**TO:** **802.1**
**FROM:** **Tony Jeffree/Mick Seaman**
**DATE:** **5 March 2001**

**Les Bell has made us aware of a problem in P802.1w/D9 that can lead to loops being formed by Backup ports that become Root Port when their Bridge loses a connection to a prior Root and becomes Root itself. In this case the Backup port will hold spanning tree priority information from the prior root, and can feed this back into its own Bridge. While the information circulating between the Designated Port and the (now) Root Port on the same Bridge will age out, the rbwhile timer does not run for long enough to prevent a loop, nor is it desirable to extend the life of this timer to twice Forward Delay (which would fix this problem, but might cause certain scenarios involving shared media to reconfigure slowly).**

**The appropriate action is to ignore information received directly by a bridge from itself when selecting a Root Port. This was overlooked in .1wD9. A proposed set of changes (underline for inserted text, strikeout for text to be removed) to clauses 17.4.2.2 and 17.19.21 are included below.**

**The distributed version of the Visio simulation will be revised to include these changes. Details of this problem and the proposed changes to fix the text of P802.1w/D9 will be included in the Sponsor ballot resolution summary to be formally discussed, and a solution agreed, during the 802.1 meeting next week.**

### 17.4.2.2 Spanning tree priority vector type definitions

The *port priority vector* is the spanning tree vector held for the port when the reception of BPDUs and any pending update of information has been completed:

*port priority vector* = {*RootBridgeID* : *RootPathCost* : *DesignatedBridgeID* : *DesignatedPortID* : *BridgePortID*}

The *message priority vector* is the spanning tree priority vector conveyed in a received Configuration Message. For a Bridge $B$ receiving a Configuration Message on Port $P_B$ from a Designated Port $P_D$ on Bridge $D$ claiming a Root identifier of $R_D$ and a Root Path Cost of $RPC_D$:

*message priority vector* = $\{R_D : RPC_D : D : P_D : P_B\}$

This message priority vector is superior to the port priority vector if, and only if, the message priority vector is better than the port priority vector, or the *Designated Bridge ID* and *Designated Port ID* components are the same in which case the message has been transmitted from the same Designated Port as a previously received superior message, i.e. if:

$(R_D < RootBridgeID)) \;||$
$((R_D == RootBridgeID) \;\&\&\; (RPC_D < RootPathCost)) \;||$
$((R_D == RootBridgeID) \;\&\&\; (RPC_D == RootPathCost) \;\&\&\; (D < DesignatedBridgeID)) \;||$
$((R_D == RootBridgeID) \;\&\&\; ((RPC_D == RootPathCost)$
$\&\& (D == DesignatedBridgeID) \;\&\&\; ((P_D < DesignatedPortID)) \;||$
$((D == DesignatedBridgeID) \;\&\&\; (P_D == DesignatedPortID))$

is true.

If the message priority vector received in a valid BPDU is superior it will replace the current port priority vector.

A *root path priority vector* for the Port can be calculated from a received port priority vector, by adding the receiving Port's path cost $PPC_{PB}$ to the *Root Path Cost* component, and including the receiving Port's Port ID as the final component.

*root path priority vector*= $\{R_D : RPC_D + PPC_{PB} : D : P_D : P_B\}$

The bridge priority vector for a Bridge $B$ is the priority vector that would, with the *Designated Port ID* set equal to the transmitting *Port ID*, be used as the message priority vector in Configuration Messages transmitted on Bridge *B's* Designated Ports if $B$ was selected as the Root Bridge.

*bridge priority vector* = $\{B : 0 : B : 0 : 0\}$

The *root priority vector* for Bridge $B$ is the best priority vector of ~~all the root path priority vectors and~~ the set of priority vectors comprising the bridge priority vector plus all root path priority vectors whose Designated-BridgeID D is not equal to B.~~ Bridge *B* having been selected as the Root in the latter case.~~ In the case that the bridge priority vector is the best of this set of priority vectors, Bridge B has been selected as the Root. Assuming the best priority root path priority vector of this set to be that of port $P_B$ above, then:

*root priority vector* = $\{B : 0 : B : 0 : 0\}$ if $B$ is better than $R_D$, or
= $\{R_D : RPC_D + PPC_{PB} : D : P_D : P_B\}$ if $B$ is worse than $R_D$

The *designated priority vector* for a port $Q$ on Bridge $B$ is the root priority vector with *B's* Bridge Identifier $B$ substituted for the *DesignatedBridgeID* and *Q's* Port Identifier $Q_B$ substituted for the *DesignatedPortID* and *BridgePortID* components.

*designated priority vector*= $\{B : 0 : B : Q_B : Q_B\}$ if $B$ is better than $R_D$, or
= $\{R_D : RPC_D + PPC_{PB} : B : Q_B : Q_B\}$ if $B$ is worse than $R_D$

If the designated priority vector is better than the port priority vector, the Port will be the Designated Port for the attached LAN and the port priority vector will be updated. The message priority vector in RSTP BPDUs transmitted by a Port always comprises the first four components of the port priority vector of the Port, even if the Port is a Root Port.

### 17.19.21 updtRolesBridge()

This procedure calculates the following Spanning Tree priority vectors (17.4.2.2) and Spanning Tree timer values:

a) The *root path priority vector* for each Bridge Port that is not Disabled and has a *port priority vector* (portPriority plus portId - see 17.18.16 and 17.18.17) that has been recorded from a received message and has not yet aged out; and
b) The *root path times* (the value of portTimes - see 17.18.18) associated with each root path priority vector;
c) The Bridge's *root priority vector*, chosen as the best of the set of Spanning Tree priority vectors comprising the Bridge's own *bridge priority vector* (BridgePriority - see 17.17.3) plus all the calculated root path priority vectors whose DesignatedBridgeID component is not equal to the DesignatedBridgeID component of the Bridge's own bridge priority vector (see 17.4.2.2) ~~and the Bridge's own *bridge priority vector* (BridgePriority - see 17.17.3)~~; and
d) The Bridge's *root times*, determined as follows:
    1) If the chosen root priority vector is the bridge priority vector, *root times* is equal to BridgeTimes (see 17.17.4).
    2) If the chosen root priority vector is not the bridge priority vector, *root times* is equal to the value of *root path times* associated with the chosen root priority vector, with the Message Age component incremented by the greater of (1/16 of the Max Age component) and 1, rounded to the nearest whole second.
e) The *designated priority vector* for each port; and
f) The *designated times* for each Port (equal to the value of *root times*).

The first four components of the root priority vector are recorded in the rootPriority variable (17.17.6) for the Bridge. The fifth component of the root priority vector is recorded in the rootPortId variable (17.17.5) for the Bridge.

The root times are recorded in the rootTimes variable (17.17.7) for the Bridge.

The first four components of the designated priority vector for each Port are recorded in the designatedPriority variable (17.18.2) for that Port.

The designated times for each Port are recorded in the designatedTimes variable (17.18.3) for that Port.

The port role for each Port is assigned, and its port priority vector and spanning tree timer information are updated as follows:

g) If the Port is Disabled (infoIs = Disabled), selectedRole is set to DisabledPort. Otherwise:
h) If the port priority vector information was aged (infoIs = Aged), updtInfo is set and selectedRole is set to DesignatedPort;
i) If the port priority vector was derived from another port on the Bridge or from the Bridge itself as the Root Bridge (infoIs = Mine), selectedRole is set to DesignatedPort. Additionally, updtInfo is set if the port priority vector differs from the designated priority vector or the Port's associated timer parameters differ from those for the Root Port;
j) If the port priority vector was received in a Configuration Message and is not aged (infoIs = Received), and the root priority vector is now derived from it, selectedRole is set to RootPort and updtInfo is reset;
k) If the port priority vector was received in a Configuration Message and is not aged (infoIs = Received), the root priority vector is not now derived from it, the designated priority vector is not higher than the port priority vector, and the designated bridge and designated port components of the port priority vector do not reflect another port on this bridge, selectedRole is set to AlternatePort and updtInfo is reset;
l) If the port priority vector was received in a Configuration Message and is not aged (infoIs = Received), the root priority vector is not now derived from it, the designated priority vector is not higher than the port

priority vector, and the designated bridge and designated port components of the port priority vector reflect another port on this bridge, selectedRole is set to BackupPort and updtInfo is reset.