High Availability Spanning Tree

Mick Seaman

The reconfiguration time of IEEE Std. 802.1D Spanning Tree Protocol can be radically improved by using port roles ("Root Port", "Designated Port", "Alternate", or "Backup") to select forwarding state transitions. At present these transitions are solely determined by the current and desired forwarding states (Blocking or Forwarding).

A newly selected "Root Port" can be moved directly from the Blocking to the Forwarding state provided that the previously selected Root Port is made Blocking. Loop-free topology is maintained.

The proposed transitions accomodate arbitrary reconfiguration during periods of network device and link changes. However, an important application is autoconfiguration of redundant connections as backup "resilient links". The 'next best' root port can be precomputed and, if the physical media can signal link failure (through "link beat" or "loss of light" indication), physical connectivity can be restored within as little as 10 milliseconds.

Such immediate recovery is in dramatic contrast to the slow (about 50 seconds) standard spanning tree reconfiguration times. It provides a basis for highly available continuous network operation based on redundant low cost network devices, with recovery times commensurate with fault tolerant telecommunications equipment. Using the campus data network for voice applications with the proposed improvement, failure and recovery may result in the loss of only a few speech samples - a 'click' on the line rather than loss of a call.

Although a redundant backup might be manually configured, this proposal minimizes administration, and is robust in the face of multiple failures spread over time.

Although High Availability Spanning Tree changes the dynamic effects of the spanning tree protocol (bridge ports becoming forwarding or blocking at certain times), there is no change in the BPDUs sent. Nor is it necessary for all bridges in the network to change their behavior. Those that do benefit from the much reduced reconfiguration delay, and can be introduced arbitrarily into an existing network.

Introduction

The IEEE Std. 802.1 D Spanning Tree Protocol automatically establishes fully connected ("spanning") and loop-free ("tree") bridged network topology. It uses a distributed algorithm that selects a "root" bridge and the shortest path to that root from each LAN. Tiebreakers are used to ensure that there is a unique shortest path to the root, while uniqueness of the root is guaranteed by using one of its MAC addresses as part of a priority identifier.

Every LAN in the network has one and only one "Designated Port" providing that LAN's shortest path to the Root, through the bridge of which the port is part. The bridge is known as the "Designated Bridge" for the LAN.

This note introduces the term "branch bridge" or "Branch" to describe bridges in the network other than the "root bridge" or "Root". Every branch bridge has a "Root Port" which is the port providing that bridge's shortest path to the Root. Ports other than the Root Port are Designated Ports, Alternate Ports, or Backup Ports. An "Alternate Port"[1] is connected to a LAN for which another bridge is the "Designated Bridge". A "Backup Port" is connected to a LAN for which another port on the same Bridge is the "Designated Port".

The connectivity through any bridge is thus between its Root Port and Designated Ports. When Spanning Tree information has been completely distributed and is stable, this connectivity will connect all the LANs (i.e. it is "spanning") and will be loop free (i.e. it is "tree").

When a bridge first receives spanning tree information that dictates new connectivity through the bridge, it does not establish the new

[1] The term "Alternate Port" has just been added to the latest revision of 802.1D. The use of the term "Backup Port" with the precise meaning given here is new.

connectivity immediately. Ports that were connected previously, but are no longer, are immediately made blocking, but the transition to forwarding of ports that were previously not connected is delayed. This delay serves two purposes:

a) frames forwarded on the previous topology may still be buffered by bridges in the network, so an instantaneous change to the new topology can cause these to be forwarded back to their LAN of origin, thus duplicating the frame once.

b) new spanning tree information in the network may not have been fully distributed yet, so an immediate change to the new topology may cause temporary loops. These loops could generate high traffic volumes, disrupting end stations, causing frame loss in bridges, and possibly delaying the propagation of spanning tree information further.

The first of these two reasons, (a), is far less important than it once was. The protocols prevalent on LANs today can deal with immediately duplicated frames. Some old implementations of LLC Type 2 will reset the connection under these circumstances, but they are no longer in widespread deployment - at least not on Ethernet. The remainder of this note ignores the possibility of single duplicates.

The second reason, (b), continues to be fundamental to spanning tree reconfiguration and is fully addressed by this note.

The analysis and port state transitions to forwarding specified in 802.1D treat each bridge port equally, whether it is a Root Port or a Designated Port. Once the spanning tree has done its job of identifying the ports that should be forwarding eventually, loop-free-ness in transition is achieved without reference to the spanning tree.

This note shows that an Alternate Port that becomes the new Root Port can forward frame immediately, provided that the prior Root Port becomes blocking, at least for a period. The new port state transition rules introduced thus cover the case of physical link loss of the Root Port, allowing immediate failover to a redundant link.

The benefit of this high availability improvement to the spanning tree is most dramatic when point to point links with “loss of light” or “link beat absent” capability interconnect the bridges. Physical failures can be detected and repaired within as little as 10 milliseconds. However the analysis is equally applicable to non-point-to-point, i.e. shared media connections

The faster reconfiguration benefits described are also not limited to failure of a Root Port. New spanning tree information can be acted upon quicker. While the new analysis based on Root Port movement would indicate that the prior Root Port should be made blocking, even if it is to be a Designated Port, the traditional and new analyses can be combined to show that if the new Root Port is already forwarding then the state of the old Root Port can be left untouched if it is to become a Designated Port[2].

In addition to automatically identifying and allowing fast failover “resilient” links, the transition tables described here incorporate a prior improvement to the spanning tree: the addition of a “Forgetting” state for once Designated Ports that have been put into Backup. This state minimizes the potential denial of service resulting from information races during extensive reconfigurations.

References

[1] 802.1D

[2] A more Robust Tree: Active Topology Maintenance in Reconfiguring Bridged Local Area Networks (STP+), Mick Seaman

Port Roles, States, and Behavior

802.1D [1] identifies, if not explicitly, four distinct port “roles”:

1. Root Port
2. Designated Port
3. Alternate Port.
4. Disabled Port

This note adds:

5. Backup Port

to explicitly identify one of two cases previously covered by “Alternate”.

These names all identify the various roles the ports play in the topology of the spanning tree (or in the case of the Disabled Port, the fact that it plays no role at all). A Root Bridge has Designated Port(s) and may have Backup Port(s), where these connect to the same shared media LANs as its Designated Port(s). A Branch Bridge has one Root Port, and may have Designated Ports and/or Alternate and Backup Ports.

802.1D also identifies five Port States:

1. Disabled
2. Blocking
3. Listening
4. Learning
5. Forwarding.

This note adds:

6. Forwards
7. Forwarder
8. Forgetting

These states define the behavior of the port:

a) forwarding frames or discarding frames

[2] However if this optimization is included any and all ports that have been the Root Port ‘recently’ have to be made blocking once the Root Port is moved to a port that was not previously forwarding, i.e. not just the immediately prior Root Port.

b) learning source addresses or not

and provide for transitions between the Port Roles.

The Standard Algorithm

Tables 1A and 1B describe the standard Spanning Tree Algorithm, unchanged since 1986. Table 1A shows the Port Roles, and the possible Port States.

As Table 1B illustrates[3] the rules for frame forwarding and source address learning can be understood fully by reference to the Port States alone, as can the changes of Port State caused by the events shown. In fact the port state transitions of [1] are unchanged by combining "become root port" and "become designated" into a single event "make forwarding", and this is how [1] is described[4].

The High Availability Algorithm

Table 2 shows port roles, states and transitions for 'high availability'. A new Root Port becomes Forwarding immediately[5]. The prior Root Port is transitioned to Listening if it becomes Designated[6], and to Blocking otherwise.

Table 3 presents the same information in a slightly different way. Before a new port can become Root Port, a 'retire root port' event is sent to all other ports. On receiving this event the existing Root Port becomes an Alternate Port, and Blocking. Subsequently, as part of the same spanning tree recomputation, the port may be selected as a Designated Port and Become Listening[7].

Table 4 enhances the transitions so that if the new Root Port was already Forwarding, the prior Root Port does not have to be 'retired' and can continue Forwarding. However if the new Root Port is not already Forwarding, all ports that had previously benefited from the graceful transition now have to block for a period. Two[8] new states 'Forwards' and 'Forwarder'[9] are introduced to track such ports.

High Availability with shared media

Tables 2 thru 4 do not accommodate the difficult case of a bridge with two ports attached to the same LAN where one is Designated and the other is a Backup. In this case new spanning tree information could cause the Backup port to become the new Root Port before the Designated Port received the same information[10]. A temporary loop could be created.

Table 5 extends the high availability algorithm to deal with this issue[11].

When a Backup port becomes the Root Port, the event 'retire primary port' is sent to the Designated port which had forced it into Backup. This forces the Designated port to begin again in the Listening state[12].

Including glitch suppression

Table 6 includes the "Forgetting State" improvement described in [2].

[3] Table 1B corresponds to Fig 4-3 in the original 802.1D with the exception that a port that is taken out of Disabled becomes Designated, which is what is specified in (at least part of) the C code description in 802.1D though not in Fig 4-3. In this case the code makes sense, and an 'error report' ought to be filed against .1D for inconsistency.

[4] "Become alternate" is described as "make blocking".

[5] Even if the previous root port is still playing an active role in the topology, i.e. the new transitions are general and are not merely limited to failure cases.

[6] Even though the port is already Forwarding.

[7] The effects of address learning are not discussed here. There is no requirement to remove all addresses from the forwarding database instantly, so an instantaneous transition from Forwarding to Listening through Blocking need not have any greater effect on learnt addresses than the direct transition from Forwarding to Listening.

[8] One state could have been used, but using two allows the Forward Delay Timer to be used for this purpose.

[9] The name 'Forwards' is Forward plus the 's' from Listening, the first Forward Delay Timer driven state. 'Forwarder' is Forward plus the 'r' from "Learning', the second timer driven state.

[10] Because of variable processing delays due to receive queues on both of the ports, or because the message was accidentally lost by the designated port – the spanning tree algorithm is meant to cope with the occasional lost message without causing temporary loops.

[11] While many campus LANs are fully switched, at least in the core, there is always a chance that a shared media device might be introduced. Hence it is vital to have an answer which encompasses shared media in order to meet plug and play goals, and allow immediate piece-meal deployment. While it would be possible to detect shared media and turn the enhancement off, it is as easy to refine the algorithm. The inclusion of extra states to cover this potentially obscure or infrequent case should have little or no effect on ordinary execution.

[12] A first choice might be to force the Designated port to become Alternate. However since it has not yet received the protocol information necessary to put it in the Alternate state that is inappropriate.

Why it Works

Imagine a tree, the bushy kind, preferably a conifer whose branches point upward. Take your secateurs and cut off a branch. The 'root port' of the branch is where the cut has been made. Where can you attach the branch back to the rest of the tree without creating a loop? Anywhere! How long do you have to wait? No time at all!

But what if you were to accidentally attach the bottom of the branch back to the branch itself?[13] How can we assure ourselves that our branch will not bend back on itself, or that its separate twigs will not touch and weld together without becoming separated further down?

A branch bending so that one part of the branch now touches a part lower down will not result in a loop unless the contact point immediately become forwarding. But according to the rules and state machine above, this instant transition can only occur if the bridge port which transitions to forwarding becomes the root port of the bridge to which it belongs. This will in turn cause the previous root port of that bridge to block. In effect the branch snaps rather than bending. The tree is still simply connected.

What about our last case, where two twigs in the tree are suddenly welded together? Before the weld at least one of the twigs will be blocking the connection between the two of them. The only way to cause the weld to happen instantaneously is to make the twigs the root port of the rest of the tree - effectively to seize the tree, turn it upside down and plunge it into the earth once more. However just as this causes the twigs to weld together it will cause the bridge that owns the previously blocking port to block its previous root port. That twig has then snapped off the rest of the tree.

The only difficulty is where the same bridge has two ports connected to the same shared media LAN. In this case there is no way of snapping off a twig. If a Backup Port becomes the Root Port, the state of the Designated Port that was making it Backup is forced to Listening. This ensures that a loop is not created before the two ports have received identical information.

A more formal explanation

A bridged LAN is best represented as a bipartite graph, i.e. a graph comprising two types of node where each node is connected only to nodes of the opposite type. In this case the nodes are bridges and LANs, and the arcs of the graph that connect them are bridge ports. The Spanning Tree Algorithm works by removing some of the arcs, i.e. blocking some of the bridge ports, so that the graph is both fully ("spanning") and simply ("tree") connected.

An important result of the operation of the algorithm is that every bridge has a unique ranking in the network, i.e. there is a "metric" against which each bridge can be measured, such that no two bridges have the same value. This metric is:

the Bridge Identifier of the (assumed) Root Bridge

+

the Root Path Cost for the bridge

+

the Bridge Identifier

where the total contribution of the root path cost is always less than one indivisible unit of Bridge Identifier for the Root Bridge, and similarly the final bridge identifier component is always less than one indivisible root path cost unit. The lower the value of this metric, the higher the priority of the bridge as compared to other bridges.

Every LAN also has a value for this metric. The value for each LAN is equal to that of the most important bridge connected to that LAN (the Designated Bridge). If that bridge is the Root Bridge, the Root Path Cost component is zero.

Each bridge, other than the Root bridge, selects one of its ports to be its Root Port, choosing the port for which the sum of the metric for the LAN to which it attaches and its own Port Cost is lowest.

The Spanning Tree Algorithm blocks ports (removes arcs) which are neither Root Ports or Designated Ports (the unique port that connects a Designated Bridge to a LAN for which it is designated).

The important bridge ranking result described above can be illustrated by diagrams like the following (Figure 1).

[13] The Spanning Tree Protocol can not prevent a temporary loop if two previously disconnected LANs were suddenly to be connected by something other than a bridge – a switchable repeater for example. This possibility, which is equivalent to swiftly and invisibly moving one end of a physical connection, is not considered here. If it does occur STP will detect and correct the loop.

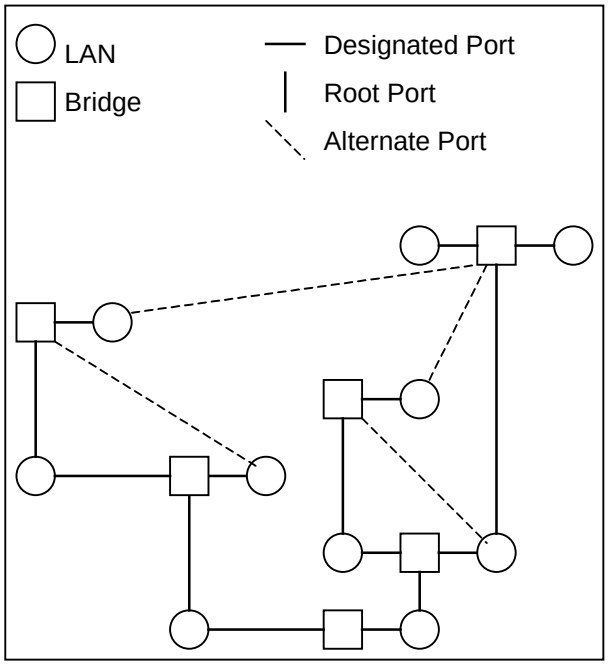


Figure 1 – Spanning Tree Metric

This figure uses height to depict the ranking metric (root identifier plus cost from the root plus bridge identifier). Absolute height is not relevant, what is important are the relative heights of bridges, that no two are at the same height, and that the Root bridge[14] is at the bottom of the diagram.

Every LAN for which a bridge is Designated Bridge is at the same height as that Bridge[15], which makes the arcs that represent the non-blocking Designated Ports horizontal. Since the horizontal dimension is unimportant it has proved convenient to show Root Ports as vertical.

Alternate Ports, which the spanning tree algorithm blocks, always connect a LAN that is lower in the figure to a higher bridge. If a port was attached from a lower bridge to a higher LAN, that LAN would be 'pulled down the figure' to the level of the lower bridge, which would then become the Designated Bridge for the LAN.

To state this another way, the current path from a bridge to a LAN with a lower (more important) metric[16] always passes through the Root Port of that bridge, while ports connected to other LANs will already be Designated[17].

Hence, if the Root Port of any of the bridges in the figure is removed, any Alternate Port on the bridge can be made the Root Port, and become non-blocking immediately, without causing a loop in the active topology - because we know that that bridge and any bridges in the subtree above it were simply attached to the rest of the tree through that previous Root Port.

In fact any port on the bridge can be chosen as the new Root Port[18], but it makes no sense to select one that is currently Designated, because it is known that that port cannot possibly attach to the rest of the tree from which the bridge is now partitioned[19]. Moreover the best Alternate Port[20] should be chosen since the spanning tree configuration rules will eventually demand that it be selected.

The principle that any port at all could have been chosen without risking an immediate loop is however very useful, since it removes the need to test that a proposed new Root Port actually leads to the same Root as the previously selected Root Port[21]. It also allows rapid reconfiguration to deal with the case of the Root bridge actually failing and being replaced, not just the case of a link from the Root failing.

Now let's use some of these diagrams to examine the potentially worrying scenarios discussed informally above. What we are looking for is ways in which moving the Root around can cause a loop through new Root Ports becoming forwarding immediately. We suspect that we can do this by making changes faster than the spanning tree can reorganize, i.e. after we make our first change, the question of the relative 'height" of bridges in the diagram can be ignored because bridges have not propagated consistent information to each other. All that can be relied upon is that a single bridge is internally consistent : it thinks there is one Root in the network at any point in time and has a single Root Port, or no Root Port at all if it is the Root itself.

[14] And the LANs to which it is directly attached.

[15] This two dimensional diagram cannot easily show a bridge as having more than two Designated Ports. A three dimensional version where the vertical continues to depict the ranking metric and bridges and LANs can be freely arranged in the horizontal plane fixes this problem. Unfortunately realistic three-dimensional drawing is beyond my capabilities at present.

[16] Strictly speaking a "lower value of the metric" or a "higher rank".

[17] With the exception of Backup Ports.

[18] Not quite true, a Backup Port must be avoided. Limiting the choice to Alternate Ports for fast failover avoids this problem.

[19] A Designated Port may or may not provide physical connectivity to the rest of the tree in the last resort. If it does there will be another port on another bridge which is currently blocking to prevent a loop. If there are no Alternate Ports the right thing for a bridge to do is to attempt to become the Root itself. This is what the unmodified spanning tree algorithm would do in any case. If there is another bridge with a blocking port this is as good a way to restore full connectivity in the network as any other. If that bridge is using other improvements to the Spanning Tree (described elsewhere) having the first bridge become Root will hasten the reconfiguration and restoration of service.

[20] The spanning tree algorithm will naturally select this as the new Root Port as it now represent the best path to the Root.

[21] Introducing history into an algorithm such as this is always difficult. Since bridge or link failure can lead to multiple bridges claiming to be the new Root is quick succession there is always a question of when to "freeze" on the "current" Root. The whole mess of introducing extra rules to judge when a stable state has been reached is best avoided.

The following diagram (figure 2) is a useful starting point.

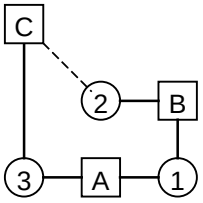


Figure 2 – A Test Network

Let's manage the priority of bridge B so it becomes a better Root than A. On becoming root, B will send out config BPDUs through both of its ports. Bridge C will receive one of these on its port 2[22], and will make that Port its new Root Port and start forwarding on that Port. However, since the new Root Port was not already forwarding it will block on the previous Root Port. So very shortly after B has been elevated to Root the connectivity will be as shown in Figure 3.

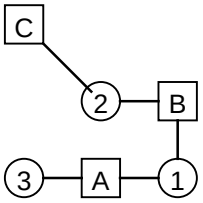


Figure 3 -Test Network Reconfiguration

And eventually the network will settle down as shown in Figure 4.

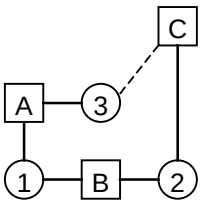


Figure 4 – Reconfiguration Complete

A similar effect could have been achieved by attaching a new Root bridge to LAN 2 in Figure 2. Figure 5 illustrates the situation just as the attachment is being made.

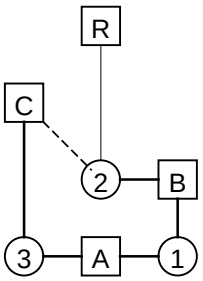


Figure 5 – Adding a new Root

While the new Root bridge may not yet be forwarding on its Port 2, it will be sending BPDUs since it thinks that port is Designated. So when B and C receive the first of these config BPDUs they will each identify Port 2 as their root port. Since B's port 2 is already forwarding, port 1 which was previously its root port but is now Designated can continue forwarding. However C will block port 3. So the forwarding connectivity shortly after R is added will be as shown in Figure 6.

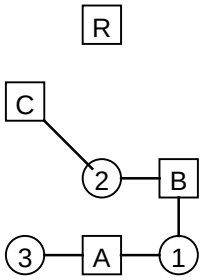


Figure 6 – New Root added

Note that this is not the final configuration. Eventually (root port costs of all the bridges in the network being equal) C will once more connect to LAN 3 and one of A's ports will be blocking. Assuming this to be port 3 the next and final steps in connectivity are as shown in Figure 7.

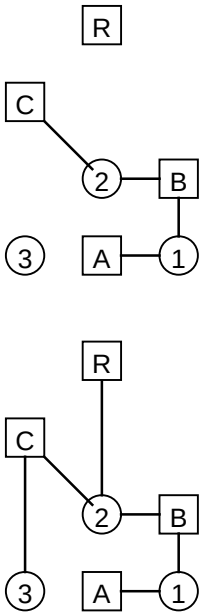


Figure 7 – Connectivity Transitions

Note that this is a different set of steps than for the completely unmodified algorithm, which would evolve as shown in Figure 8.

[22] In this set of diagrams the convention will be that bridge ports have the same number as the LANs they are attached to.

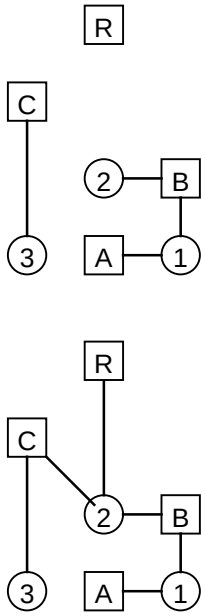
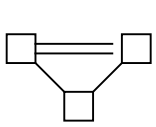


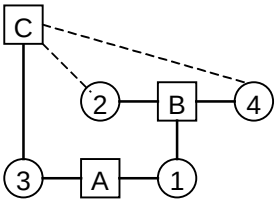
Figure 8 – Standard Connectivity Transitions

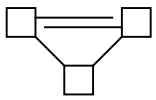
It there is a further LAN attached to bridge C it is a matter of opinion as to which set of steps are preferred. The revised algorithm does not guarantee connectivity during reconfiguration in every case that the standard algorithm would. In structured networks it will most often prove the superior, quite apart from providing continuous connectivity to the heart of the network.

Returning to the problem of attempting to move the Root around so as to create temporary loops. It can be observed that these have been defeated in Figures 2 thru 7 by the simple expedient of accompanying each gain in connectivity from a new root port by blocking the former root port. In fact this is logically enough to guarantee loop free behavior. However the suspicion lingers that perhaps two or more bridges could be arranged in a potential loop and new Root information simultaneously injected at several points and possibly selectively processed by some bridges with others accidentally dropping the BPDUs, all in such a way as to force a temporary loop. Consider network fragments 1 thru 7.

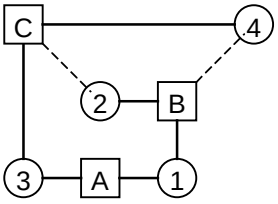


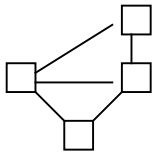
Fragment 1



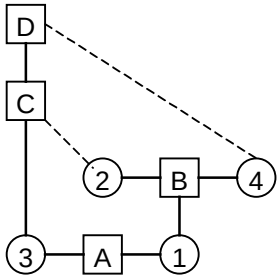


Fragment 2

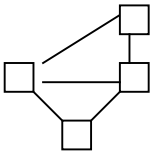




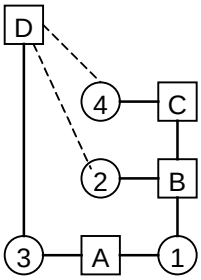
Fragment 3

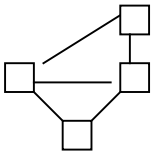


Each is represented by a sketch (to the left) and a full graph (to the right) laid out according to the metric ranking rules given above.

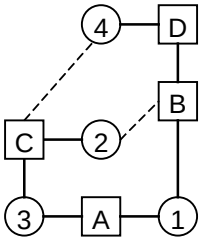


Fragment 4

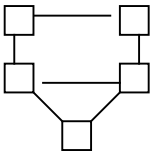




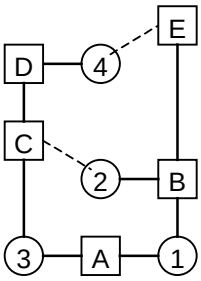
Fragment 5

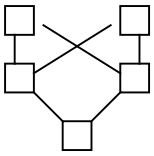


Of all these fragments, 1 thru 7, only in numbers 2 and 5 can a loop be forced by feeding different information to different bridges.

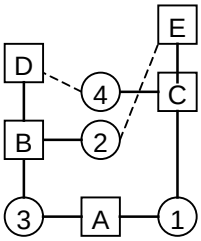


Fragment 6





Fragment 7



However the preconditions illustrated in 2 and 5 cannot exist in the steady state[23], and hence the ports required to have transitioned to forwarding for the loop hazard to exist will not in fact have done so. In fragment 2, C is designated for LAN 4 while B has the better claim. In fragment 5, D is designated for LAN 4 while C has the better claim.

In the particular case of a Backup Port, a port attaching a bridge to a LAN with the same metric value is blocked. This allows a loop to be created if that Backup Port receives information that would make it the Root Port. Figure 9 shows how this could occur.

····· Backup Port

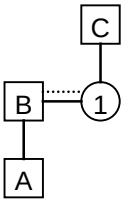


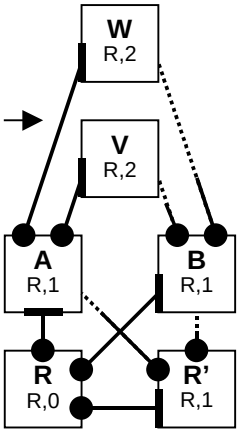
Figure 9 – Backup Port

B has two ports connected to LAN 1. The priority of bridge C is manged so that it becomes the Root bridge and transmits a config BPDU. Without special behavior for a Backup Port, a loop will be created even though the previous Root Port of Bridge B has become blocking. This is addressed above under the heading of 'High Availability with shared media'.

Some Real Examples

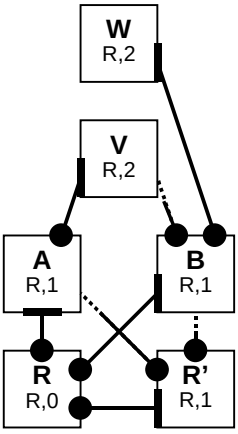
Some realistic example reconfigurations in redundantly configured structured wiring networks are shown below.

[23] The 'steady state' qualification is important. It is possible to cause a loop by a combination of luck and careful feeding continually changing spanning tree information into a network. However this is nothing new, and the time for which the information needs to be changed continues to be longer than twice Forward Delay. The original spanning tree algorithm relies on a notion of 'one change at a time', although the change might be major – applying power to all the bridges, for example.



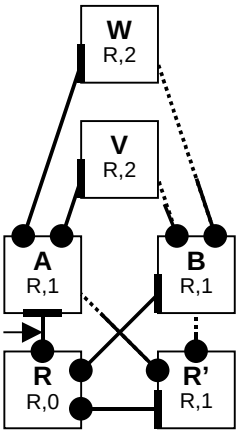
Spanning tree links (solid), redundant links (dashed).
Configuration: <root>,<root path cost>
Black blobs indicate designated bridge for link. Short black line indicates root port.
R is root, R' is backup root (next priority), A, then B next priority.

Link that will fail (A to W) indicated by arrow.

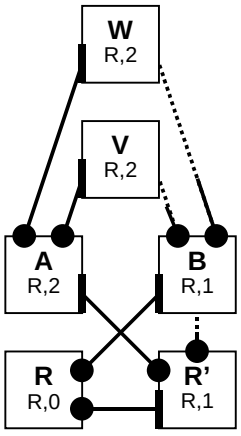


A-W link fails. B-W becomes root port for W.

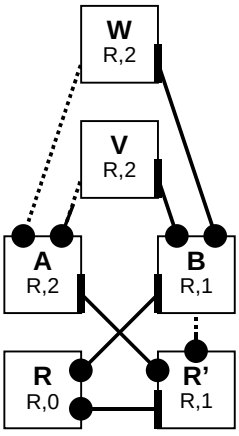
Structured Wiring Reconfiguration - Example 1



Link that will fail (R to A) indicated by arrow.

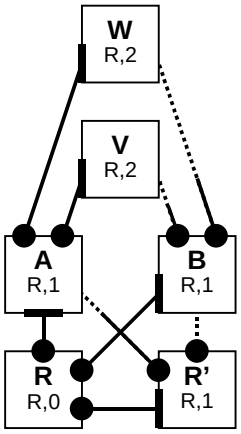


R-A link fails. R'-A becomes root port for A. V and W still holding onto prior information from A.

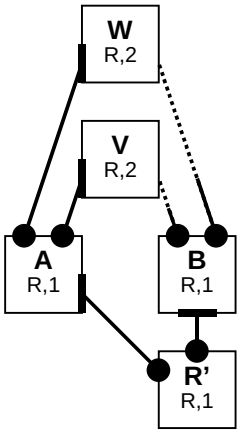


V and W time out information from A and select B-V and B-W as root ports respectively. Old root ports are blocked immediately to prevent loops.

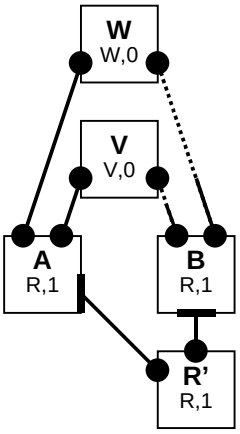
Structured Wiring Reconfiguration - Example 2



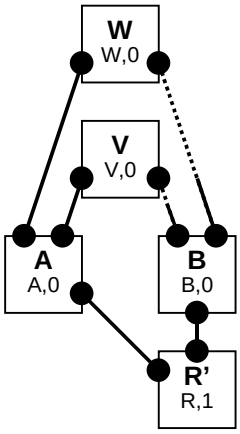
Bridge R will fail shortly.



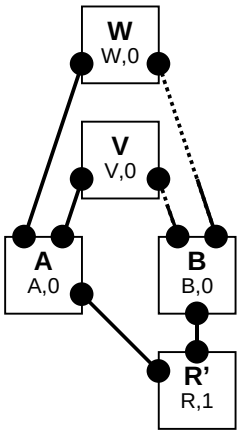
R-R', R-A and R-B links all fail.
R' continues as Designated Bridge for R'-A and R'-B.
A selects R'-A as new root port, B selects R'-B



Approximately spanning tree max age time later V and W timeout all information about R. They become Designated on both their downlinks, continuing to forward to A, and Listening on V-B and W-B.

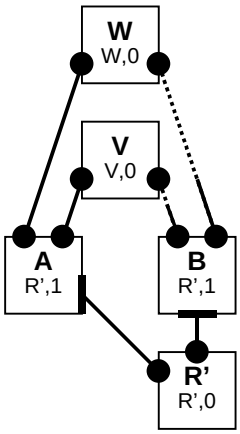


Very soon after this A and B also timeout information about R.

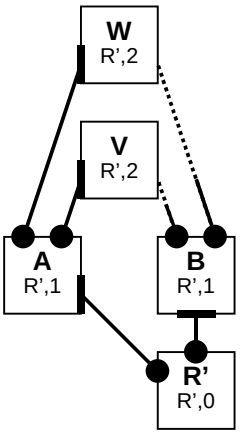


Very soon after this R' times out R, and starts to send BPDUs.

(V, W, A and B may have sent BPDUs already but these will be superseded by those now being sent by R').



A and B receive new BPDUs from R' and put their Root Port into Forwarding, forwarding BPDUs to V and W.



V and W receive new BPDUs from A and B, and put their Root Ports into Forwarding.

Reconfiguration complete with only momentary loss of connectivity.

Structured Wiring Reconfiguration - Example 3

Table 1A - Standard Port Roles, States, and transitions[24]

Port Role	**Disabled Port**	**Root Port**			**Designated Port**			**Alternate Port**
Port State	Disabled	Listening	Learning	Forwarding	Listening	Learning	Forwarding	Blocking
forwarding?	No	No	No	Yes	No	No	Yes	No
learning?	No	No	Yes	Yes	No	Yes	Yes	No
Events	**Transitions**							
become disabled	X	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
become root port	X	X	X	X	Root Port Listening[c]	Root Port Learning[c]	Root Port Forwarding	Root Port Listening[a]
become designated[25]	Designated Listening[a]	Designated Listening[c]	Designated Learning[c]	Designated Forwarding	X	X	X	Designated Listening[a]
become alternate[26]	X	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking	X
forward delay timer expiry	X	Root Port Learning[a]	Root Port Forwarding	X	Designated Learning[a]	Designated Forwarding	X	X

Table 1B - Standard Port States and transitions

Port State	**Disabled**	**Listening**	**Learning**	**Forwarding**	**Blocking**
forwarding?	No	No	No	Yes	No
learning?	No	No	Yes	Yes	No
Events	**Transitions**				
become disabled	X	Disabled	Disabled	Disabled	Disabled
become root port	X	--	--	--	Listening[a]
become designated	Listening[a]	--	--	--	Listening[a]
become alternate	X	Blocking[d]	Blocking[d]	Blocking	--
forward delay timer expiry	X	Learning[a]	Forwarding	X	X

[a] Start the forward delay timer.[b] Restart the forward delay timer. [c] Do not restart the forward delay timer (let it continue running). [d] Stop the forward delay timer.

[24] X means event can't happen. – means no action is taken, i.e. no state changes or timer changes.

[25] In the Disabled state this is 'become enabled'.

[26] Covers the transition described as 'become backup' below as Backup is not distinguished from Alternate in the standard.

Table 2 - High availability transitions for point-to-point networks only

Port Role	Disabled Port	Root Port	Designated Port			Alternate Port
Port State	Disabled	Forwarding	Listening	Learning	Forwarding	Blocking
forwarding?	No	Yes	No	No	Yes	No
learning?	No	Yes	No	Yes	Yes	No
Events	**Transitions**					
become disabled	X	Disabled	Disabled[d]	Disabled[d]	Disabled	Disabled
become root port	X	X	Root Port Forwarding[d]	Root Port Forwarding[d]	Root Port Forwarding	Root Port Forwarding
become designated	Designated Listening[a]	Designated Listening[a]	X	X	X	Designated Listening[a]
become alternate	X	Alternate Blocking	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking	X
forward delay timer expiry	X	X	Designated Learning[a]	Designated Forwarding	X	X

X Event cannot occur in this state. – Event has no effect in this state (not used in this table).
[a] Start the forward delay timer.[b] Restart the forward delay timer. [c] Do not restart the forward delay timer (let it continue running).
[d] Stop the forward delay timer.

Table 3 - High availability transitions for point-to-point networks only[27]

Port Role	Disabled Port	Root Port	Designated Port			Alternate Port
Port State	Disabled	Forwarding	Listening	Learning	Forwarding	Blocking
forwarding?	No	Yes	No	No	Yes	No
learning?	No	Yes	No	Yes	Yes	No
Events	**Transitions**					
become disabled	X	Disabled	Disabled[d]	Disabled[d]	Disabled	Disabled
become root port	X	X	retire root port Root Port Forwarding[d]	retire root port Root Port Forwarding[d]	retire root port Root Port Forwarding	retire root port Root Port Forwarding
retire root port	--	Alternate Port Blocking	--	--	--	--
become designated	Designated Listening[a]	Designated[28] Listening[a]	X	X	X	Designated Listening[a]
become alternate	X	Alternate Blocking	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking	X
forward delay timer expiry	X	X	Designated Learning[a]	Designated Forwarding	X	X

X Event cannot occur in this state. – Event has no effect in this state (not used in this table).
[a] Start the forward delay timer.[b] Restart the forward delay timer. [c] Do not restart the forward delay timer (let it continue running).
[d] Stop the forward delay timer.

[27] This table has exactly the same effect as Table 2, but separates the transition from being a 'Root Port' to another role into two stages, the first of which is 'retiring' the Root Port and is effected by an internal event 'retire root port' caused when another port is selected as the Root Port. The 'retire root port' event is 'sent' to all other ports.
[28] Occurs only when the root port information times out and there are no alternate ports, so the bridge becomes the root bridge. If another port were selected as root a 'retire root port' event would have preempted this event, though a subsequent 'become designated' event might still occur.

Table 4 – Enhanced high availability transitions for point-to-point networks

Port Role	**Disabled Port**	**Root Port**	**Designated Port**					**Alternate Port**
Port State	Disabled	Forwarding	Listening	Learning	Forwards	Forwarder	Forwarding	Blocking
forwarding?	No	Yes	No	No	Yes	Yes	Yes	No
learning?	No	Yes	No	Yes	Yes	Yes	Yes	No
Events	**Transitions**							
become disabled	X	Disabled	Disabled[d]	Disabled[d]	Disabled[d]	Disabled[d]	Disabled	Disabled
become root port	X	X	retire root port Root Port Forwarding[d]	retire root port Root Port Forwarding[d]	Root Port Forwarding[d]	Root Port Forwarding[d]	Root Port Forwarding	retire root port Root Port Forwarding
retire root port[29]	--	Designated Forwards[a]	--	--	Designated Listening[c]	Designated Learning[c]	--	--
become designated	Designated Listening[a]	Designated[30] Forwards[a]	X	X	X	X	X	Designated Listening[a]
become alternate	X	Alternate Blocking	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking	X
forward delay timer expiry	X	X	Designated Learning[a]	Designated Forwarding	Designated Forwarder[a]	Designated Forwarding	X	X

X Event cannot occur in this state. – Event has no effect in this state (not used in this table).
[a] Start the forward delay timer.[b] Restart the forward delay timer. [c] Do not restart the forward delay timer (let it continue running).
[d] Stop the forward delay timer.

[29] Must be followed by a 'become alternate' event immediately if the port is not to be Designated.
[30] Occurs only when the root port information times out and there are no alternate ports, so the bridge becomes the root bridge. If another port were selected as root a 'retire root port' event would have preempted this event, though a subsequent 'become designated' event might still occur.

Table 5 – Enhanced high availability transitions for point-to-point and shared media networks

Port Role	**Disabled Port**	**Root Port**	**Designated Port**					**Alternate Port**	**Backup Port**
Port State	Disabled	Forwarding	Listening	Learning	Forwards	Forwarder	Forwarding	Blocking	Blocking
forwarding?	No	Yes	No	No	Yes	Yes	Yes	No	No
learning?	No	Yes	No	Yes	Yes	Yes	Yes	No	No
Events	**Transitions**								
become disabled	X	Disabled	Disabled[d]	Disabled[d]	Disabled[d]	Disabled[d]	Disabled	Disabled	Disabled
become root port	X	X	retire root port Root Port Forwarding[d]	retire root port Root Port Forwarding[d]	Root Port Forwarding[d]	Root Port Forwarding[d]	Root Port Forwarding	retire root port Root Port Forwarding	retire root port retire primary Root Port Forwarding
retire root port[31]	--	Designated Forwards[a]	--	--	Designated Listening[c]	Designated Learning[c]	--	--	--
retire primary port[32]	--	X	Designated Listening[b]	Designated Listening[b]	Designated Listening[b]	Designated Listening[b]	Designated Listening[b]	--	--
become designated	Designated Listening[a]	Designated[33] Forwards[a]	X	X	X	X	X	Designated Listening[a]	Designated Listening[a]
become alternate	X	Alternate Blocking	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking	X	X
forward delay timer expiry	X	X	Designated Learning[a]	Designated Forwarding	Designated Forwarder[a]	Designated Forwarding	X	X	X

X Event cannot occur in this state. – Event has no effect in this state (not used in this table).
[a] Start the forward delay timer. [b] Restart the forward delay timer. [c] Do not restart the forward delay timer (let it continue running).
[d] Stop the forward delay timer.

[31] Must be followed by a 'become alterante' event immediately if the port is not to be Designated.
[32] Only the primary port corresponding to the Backup Port which is about to become Root Port is to be retired.
[33] Occurs only when the root port information times out and there are no alternate ports, so the bridge becomes the root bridge. If another port were selected as root a 'retire root port' event would have preempted this event, though a subsequent 'become designated' event might still occur.

Table 6 – Enhanced high availability transitions with glitch suppression

Port Role	**Disabled Port**	**Root Port**	**Designated Port**					**Alternate Port**		**Backup Port**
Port State	Disabled	Forwarding	Listening	Learning	Forwards	Forwarder	Forwarding	Forgetting	Blocking	Blocking
forwarding?	No	Yes	No	No	Yes	Yes	Yes	No	No	No
learning?	No	Yes	No	Yes	Yes	Yes	Yes	No	No	No
become disabled	X	Disabled	Disabled[d]	Disabled[d]	Disabled[d]	Disabled[d]	Disabled	Disabled	Disabled	Disabled
become root port	X	X	retire root port Root Port Forwarding[d]	retire root port Root Port Forwarding[d]	Root Port Forwarding[d]	Root Port Forwarding[d]	Root Port Forwarding	retire root port Root Port Forwarding[f]	retire root port Root Port Forwarding	retire root port retire primary Root Port Forwarding
retire root port[34]	--	Designated Forwards[a]	--	--	Designated Listening[c]	Designated Learning[c]	--	--	--	--
retire primary port[35]	--	X	Designated Listening[b]	Designated Listening[b]	Designated Listening[b]	Designated Listening[b]	Designated Listening[b]	--	--	--
become designated	Designated Listening[a]	Designated[36] Forwards[a]	X	X	X	X	X	Designated Forwarding[f]	Designated Listening[a]	Designated Listening[a]
become alternate	X	Alternate Blocking	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Blocking[d]	Alternate Forgetting[e]	--	X	X
forward delay timer expiry	X	X	Designated Learning[a]	Designated Forwarding	Designated Forwarder[a]	Designated Forwarding	X	X	X	X
forgetting timer expiry	X	X	X	X	X	X	X	Alternate Blocking	X	X

X Event cannot occur in this state. – Event has no effect in this state (not used in this table).
[a] Start the forward delay timer. [b] Restart the forward delay timer. [c] Do not restart the forward delay timer (let it continue running). [d] Stop the forward delay timer.
[e] Start the forgetting timer. [f] Stop the forgetting timer.

[34] Must be followed by a 'become alterante' event immediately if the port is not to be Designated.
[35] Only the primary port corresponding to the Backup Port which is about to become Root Port is to be retired.
[36] Occurs only when the root port information times out and there are no alternate ports, so the bridge becomes the root bridge. If another port were selected as root a 'retire root port' event would have preempted this event, though a subsequent 'become designated' event might still occur.