Understanding Rapid Spanning Tree Protocol (802.1w)

TAC Notice: What's Changing on TAC Web

Contents

Introduction

The 802.1D Spanning Tree Protocol (STP) standard was designed at a time when the recovery of connectivity after an outage within a minute or so was considered adequate performance. With the advent of Layer 3 switching in LAN environments, bridging now competes with routed solutions where protocols, such as Open Shortest Path First (OSPF) and Enhanced Interior Gateway Routing Protocol (EIGRP), are able to provide an alternate path in less time.

Cisco enhanced the original 802.1D specification with features such as Uplink Fast, Backbone Fast, and Port Fast to speed up the convergence time of a bridged network. The drawback is that these mechanisms are proprietary and need additional configuration.

Rapid Spanning Tree Protocol (RSTP; IEEE 802.1w) can be seen as an evolution of the 802.1D standard more than a revolution. The 802.1D terminology remains primarily the same. Most parameters have been left unchanged so users familiar with 802.1D can rapidly configure the new protocol comfortably. In most cases, RSTP performs better than proprietary extensions of Cisco without any additional configuration. 802.1w can also revert back to 802.1D in order to interoperate with legacy bridges on a per-port basis. This drops the benefits it introduces.

The new edition of the 802.1D standard, IEEE 802.1D-2004, incorporates IEEE 802.1t-2001 and IEEE 802.1w standards.

This document provides information about the enhancements added by RSTP to the previous 802.1D standard.

Support of RSTP in Catalyst Switches

This table shows the support of RSTP in Catalyst switches, and the minimum software required for that support.

Catalyst Platform	MST w/ RSTP	RPVST+ (also known as PVRST+)
Catalyst 2900 XL / 3500 XL	Not available.	Not available.
Catalyst 2940	12.1(20)EA2	12.1(20)EA2
Catalyst 2950/2955/3550	12.1(9)EA1	12.1(13)EA1
Catalyst 2970/3750	12.1(14)EA1	12.1(14)EA1
Catalyst 3560	12.1(19)EA1	12.1(19)EA1
Catalyst 3750 Metro	12.1(14)AX	12.1(14)AX
Catalyst 2948G-L3/4908G-L3	Not available.	Not available.
Catalyst 4000/2948G/2980G (CatOS)	7.1	7.5
Catalyst 4000/4500 (IOS)	12.1(12c)EW	12.1(19)EW
Catalyst 5000/5500	Not available.	Not available.
Catalyst 6000/6500	7.1	7.5
Catalyst 6000/6500 (IOS)	12.1(11b)EX, 12.1(13)E, 12.2 (14)SX	12.1(13)E

Catalyst 8500	Not available.	Not available.

New Port States and Port Roles

The 802.1D is defined in these five different port states:

- disabled
- listening
- learning
- blocking
- forwarding

See the table in the Port States section of this document for more information.

The state of the port is mixed, whether it blocks or forwards traffic, and the role it plays in the active topology (root port, designated port, and so on). For example, from an operational point of view, there is no difference between a port in the blocking state and a port in the listening state. Both discard frames and do not learn MAC addresses. The real difference lies in the role the spanning tree assigns to the port. It can safely be assumed that a listening port is either designated or root and is on its way to the forwarding state. Unfortunately, once in the forwarding state, there is no way to infer from the port state whether the port is root or designated. This contributes to demonstrate the failure of this state-based terminology. RSTP decouples the role and the state of a port to address this issue.

Port States

There are only three port states left in RSTP that correspond to the three possible operational states. The 802.1D disabled, blocking, and listening states are merged into a unique 802.1w discarding state.

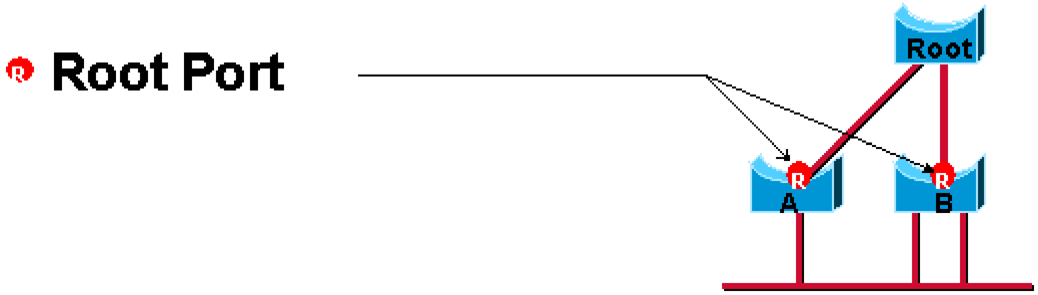
STP (802.1D) Port State	RSTP (802.1w) Port State	Is Port Included in Active Topology?	Is Port Learning MAC Addresses?
Disabled	Discarding	No	No
Blocking	Discarding	No	No
Listening	Discarding	Yes	No
Learning	Learning	Yes	Yes
Forwarding	Forwarding	Yes	Yes

Port Roles

The role is now a variable assigned to a given port. The root port and designated port roles remain, while the blocking port role is split into the backup and alternate port roles. The Spanning Tree Algorithm (STA) determines the role of a port based on Bridge Protocol Data Units (BPDUs). In order to simplify matters, the thing to remember about a BPDU is there is always a method to compare any two of them and decide whether one is more useful than the other. This is based on the value stored in the BPDU and occasionally on the port on which they are received. This considered, the information in this section explains practical approaches to port roles.

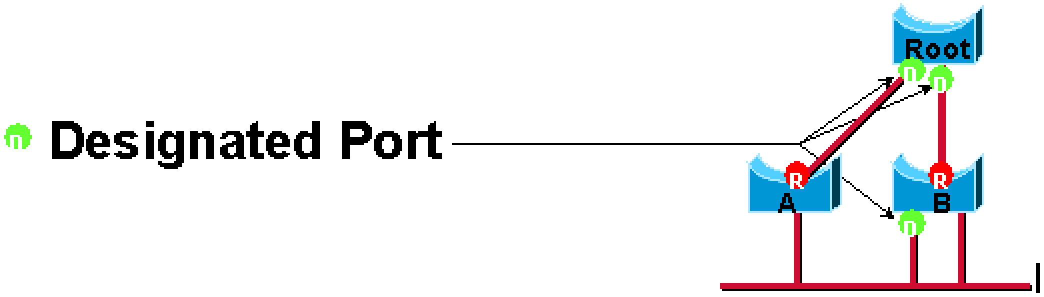
Root Port Roles

- The port that receives the best BPDU on a bridge is the root port. This is the port that is the closest to the root bridge in terms of path cost. The STA elects a single root bridge in the whole bridged network (per-VLAN). The root bridge sends BPDUs that are more useful than the ones any other bridge sends. The root bridge is the only bridge in the network that does not have a root port. All other bridges receive BPDUs on at least one port.



Designated Port Role

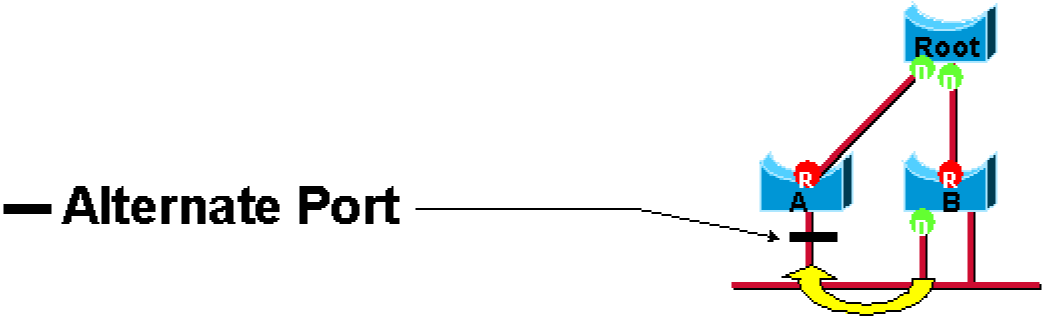
- A port is designated if it can send the best BPDU on the segment to which it is connected. 802.1D bridges link together different segments, such as Ethernet segments, to create a bridged domain. On a given segment, there can only be one path toward the root bridge. If there are two, there is a bridging loop in the network. All bridges connected to a given segment listen to the BPDUs of each and agree on the bridge that sends the best BPDU as the designated bridge for the segment. The port on that bridge that corresponds is the designated port for that segment.



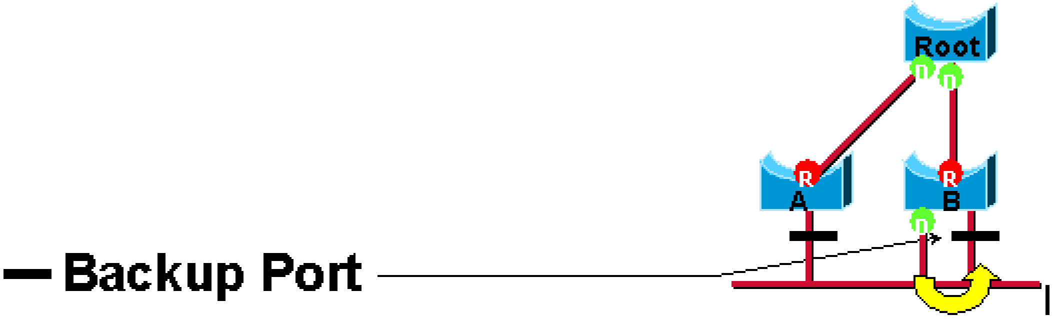
Alternate and Backup Port Roles

- These two port roles correspond to the blocking state of 802.1D. A blocked port is defined as not being the designated or root port. A blocked port receives a more useful BPDU than the one it sends out on its segment. Remember that a port absolutely needs to receive BPDUs in order to stay blocked. RSTP introduces these two roles for this purpose.

- An alternate port receives more useful BPDUs from another bridge and is a port blocked. This is shown in this diagram:



- A backup port receives more useful BPDUs from the same bridge it is on and is a port blocked. This is shown in this diagram:



This distinction is already made internally within 802.1D. This is essentially how Cisco UplinkFast functions. The rationale is that an alternate port provides an alternate path to the root bridge and therefore can replace the root port if it fails. Of course, a backup port provides redundant connectivity to the same segment and cannot guarantee an alternate connectivity to the root bridge. Therefore, it is excluded from the uplink group.

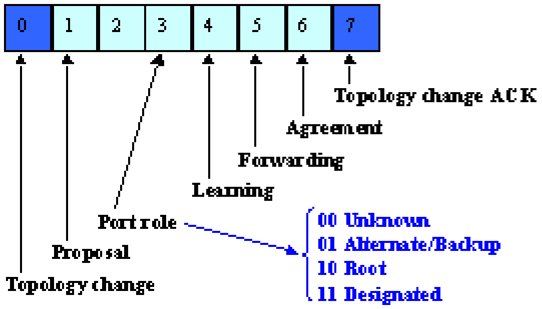
As a result, RSTP calculates the final topology for the spanning tree that uses the same criteria as 802.1D. There is absolutely no change in the way the different bridge and port priorities are used. The name blocking is used for the discarding state in Cisco implementation. CatOS releases 7.1 and later still display the listening and `learning` states. This gives even more information about a port than the IEEE standard requires. However, the new feature is now there is a difference between the role the protocol determines for a port and its current state. For example, it is now perfectly valid for a port to be designated and blocking at the same time. While this typically occurs for very short periods of time, it simply means that this port is in a transitory state towards the designated forwarding state.

New BPDU Format

Few changes have been introduced by RSTP to the BPDU format. Only two flags, Topology Change (TC) and TC Acknowledgment (TCA), are defined in 802.1D. However, RSTP now uses all six bits of the flag byte that remain in order to perform:

- Encode the role and state of the port that originates the BPDU

- Handle the proposal/agreement mechanism



Another important change is that the RSTP BPDU is now of type 2, version 2. The implication is that legacy bridges must drop this new BPDU. This property makes it easy for a 802.1w bridge to detect legacy bridges connected to it.

New BPDU Handling

BPDU are Sent Every Hello-Time

BPDU are sent every hello-time, and not simply relayed anymore. With 802.1D, a non-root bridge only generates BPDUs when it receives one on the root port. In fact, a bridge relays BPDUs more than it actually generates them. This is not the case with 802.1w. A bridge now sends a BPDU with its current information every <hello-time> seconds (2 by default), even if it does not receive any from the root bridge.

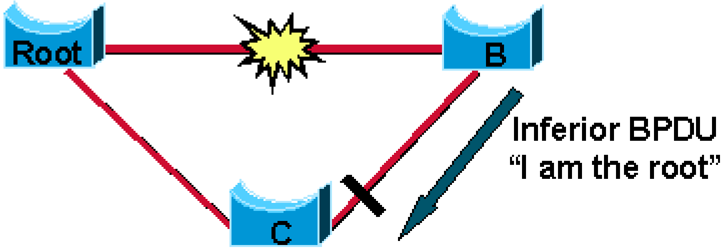
Faster Aging of Information

On a given port, if hellos are not received three consecutive times, protocol information can be immediately aged out (or if max_age expires). Because of the previously mentioned protocol modification, BPDUs are now used as a keep-alive mechanism between bridges. A bridge considers that it loses connectivity to its direct neighbor root or designated bridge if it misses three BPDUs in a row. This fast aging of the information allows quick failure detection. If a bridge fails to receive BPDUs from a neighbor, it is certain that the connection to that neighbor is lost. This is opposed to 802.1D where the problem might have been anywhere on the path to the root.

Note: Failures are detected even much faster in case of physical link failures.

Accepts Inferior BPDUs

This concept is what makes up the core of the BackboneFast engine. The IEEE 802.1w committee decided to incorporate a similar mechanism into RSTP. When a bridge receives inferior information from its designated or root bridge, it immediately accepts it and replaces the one previously stored.



Because Bridge C still knows the root is alive and well, it immediately sends a BPDU to Bridge B that contains information about the root bridge. As a result, Bridge B does not send its own BPDUs and accepts the port that leads to Bridge C as the new root port.

Rapid Transition to Forwarding State

Rapid transition is the most important feature introduced by 802.1w. The legacy STA passively waited for the network to converge before it turned a port into the forwarding state. The achievement of faster convergence was a matter of tuning the conservative default parameters (forward delay and max_age timers) and often put the stability of the network at stake. The new rapid STP is able to actively confirm that a port can safely transition to the forwarding state without having to rely on any timer configuration. There is now a real feedback mechanism that takes place between RSTP-compliant bridges. In order to achieve fast convergence on a port, the protocol relies upon two new variables: edge ports and link type.

Edge Ports

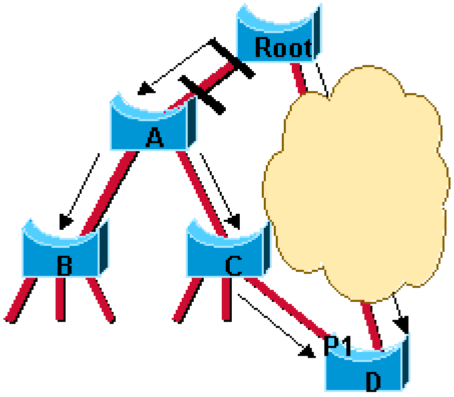
The edge port concept is already well known to Cisco spanning tree users, as it basically corresponds to the PortFast feature. All ports directly connected to end stations cannot create bridging loops in the network. Therefore, the edge port directly transitions to the forwarding state, and skips the listening and learning stages. Neither edge ports or PortFast enabled ports generate topology changes when the link toggles. An edge port that receives a BPDU immediately loses edge port status and becomes a normal spanning tree port. At this point, there is a user-configured value and an operational value for the edge port state. The Cisco implementation maintains that the *PortFast* keyword be used for edge port configuration. This makes the transition to RSTP simpler.

Link Type

RSTP can only achieve rapid transition to the forwarding state on edge ports and on point-to-point links. The link type is automatically derived from the duplex mode of a port. A port that operates in full-duplex is assumed to be point-to-point, while a half-duplex port is considered as a shared port by default. This automatic link type setting can be overridden by explicit configuration. In switched networks today, most links operate in full-duplex mode and are treated as point-to-point links by RSTP. This makes them candidates for rapid transition to the forwarding state.

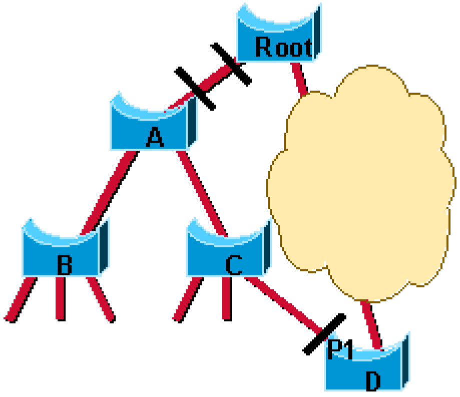
Convergence with 802.1D

This diagram illustrates how 802.1D deals with a new link that is added to a bridged network:



New ports coming up on the root and switch A are immediately put in listening state, blocking traffic.
BPDUs from the root start propagating towards the leaves through A

In this scenario, a link between the root bridge and Bridge A is added. Suppose there already is an indirect connection between Bridge A and the root bridge (via C - D in the diagram). The STA blocks a port and disables the bridging loop. First, as they come up, both ports on the link between the root and Bridge A are put in the listening state. Bridge A is now able to hear the root directly. It immediately propagates its BPDUs on the designated ports, toward the leaves of the tree. As soon as Bridges B and C receive this new superior information from Bridge A, they immediately relay the information towards the leaves. In a few seconds, Bridge D receives a BPDU from the root and instantly blocks port P1.



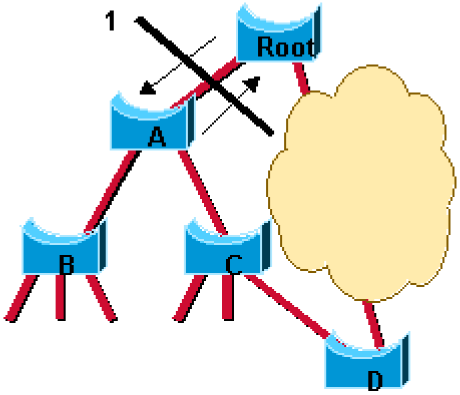
Very quickly, the BPDUs from the root reach D that immediately blocks its port P1.
The topology has now converged, though the network is disrupted for twice forward_delay.

Spanning tree is very efficient in how it calculates the new topology of the network. The only problem now is that twice the forward delay must elapse before the link between the root and Bridge A eventually ends up in the forwarding state. This means 30 seconds of disruption of traffic (the entire A, B, and C part of the network is isolated) because the 8021.D algorithm lacks a feedback mechanism to clearly advertise that the network converges in a matter of seconds.

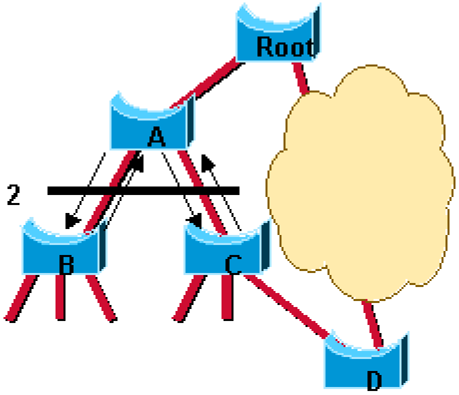
Convergence with 802.1w

Now, you can see how RSTP deals with a similar situation. Remember that the final topology is exactly the same as the one calculated by 802.1D (that is, one blocked port at the same place as before). Only the steps used to reach this topology have changed.

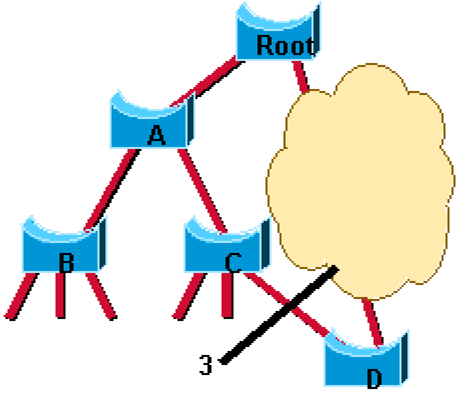
Both ports on the link between A and the root are put in designated blocking as soon as they come up. Thus far, everything behaves as in a pure 802.1D environment. However, at this stage, a negotiation takes place between Switch A and the root. As soon as A receives the BPDU of the root, it blocks the non-edge designated ports. This operation is called sync. Once this is done, Bridge A explicitly authorizes the root bridge to put its port in the forwarding state. This diagram illustrates the result of this process on the network. The link between Switch A and the root bridge is blocked, and both bridges exchange BPDUs.



Once Switch A blocks its non-edge designated ports, the link between Switch A and the root is put in the forwarding state and you reach the situation:



There still cannot be a loop. Instead of blocking *above* Switch A, the network now blocks below Switch A. However, the potential bridging loop is cut at a different location. This cut travels down the tree along with the new BPDUs originated by the root through Switch A. At this stage, the newly blocked ports on Switch A also negotiate a quick transition to the forwarding state with their neighbor ports on Switch B and Switch C that both initiate a sync operation. Other than the root port towards A, Switch B only has edge designated ports. Therefore, it has no port to block in order to authorize Switch A to go to the forwarding state. Similarly, Switch C only has to block its designated port to D. The state shown in this diagram is now reached:

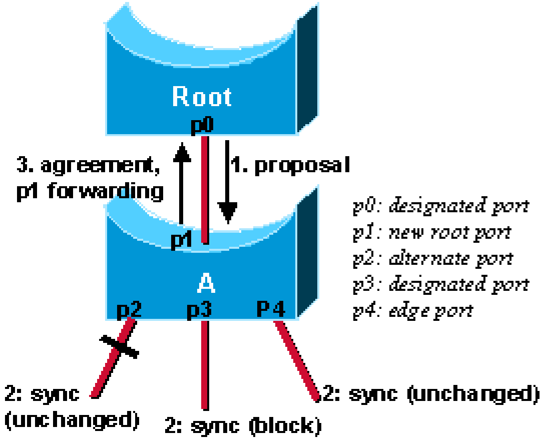


Remember that the final topology is exactly the same as the 802.1D example, which means that port P1 on D ends up blocking. This means that the final network topology is reached, just in the time necessary for the new BPDUs to travel down the tree. No timer is involved in this quick convergence. The only new mechanism introduced by RSTP is the acknowledgment that a switch can send on its new root port in order to authorize immediate transition to the forwarding state, and bypasses the twice-the-forward-delay long listening and learning stages. The administrator only needs to remember these to benefit from fast convergence:

- This negotiation between bridges is only possible when bridges are connected by point-to-point links (that is, full-duplex links unless explicit port configuration).
- Edge ports play an even more important role now that PortFast is enabled on ports in 802.1D. For instance, if the network administrator fails to properly configure the edge ports on B, their connectivity is impacted by the link between A and the root that comes up.

Proposal/Agreement Sequence

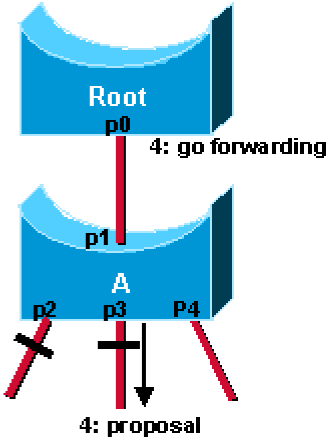
When a port is selected by the STA to become a designated port, 802.1D still waits twice <forward delay> seconds (2x15 by default) before it transitions it to the forwarding state. In RSTP, this condition corresponds to a port with a designated role but a blocking state. These diagrams illustrate how fast transition is achieved step-by-step. Suppose a new link is created between the root and Switch A. Both ports on this link are put in a designated blocking state until they receive a BPDU from their counterpart.



When a designated port is in a discarding or learning state (and only in this case), it sets the proposal bit on the BPDUs it sends out. This is what occurs for port p0 of the root bridge, as shown in step 1 of the preceding diagram. Because Switch A receives superior information, it immediately knows that p1 is the new root port. Switch A then starts a sync to verify that all of its ports are in-sync with this new information. A port is in sync if it meets either of these criteria:

- The port is in the blocking state, which means discarding in a stable topology.
- The port is an edge port.

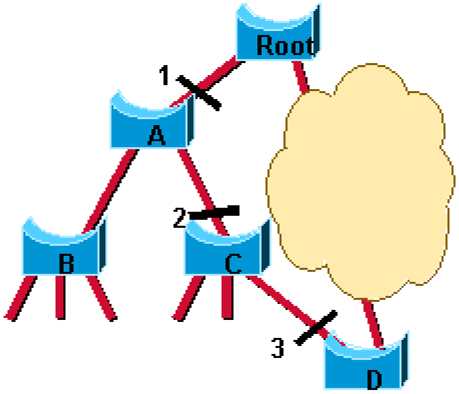
In order to illustrate the effect of the sync mechanism on different kind of ports, suppose there exists an alternate port p2, a designated forwarding port p3, and an edge port p4 on Switch A. Notice that p2 and p4 already meet one of the criteria. In order to be in sync (see step 2 of the preceding diagram), Switch A just needs to block port p3, and assign it the discarding state. Now that all of its ports are in sync, Switch A can unblock its newly selected root port p1 and send an agreement message to reply to the root. (see step 3). This message is a copy of the proposal BPDU, with the agreement bit set instead of the proposal bit. This ensures that port p0 knows exactly to which proposal the agreement it receives corresponds.



Once p0 receives that agreement, it can immediately transition to the forwarding state. This is step 4 of the preceding figure. Notice that port p3 is left in a designated discarding state after the sync. In step 4, that port is in the exact same situation as port p0 is in step 1. It then starts to propose to its neighbor, and attempts to quickly transition to the forwarding state.

- The proposal agreement mechanism is very fast, as it does not rely on any timers. This wave of handshakes propagates quickly towards the edge of the network, and quickly restores connectivity after a change in the topology.
- If a designated discarding port does not receive an agreement after it sends a proposal, it slowly transitions to the forwarding state, and falls back to the traditional 802.1D listening-learning sequence. This can occur if the remote bridge does not understand RSTP BPDUs, or if the port of the remote bridge is blocking.
- Cisco introduced an enhancement to the sync mechanism that allows a bridge to put only its former root port in the discarding state when it syncs. Details of how this mechanism works are beyond the scope of this document. However, one can safely assume that it is invoked in most common reconvergence cases. The scenario described in the Convergence with 802.1w section of

this document becomes extremely efficient, as only the ports on the path to the final blocked port are temporarily confused.



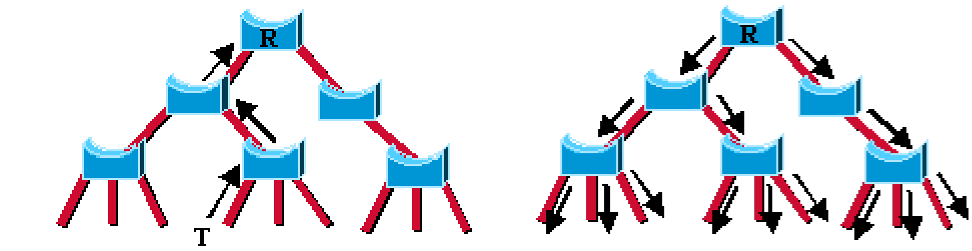
UplinkFast

Another form of immediate transition to the forwarding state included in RSTP is similar to the Cisco UplinkFast proprietary spanning tree extension. Basically, when a bridge loses its root port, it is able to put its best alternate port directly into the forwarding mode (the appearance of a new root port is also handled by RSTP). The selection of an alternate port as the new root port generates a topology change. The 802.1w topology change mechanism clears the appropriate entries in the Content Addressable Memory (CAM) tables of the upstream bridge. This removes the need for the dummy multicast generation process of UplinkFast.

UplinkFast does not need to be configured further because the mechanism is included natively and enabled in RSTP automatically.

New Topology Change Mechanisms

When an 802.1D bridge detects a topology change, it uses a reliable mechanism to first notify the root bridge. This is shown in this diagram:



A topology change is generated on point T.
1st step: A TCN is going up to the root.

2nd step: the root advertises the TC for max-age+ forward delay.

Once the root bridge is aware of a change in the topology of the network, it sets the TC flag on the BPDUs it sends out, which are then relayed to all the bridges in the network. When a bridge receives a BPDU with the TC flag bit set, it reduces its bridging-table aging time to forward delay seconds. This ensures a relatively quick flush of stale information. Refer to Understanding Spanning-Tree Protocol Topology Changes for more information on this process. This topology change mechanism is deeply remodeled in RSTP. Both the detection of a topology change and its propagation through the network evolve.

Topology Change Detection

In RSTP, only non-edge ports that move to the forwarding state cause a topology change. This means that a loss of connectivity is not considered as a topology change any more, contrary to 802.1D (that is, a port that moves to blocking no longer generates a TC). When a RSTP bridge detects a topology change, these occur:

- It starts the TC While timer with a value equal to twice the hello-time for all its non-edge designated ports and its root port, if necessary.
- It flushes the MAC addresses associated with all these ports.

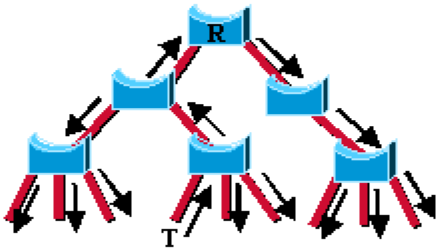
Note: As long as the TC While timer runs on a port, the BPDUs sent out of that port have the TC bit set. BPDUs are also sent on the root port while the timer is active.

Topology Change Propagation

When a bridge receives a BPDU with the TC bit set from a neighbor, these occur:

- It clears the MAC addresses learned on all its ports, except the one that receives the topology change.
- It starts the TC While timer and sends BPDUs with TC set on all its designated ports and root port (RSTP no longer uses the specific TCN BPDU, unless a legacy bridge needs to be notified).

This way, the TCN floods very quickly across the whole network. The TC propagation is now a one step process. In fact, the initiator of the topology change floods this information throughout the network, as opposed to 802.1D where only the root did. This mechanism is much faster than the 802.1D equivalent. There is no need to wait for the root bridge to be notified and then maintain the topology change state for the whole network for <max age plus forward delay> seconds.



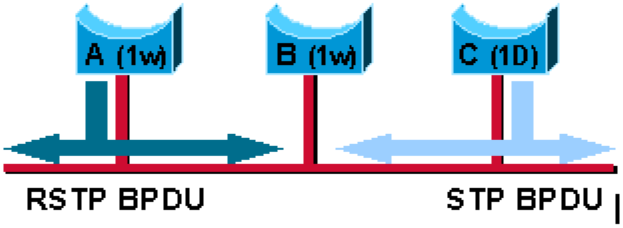
The originator of the TC directly floods this information through the network

In just a few seconds, or a small multiple of hello-times, most of the entries in the CAM tables of the entire network (VLAN) flush. This approach results in potentially more temporary flooding, but on the other hand it clears potential stale information that prevents rapid connectivity restitution.

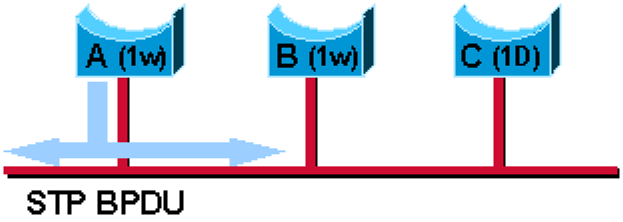
Compatibility with 802.1D

RSTP is able to interoperate with legacy STP protocols. However, it is important to note that the inherent fast convergence benefits of 802.1w are lost when it interacts with legacy bridges.

Each port maintains a variable that defines the protocol to run on the corresponding segment. A migration delay timer of three seconds also starts when the port comes up. When this timer runs, the current STP or RSTP mode associated to the port is locked. As soon as the migration delay expires, the port adapts to the mode that corresponds to the next BPDU it receives. If the port changes its mode of operation as a result of a BPDU received, the migration delay restarts. This limits the possible mode change frequency.



For instance, suppose Bridges A and B in the preceding figure both run RSTP, with Switch A designated for the segment. A legacy STP Bridge C is introduced on this link. As 802.1D bridges ignore RSTP BPDUs and drop them, C believes there are no other bridges on the segment and starts to send its inferior 802.1D-format BPDUs. Switch A receives these BPDUs and, after twice hello-time seconds maximum, changes its mode to 802.1D on that port only. As a result, C now understands the BPDUs of Switch A and accepts A as the designated bridge for that segment.



Notice in this particular case, if Bridge C is removed, Bridge A runs in STP mode on that port even though it is able to work more efficiently in RSTP mode with its unique neighbor B. This is because A does not know Bridge C is removed from the segment. For this particular (rare) case, user intervention is required in order to restart the protocol detection of the port manually.

When a port is in 802.1D compatibility mode, it is also able to handle topology change notification (TCN) BPDUs, and BPDUs with TC or TCA bit set.

Conclusion

RSTP (IEEE 802.1w) natively includes most of the Cisco proprietary enhancements to the 802.1D spanning tree, such as BackboneFast, UplinkFast, and PortFast. RSTP can achieve much faster convergence in a properly configured network, sometimes in the order of a few hundred milliseconds. Classic 802.1D timers, such as forward delay and max_age, are only used as a backup and should not be necessary if point-to-point links and edge ports are properly identified and set by the administrator. Also, the timers should not be necessary if there is no interaction with legacy bridges.

Cisco Support Community - Featured Conversations

Cisco Support Community is a forum for you to ask and answer questions, share suggestions, and collaborate with your peers. Below are just some of the most recent and relevant conversations happening right now.

Want to see more? Join us by clicking **here**

Spanning Tree Protocol 802.1D vs. 802.1w	**mike15400**	**2 Replies**	1 year, 4 months ago
Rapid Spanning Tree protocol on 7613	**admin**	**1 Reply**	6 years, 1 month ago
802.1d spanning tree vs rapid spanning...	**carl_townshend**	**5 Replies**	2 years, 1 day ago
Understanding Spanning-Tree Protocol	**joealbergo**	**6 Replies**	3 weeks, 5 days ago
Per-VLAN Spanning Tree Protocol vs...	**s-durando**	**1 Reply**	2 years, 1 month ago
Rapid spanning tree combnation	**sholeh**	**3 Replies**	4 years, 2 months ago
rapid spanning tree	**ward**	**2 Replies**	6 years, 11 months ago
rapid spanning tree / portfast	**lwallimann**	**2 Replies**	4 years, 2 months ago
Rapid Spanning Tree	**randolphcco**	**1 Reply**	4 years, 3 months ago
rapid spanning tree	**frey-telecom**	**1 Reply**	3 years, 4 months ago

Start A New Discussion **Subscribe**

Related Information

- **Configuring MST (802.1s)/RSTP (802.1w) on Catalyst Series Switches Running CatOS**
- **Understanding and Configuring the Cisco Uplink Fast Feature**
- **LAN Switching Technology Support**
- **LAN Product Support**
- **Tools and Resources**
- **Technical Support & Documentation - Cisco Systems**

Home | How to Buy | Login | Profile | Feedback | Site Map | Help

Contacts & Feedback | Help | Site Map