanning tree, any change to any part of the spanning tree affects all VLANs.

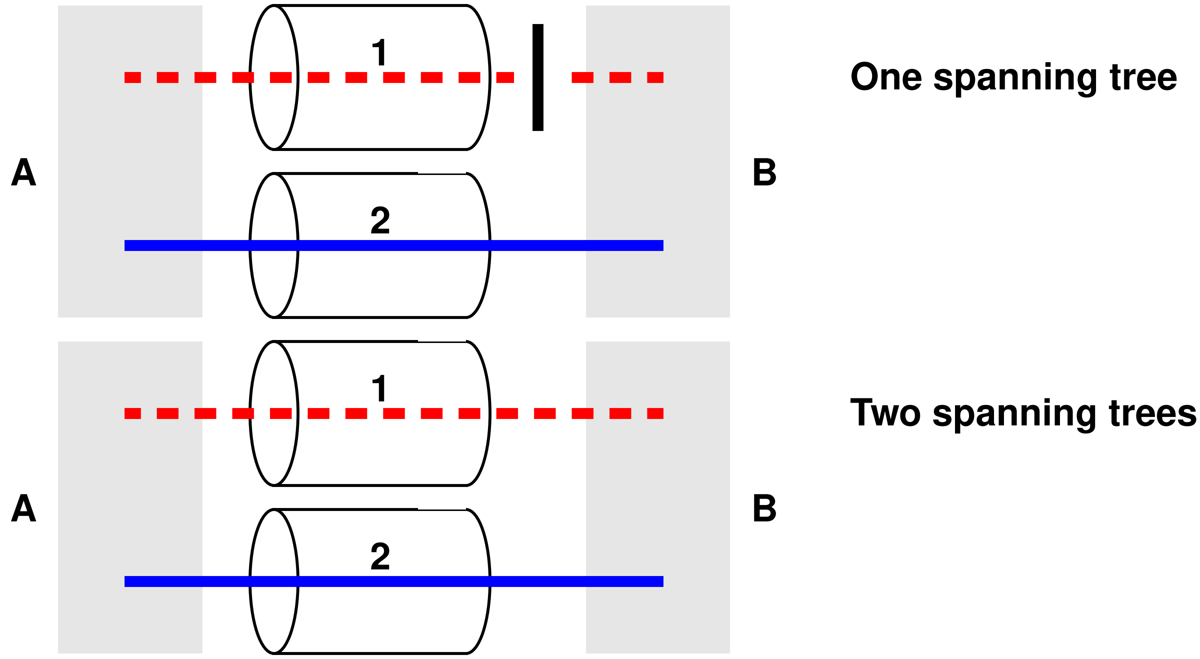

FIGURE 7. Security versus Connectivity

Multiple spanning trees improve this situation in two ways:

1. One may isolate certain VLANs and their spanning tree(s) to one part of the network. Changes in that spanning tree have no effect whatever on other parts of the network.
2. Having multiple spanning trees even in common areas of the network make it less likely that any given physical backbone link carries the VLANs of any given spanning tree, and thus less likely that a failure will affect any given VLAN. This is a corollary of Section 3.6, ”Security versus Connectivity”.

The utility of this separation is particularly useful if one considers the reasons for security listed in Section 3.6. Isolation of the production VLANs from the laboratory VLANs is clearly desirable.

4.0 Rules for interaction between 802.1D and 802.1Q

The specific rules required to coordinate 802.1D and 802.1Q in order to allow (but not require) multiple spanning trees are:

1. A Spanning Tree Group (STG) comprises one or more VLANs which share the same instance of the Spanning Tree Protocol (STP) of 802.1D.
2. A separate instance of the 802.1D STP runs in each switch for each STG enabled in that switch.
3. If a given physical switch port P is enabled for carrying traffic for any VLAN belonging to STG S, (and if STP is enabled for STG S on port P,) then a BPDU transmitted for STG S on port P may be transmitted exactly once on some one of the STG S VLANs enabled for port P. (The robustness of the model can be improved by constraining all BPDUs for STG S to be sent on one particular VLAN belonging to STG S on port P.)

4. If a given physical switch port P is enabled for carrying traffic for any VLAN belonging to STG S, (and if STP is enabled for STG S on port P,) then the switch must be able to receive BPDUs for STG S on any of the STG S VLANs enabled for port P.
5. A given physical switch port P has one 802.1D blocking state (blocked, learning, etc.) per STG (STP instance) enabled on that port. That state applies to all of the VLANs in the STG, but affects no VLANs in any other STG.

It is helpful to list some of the corollaries of these rules that can be easily derived:

6. A typical endstation port might carry exactly one VLAN, with no 802.1Q tagging. If STP is run on this port, it carries one BPDU per hello time from the STP instance associated with its STG.
7. If all of the VLANs on a “trunk port” carrying multiple VLANs are in the same STG, then that trunk carries only one BPDU per hello time.
8. If the VLANs on a “trunk port” belong to more than one STG, then there will be one BPDU passing through that port per STG per hello time.
9. A switch’s total BPDU load is thus not the number of STGs times the number of physical ports. It is, typically, the same as the single spanning tree case, increased by the number of extra STGs times the number of “trunk ports”.

5.0 Continuum Between ST per Network and ST per VLAN

Given the rules of Section 4.0, one may configure a network with one spanning tree or many spanning trees. If a network with 100 VLANs is configured with one STG for all 100 VLANs, it is employing the single spanning tree model. If it is configured with 100 STGs, one for each VLAN, then it is at the opposite pole. In-between configuration, with 10 VLANs in each of 10 STGs, or one STG with 90 VLANs and 5 STGs with 2 VLANs each, are equally possible. Any trade-off between flexibility and management load is possible.

We attempt to illustrate the compatibility of the rules in Section 4.0 with some common current assumptions about how BPDUs should be carried between 802.1Q switches:

5.1 Physical BPDUs and Logical VLANs

One common view of the relationship between 802.1D and 802.1Q is that BPDUs and the STP apply to the physical link. STP establishes a loop-free topology of physical links over which 802.1Q-tagged VLAN frames are carried.

To map this view to definitions and rules in Section 4.0, we simply establish one extra VLAN, an “STP VLAN”. There are no ports assigned exclusively to the STP VLAN. No data frames are carried on the STP VLAN. The STP VLAN is never tagged on any port. All other VLANs are grouped with the STP VLAN into one STG, the only STG configured.

On a “trunk port”, the “normal” VLANs are tagged with 802.1Q tags (or not, if implicit tagging is used). The BPDUs, and only BPDUs, are sent on the STP VLAN. (This is the reason for the last restriction in point 3. of Section 4.0.) Since this VLAN is not tagged, the BPDUs traverse the port in native mode. If STP blocks the port, all VLAN traffic except the BPDUs stops moving through the port.

On an “endstation port” carrying only one untagged VLAN V, if STP is enabled, the BPDUs are sent on VLAN V. Since VLAN V is untagged on this port, there is no difference between transmitting the BPDU on VLAN V or the STP VLAN.

5.2 Spanning Tree per VLAN

At the opposite extreme, if each VLAN is the only member of its own STG, every VLAN carries its own BPDUs. On any port (such as a “trunk port”) on which the VLAN is tagged, the BPDU is tagged. On any port where the VLAN has no tag, the BPDU is untagged. If two implicitly-tagged VLANs (with no 802.1Q tag or equivalent proprietary or semi-standard tag) share the same wire, and both are different STGs, then both BPDUs are transmitted and received on that port. (See Section 6.2, ”Consequences of Misconfiguration”.)

5.3 Intermediate Configurations

Each of the subheadings in Section 3.0 may be viewed as an advantage of using multiple spanning trees, or as a problem with using a single spanning tree. Between VLANs in separate STGs, all of the advantages are realized. Within the VLANs composing a single STG, all of the corresponding problems are realized. With configurations intermediate between the poles, a system administrator can balance needs against cost.

Note that this model allows interoperability between switches that assume the one spanning tree model and switches that support multiple spanning trees. VLANs which traverse switches supporting only one spanning tree must be bundled into the same STG. (This is not to suggest that such mixtures are a good idea; but to illustrate the flexibility of the model.)

6.0 Costs, Problems, and Rev 2 Work Items

The direct costs of this proposal are minimal. The multiple spanning tree model does generate some long-term desires for additional requirements on any forthcoming VLAN coordination protocol. Although the model does not generate any new problems, it does aggravate certain known problems of spanning trees.

6.1 Tagged BPDUs

Clearly, the tagging a BPDU with an 802.1Q tag must be allowed for this proposal to work.

6.2 Consequences of Misconfiguration

One may ask what happens if two switches have a different idea about which VLANs belong to which STG, or even how many STGs exist. Clearly, the perceived benefits of the multiple spanning tree model cannot be realized in the face of such misconfigurations. Fortunately, the robust nature of the 802.1D STP mitigates the severity of the problems generated by misconfiguration.

Differences in the configured relationships between VLANs and STGs result in the same kinds of errors as occur when two spanning trees are linked with a wire; the two spanning trees are merged. Similarly, when VLAN/STG relationships differ among switches, one spanning tree is formed over the union of all VLANs linked together into the same STG by any switch. In fairness, this concatenation is not without risk: the resultant combined spanning tree could exceed the maximum STP diameter.

6.3 Non-VLAN-Aware Switches

If non-VLAN-aware switches are mixed with VLAN-aware switches on untagged links, multiple parallel spanning trees introduce no new problems. If a non-VLAN-aware switch is placed on a trunk carrying tagged BPDUs, there will be a problem. The BPDUs will be stopped, because their destination MAC addresses will be recognized by the non-VLAN-aware switch. They will not, however, be interpreted properly. The data packets, however, will be passed by the switch.

This problem can be avoided by using the technique of Section 5.1, ”Physical BPDUs and Logical VLANs” in sections of the network in which non-VLAN-aware switches are used. If tagged BPDUs are accidentally directed (by mis-configuration or mis-wiring) to a non-VLAN-aware switch, then the VLAN-aware switches will be made aware of the problem when they receive the untagged BPDUs.

6.4 Ensuring Consistent VLAN/STG Relationships

There are better ways to ensure that each switch has the same idea about which VLANs belong together in which spanning tree group than to trust that each switch is configured the same way. Two of them are:

1. **Arbitrary association.** Allow an arbitrary association of VLAN with STG. That is, the definition of an STG is a list of VLAN-IDs that may contain anything from one to all of the VLANs, in any combination.

   To ensure consistency, this approach requires that the VLAN/STG associations be distributed among the switches. The “distribution protocol(s)” assumed by 802.1Q/D2 should be sufficient to distribute this information.

2. **Subdivide the VLAN-ID field.** We can break the VLAN-ID into two pieces, one of which identifies the STG, and one the VLAN within the STG. For example, a 12-bit VLAN-ID field could be split into a 4-bit STG-ID and an 8-bit sub-VLAN-ID within the STG. (Presumably, two VLANs with the same sub-VLAN-ID in two different STGs would be considered as two different VLANs.)

Some ports, especially “trunk ports”, may carry more than one STG, and said STGs may be in different states on that port. Both of these approaches therefore require that a switch have the ability to use the VLAN/STG association information to condition the treatment of an incoming packet by its VLAN-ID, and not merely by the bridge port on which it was received.

The subdivided VLAN-ID field may well be easier to implement in hardware than the arbitrary association. It reduces the size of the VLAN-ID-to-STP-state translation table to the number of STGs accommodated by the division (16 in the above example) instead of the number of VLAN supported (likely either 4k or 32k).

The method of subdividing the VLAN-ID is relatively inflexible. (If the dividing line between the two parts of the VLAN-ID can vary, then some means of ensuring that all switches agree on the division is required, and much of the attraction of a sub-divided VLAN-ID is lost.) It may be possible to find a division of the VLAN-ID that satisfies enough requirements to set it in the standard.

Clearly, this use of VLAN-IDs cannot be discussed in isolation from the question of the scope of VLAN-IDs. Whether VLAN-IDs are global to a network, local to an administrative domain within a network, local to a LAN segment, or local to a set of ports on a LAN segment, must be related to other definitions of VLAN-ID formats.

7.0 Summary

It is not the primary purpose of this proposal to claim that the multiple spanning tree model is invariably "better" than the single spanning tree model. The management load incurred by a configuration with 4000 VLANs running on 4000 separate spanning trees, for example, would be difficult to justify. The essence of this proposal is that:

1. by defining the appropriate relationship between 802.1D and 802.1Q, any configuration on the continuum from one spanning tree per network to one spanning tree per VLAN is possible, and
2. that this flexibility incurs no significant cost to those who choose tailor their implementations to one or the other pole of the continuum.

8.0 Acknowledgments

Many thanks to Paul Frantz, Richard Hausman, Keith McCloghrie, and Luc Pariseau for their comments and suggestions made in response to earlier drafts of this contribution.