Speedy Tree Protocol

Mick Seaman

The IEEE Std. 802.1D Spanning Tree Protocol updates the protocol information held by each bridge by propagating better or newer information received from the root, and by ageing out old information. Thus “good news” – the availability of a better root or link to the root – travels quickly, while “bad news” – failure of a link or bridge travels slowly. The timing out of information is necessarily based on a worst case estimate, and factors directly into reconfiguration times.

This note proposes changes to the details of configuration message reception and propagation. These changes keep the familiar standard message formats, management parameters, and basic algorithm, but significantly improve reconfiguration performance. They are additional to and compatible with the recent “High Availability Spanning Tree” proposal.

Information previously received is expired immediately on link failure. In addition a configuration message from a designated bridge is always accepted even if it contains inferior information. Spanning tree recomputation occurs on both these events and may cause changes in root and designated ports. Changed information on designated ports is propagated to other bridges.

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. Tie breakers 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.

Configuration messages are originated periodically by the root and this information is distributed to all other bridges as follows. Better information received by a bridge port replaces that previously recorded, and is propagated further if it is the best that that bridge has recorded for any port[1]. All information has a maximum age so current information will be forgotten eventually, if the root or a bridge or link on the shortest path to it fails. Periodic message transmission by the root and potential roots together with information ageing ensures that the spanning tree maintains full loop-free connectivity even as bridges and links fail, or are added and removed from the network.

The maximum age of spanning tree information may be managed precisely to accommodate worst case message propagation delays, lost messages, the maximum number of bridges between the root and any LAN in the network, and their estimated adjustments to the message age. However, in most cases generous worst case “out of the box” defaults are used. Either way the operation of the protocol has the effect that “good news” – the availability of a better root or link to the root travels – travels quickly, while “bad news” – failure of a link or bridge – travels slowly.[2]

Unfortunately bridge or link failure is always bad news. To initiate reconfiguration, a bridge ages out current information, while receiving no better message. A bridge close to the root uses the same maximum information age as one at the network edge, so the detection time is set by worst case propagation times or defaults. Even if a bridge were to use local link specific failure detection, other bridges will discard this bad news until they have aged out the original better information.

This note proposes modifications to the spanning tree algorithm to allow bad news to propagate quickly. Specifically, a bridge will process inferior information sent by the designated bridge for each LAN. In an additional change, bridges use a per port hello timer to stimulate information propagation, setting it to suit local link characteristics. This enables early link failure detection.

If all[3] bridges implement these changes the Maximum Age parameter no longer contributes to reconfiguration delays[4]. Further, Forward

[1] “Better” means information from a higher priority root, or from the current root along a shorter (lower cost) path, or simply more recent information from the current root at the current path cost.

[2] In fact bad news cannot travel faster than the very worst case for good news by design. Since the best case propagation times under light or typical loads are very different from the worst under extreme loads, the expected difference is significant.

[3] If only some of the bridges in the network implement the changes, their effect is at worst harmless.

[4] An oversimplification. Large values of Maximum Age can delay the process of purging old information from the network, see the

Delay can be substantially reduced, since the delay in transitioning a bridge port from forwarding to blocking is set by the worst case information propagation time.

No changes are proposed to the format of the BPDUs[5] specified in 802.1D, and the algorithm is still very much the familiar spanning tree.

References

[1] ANSI/IEEE Std 802.1D, 1998 Edition.

[2] High Availability Spanning Tree, Rev 1.1, 10/26/98. Mick Seaman

Accepting Inferior Information

The basic improvement is that a BPDU sent on a LAN by the current Designated Bridge[6] is accepted and processed, even if it is inferior to information previously received.

In addition to modifying message acceptance in this way, a number of further detailed changes are required, since a wider variety of outcomes are possible when a message is accepted. The reception port could have been the Root Port for the bridge, but may be so no longer. Indeed the receiving bridge may find itself Designated on the reception port[7]. The existing code in 802.1D attempts a middle course between identifying the limited set of outcomes possible for each protocol event, and completely recomputing on every event. While distinct cases can be identified, all the actions previously possible on message ageing can now be required on reception.

Coincidentally these changes clear up a long standing defect in 802.1D, i.e. the current lack of specification of the action to be taken on receipt of a BPDU whose current age is already equal to or greater than its maximum age.

Expiring Information

Spanning tree information received on the Root Port, an Alternate Port, or a Backup Port[8] is expired and the spanning tree recomputed if:

(a) the link attached to that port has failed[9]

(b) the age of received information exceeds its accompanying Max Age[10]

(c) more than twice the Hello Time signaled with the received information has elapsed since it was received[11], and the receiving bridge is configured to assume that the transmitter is operating a per link hello timer[12].

Propagating Information

If information is to be propagated rapidly, neither an individual bridge nor the network as a whole should be left in an inconsistent quiescent state after the reception of inferior information.

For individual bridges it is an explicit goal that there are no management visible states[13] that appear strange to a devotee of the standard algorithm. This is achieved by correctly computing the new state.

For the network as a whole the goal is to propagate information upon change to resolve inconsistencies quickly[14]. In particular retransmissions from the current root cannot be relied upon to do that job, since it may have failed.

A BPDU is sent at least once per link hello time in order to provide a link “keep alive” functionality without introducing extra protocol.

Following a configuration update for any reason (message reception, message expiry, or management change) a BPDU is sent on every port for which the bridge was Designated prior to the change if the information on the port has been updated.

Additionally, if a BPDU is received on a port that continues or becomes the Root Port, BPDUs are transmitted on all ports for which the bridge is

discussion on “burning out” information. The important remains that Max Age is no longer directly additive to the reconfiguration time.

[5] A BPDU is a Bridge Protocol Data Unit, i.e. the frame that carries the spanning tree protocol information.

[6] To be more accurate this should read “the information sent by the current Designated Port”. The designated bridge may have two ports attached to the same shared media.

[7] This may happen to several bridges at once on a shared media LAN. The result will be that they all send BPDUs announcing themselves as Designated, which will cause the new Designated Bridge to be chosen.

[8] A Backup Port is simply an alternate port, i.e. neither designated nor root, where the designated port for the attached LAN belongs to the same bridge.

[9] If P802.3ad, Link Aggregation, is in operation only failures at the aggregate port level are relevant, and the last physical link has to fail before the spanning tree should react. The current proposal does support changing the Root Port Cost for a receiving root port in response to failure of an underlying physical link and recalculating the spanning tree in consequence. The author does not want to advocate this approach which negates some of the availability benefits provided by link aggregation.

[10] As per the existing 802.1D specification. The detailed changes however allow for the possibility that the information has already expired, so it is possible for a bridge to become the root as a consequence of receiving a message.

[11] The receiving implementation is also responsible for allowing for any variance in its own timeliness in taking note of received BPDUs.

[12] I would like to be able to detect this without invoking configuration. This early timeout provides a link keep alive functionality without the need for additional messages.

[13] With the possible exception of whether the hello timer is running or not, which can easily be concealed.

[14] In the redundant network topologies now typical for new installations, the High Availability Spanning Tree proposal by itself may provide equally short periods of service interruption, but the entire reconfiguration will take longer to complete. Using both proposals provides the best of both worlds.

Designated after updating the configuration. Further, a bridge transmits BPDUs on all ports after first becoming or believing itself to be the root.

These rules ensure rapid propagation of configuration information without adding excessively to the total number of BPDUs transmitted[15][16]. They differ from the 802.1D rules in two respects:

(a) a bridge may send BPDUs in additional circumstances without receiving a message from the root

(b) a bridge does not reply immediately to inferior information.

The reply() procedure has been removed because it leads to excessively “chatty” behavior when the port on which the reply was to be sent was previously the root port but is no longer[17]. With the introduction of per link hello timers, the process of contradicting bad information arising from message loss no longer relies on the next configuration message propagating all the way from the root. The timeliness of information distribution, which was the goal of the reply procedure, is thus already assured.

Burning out Information

Accepting and propagating new information from designated bridges allows spanning tree changes to be propagated soon after they are detected.

If there are no loops in the physical topology the old information will obviously be driven out to, and out of, the edge switches[18].

Where loops in the physical topology occur, there would appear to be a risk of old information[19] circulating around these loops, increasing path cost and message age as it circulates. Fortunately 802.1D mandates that the age of information in BPDUs never be underestimated so that information that returns to a bridge on the path from its original source will find better information there already[20], unless yet worse information has been propagated to that point.

To guard against this last eventuality[21], this proposal mandates a minimum increment to message age on each transmission by a bridge of at least 1/16th of max age[22]. Doing so ensures that circulating information is “burnt out” of the network if there is no bridge or bridge port remaining that is the source of the information.[23]

It is important to ensure that old information is guaranteed to be aged out before forwarding delays are complete. Otherwise the dynamically circulating information could create and sustain a data loop for a period. However the removal of old information is now achieved as rapidly as messages can be forwarded in any part of the network with redundant physical connectivity.[24]

To ensure that lost messages do not halt the burning out of old information, the link specific hello timers will cause the burning out process to continue if it has stopped.

Comparison with RIP v2

The proposed improvements make the operation of the Spanning Tree Protocol much closer to RIPv2, though of course for only one routed destination – the root. STP already ensures that information received on a root port is not reflected out of that port. This is equivalent to “split horizon”. The process of “burning out” information is essentially the same as “counting to infinity” where infinity is 16 so far as hop counts are concerned.

Fortunately STP is only concerned with one routed destination (see above), so does not forward many messages. This allows the new information distribution and burning out processes to operate on an event driven basis,

[15] They fall short of ensuring complete propagation of information without any further timer expiry in the network. Continuing the use of the reply procedure would have achieved that, but at the expense of transmitting a considerable number of BPDUs in richly connected topologies – all assumed received without loss. The design has to strike a balance between responsiveness, peak processing and buffering demands, and average demands. The question is “what timeliness can be achieved at a given level of resources?”. Adopting more aggressive timers is probably a better use of resources than continuing use of the reply procedure.

[16] Note that the generation and acceptance of inferior information does provide much better performance than simply waiting on timer expiry even if that is done on a link by link basis. The latter only propagates through the network at a rate of one hop per expiry time.

[17] Due to the reception of inferior information from the bridge that was Designated for the LAN.

[18] But in this case spanning tree was not required in the first place.

[19] A memory of a root bridge that has failed sometime ago, for example.

[20] i.e. the information that gave rise to the returning message, so this provision doesn’t stop information “chasing its tail”.

[21] The spanning tree protocol is not proof against continually changing information, but this still appears to be a useful safety mechanism.

[22] 1/8th might be a better value. If all bridges decrement the age by the same amount this provides for the current recommended maximum bridge diameter of the network (7 bridges) without any contraints on the placement of the root. The downside is that there is now no scope for individually tuning bridge timings at different levels in the network hierarchy where maximum bridge diameters are used. The combination seems very unlikely.

[23] An alternative would be to cap root path cost in configuration messages, but that would not work as well in an environment where Gigabit links might be accidentally mixed with 10 Mb/s links.

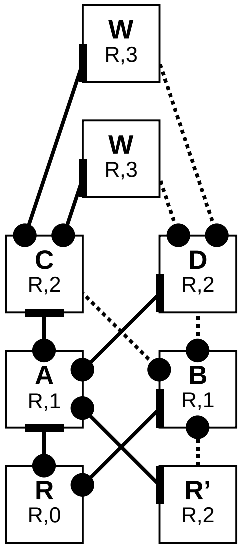
[24] This requires considering whether the current hold time of 1 second is appropriate, a subject that has already been raised in 802.1. One preferred fully redundant topology has loops of 4 bridges within it, so changing the hold timer specification to allow 2 or 3 BPDUs within a hold timer interval would meet the timeliness requirements.

with only a small hold timer to guard against over rapid transmission and loss at a receiver.

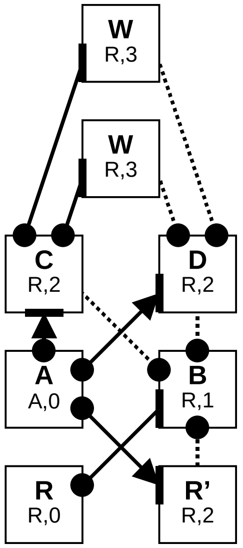
Examples

Two example reconfigurations are described below. Note that they deal with a richly connected topology such as might be deployed in a high availability scenario. Simpler topologies, such as backbone rings reach the final state more quickly.

Example 1

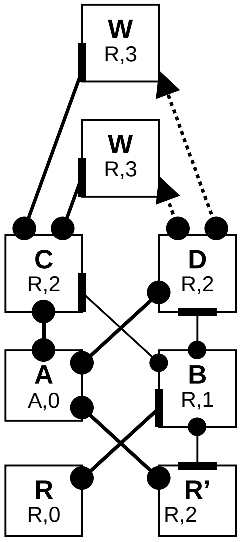


Spanning tree links (solid), redundant links (dashed).
Configuration: <root>,,<root path cost>
Black blobs indicate designated bridge for link. Short black line indicates root port.

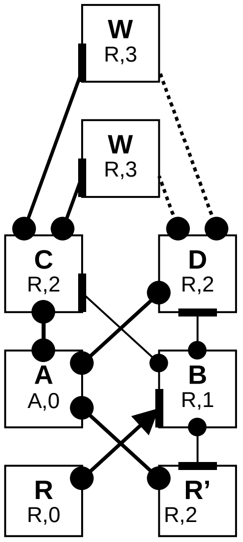


R-A link fails.
A removes root port info from configuration, concludes A is new root and sends PDUs to R', C, and D.

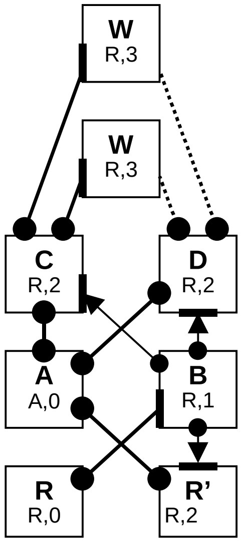
Arrows indicate PDUs in transit.



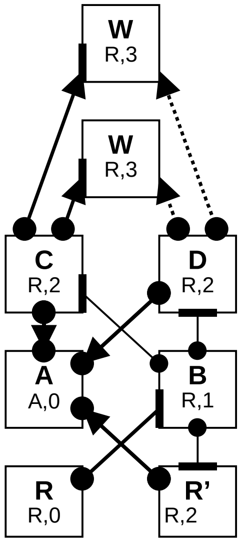
C concludes that root port is now C-B, begins to transition it to Forwarding, thinks itself designated on C-A. Similar actions at D and R'.



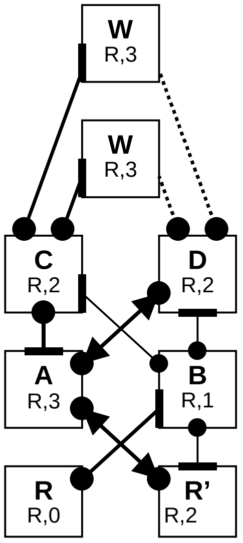
R, believing itself to be root, transmits next.



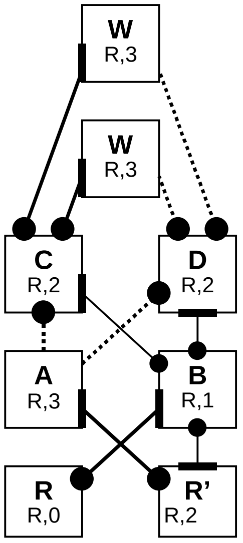
B forwards message from root to C, D, and R'.



C, D, and R' receive messages from B and forward to A and Ws.



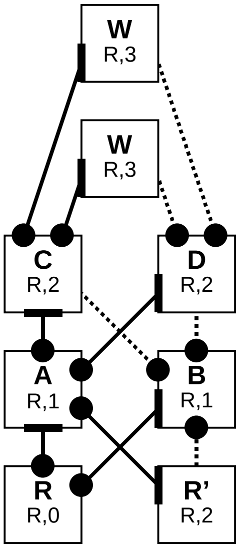
A receives message from C first (arbitrary), chooses A-C as new root and forwards message to D and R'. Ws receive messages from C and D without change.



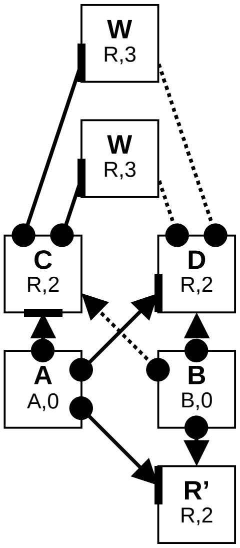
A receives messages from R' and D, moves root port to A-R' and transitions A-C and A-D to Blocking.
D and R' discard messages from A.

Final configuration (once C-B is Forwarding)

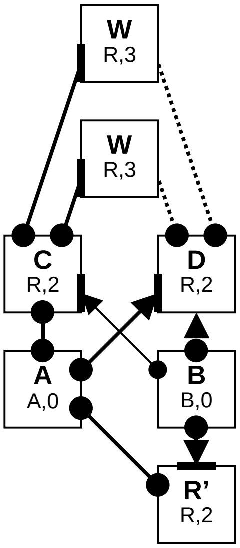
Example 2



Same initial configuration as before.

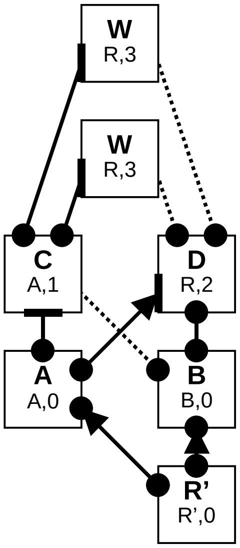


R fails, bringing down links R-A and R-B. A and B remove root port info from configuration, each concludes it is the new root and sends PDUs on all remaining ports.



Many possible next steps depending on order of processing of messages in transit.

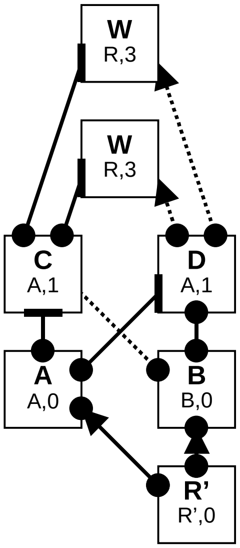
Say C receives from A first and moves root port to C-B. Similarly R’ receives from A and moves root port to R’-B.



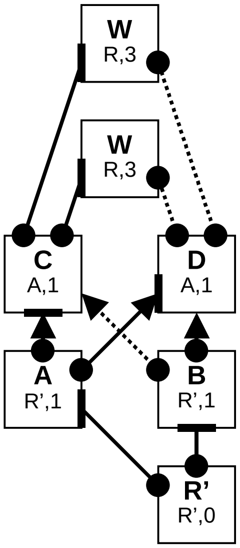
D receives from B, becomes designated on D-B.

C receives from B, moves root port to C-A with A as root.

R’ receives from B and becomes root, transmits on R’-A and R’-B.



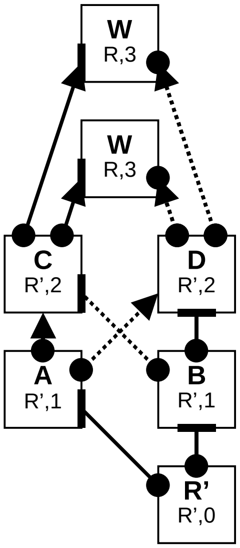
D receives from A, recognizes A as root and transmits to Ws.



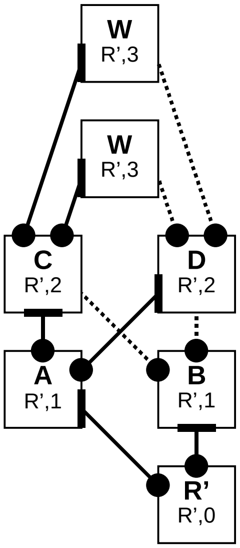
B receives from R’, recognizes R’ as root and transmits to C and D.

Similarly A receives from R’ and transmits to C and D.

Ws receive from D and become designated on W-D.



Assuming C and receive from B first, they acknowledge R’ as root, selecting C-B and D-B as root ports.and transmit to Ws,



Ws receive messages and adopt R’ as root.

C and D receive message from A and select C-A and D-A as root ports. Messages will be forwarded to Ws, but final configuration has been reached.