The Spanning Tree

802.1D (2004)
RSTP
MSTP

Problem Description

- We want redundant links in bridged networks
- But transparent bridging cannot deal with redundancy
  - Broadcast storms and other problems (see later)
- Solution: the spanning tree protocol
  - Allows for redundant paths
  - Ensures non-redundant active paths

Standard STP

A short repetition of why and how

Bridging Problems

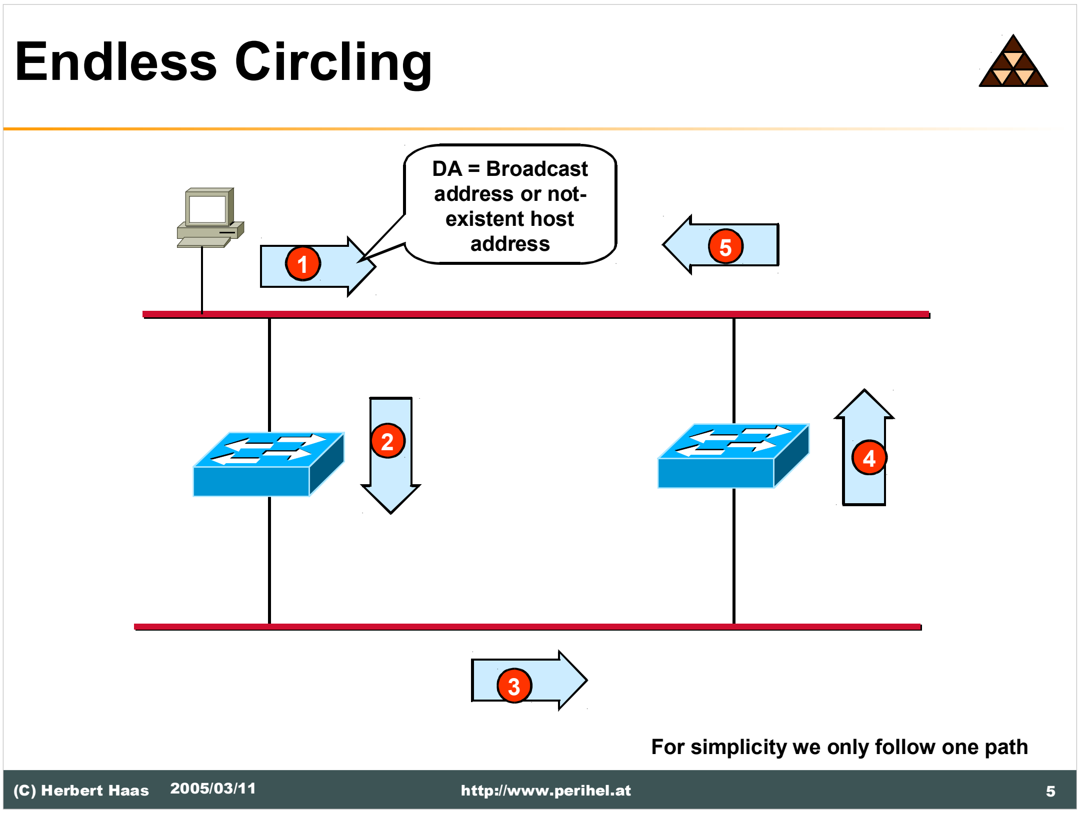
- Redundant paths lead to
  - Broadcast storms
  - Endless cycling
  - Continuous table rewriting
- No load sharing possible
- No ability to select best path

You might have noticed that bridges do not really learn the network topology. They only learn a simple destination to port association! Because of this there is no means to determine the best path, and furthermore frames might be caught in a loop.

Especially broadcast frames have no defined destination and would be forwarded over all parallel paths—endlessly! This results in endless circling of frames, or more dangerous, in a so-called "broadcast storm".

Also a continuous table rewriting might occur (this is not so widely known but also explained in the next pages).

Most people are not aware that frames might be stored up to 4 seconds inside the buffer of a switch—and it still complies to the IEEE standard. Although this would happen only in rare cases of congestion, transparent bridging is not suitable for hard realtime applications. Today the situation has changed, QoS features are included to assure bounded delays.

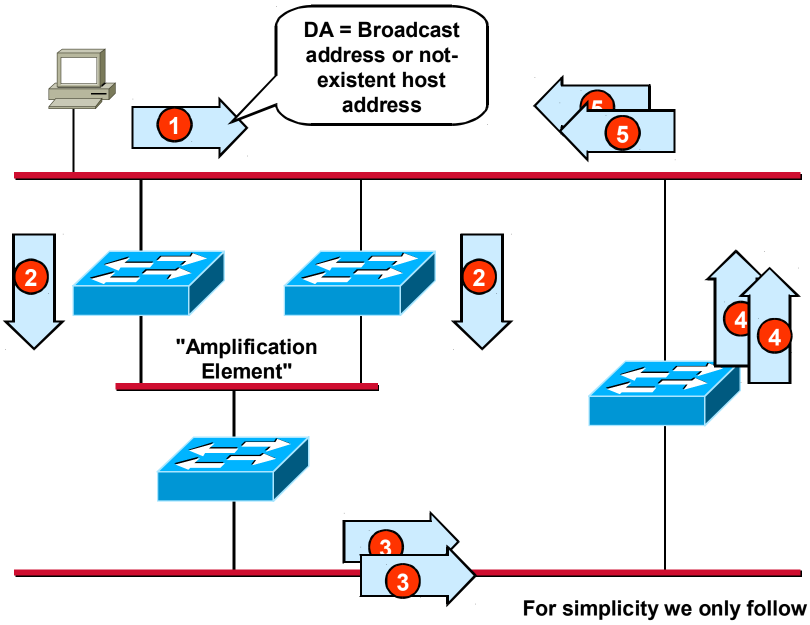


The picture above illustrates the endless circling phenomena. Assume a network with parallel paths between two LAN segments, realized by two bridges. Any frame with a broadcast destination address would be forwarded by both bridges to the other segment and back and forth and so on.

Obviously endless circling leads to congestion problems an is not desired. Remember that there is not hop count or time-to-live number within the Ethernet header.

But endless circling is not the main problem... (see next slide)

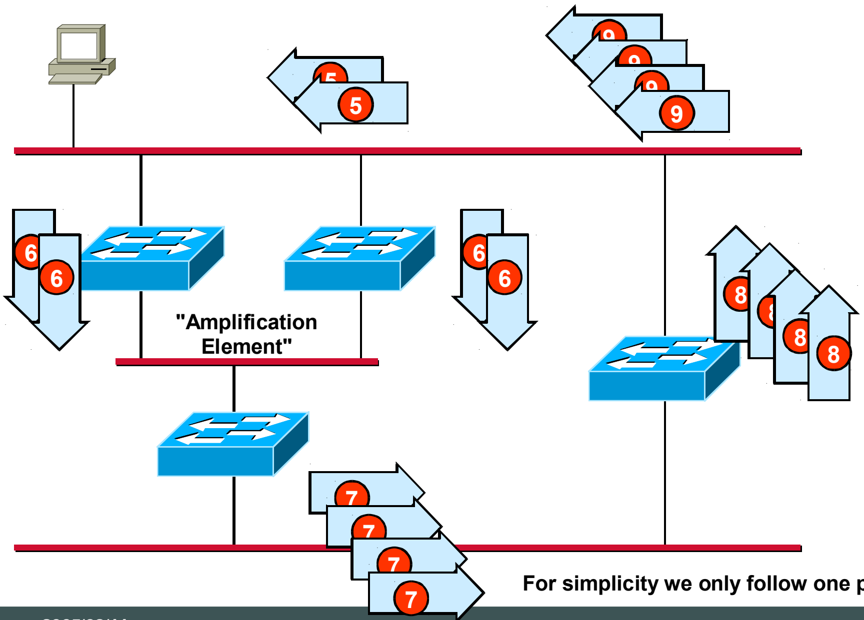
Broadcast Storm (1)



The most feared issue with bridging are broadcast storms. Broadcast storms can be considered as a dramatically "enhanced" endless circling problem. Broadcast storms appear when there is an "amplification" element within the network, such as those threefold parallel paths in the diagram above.

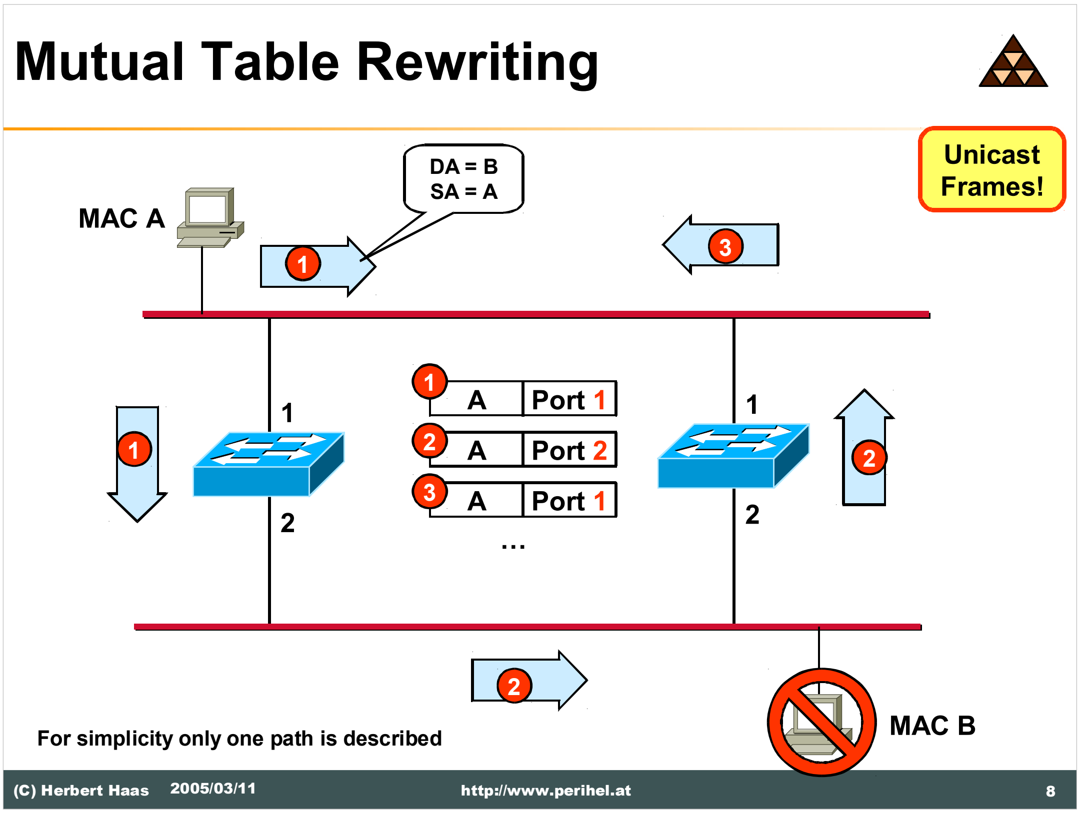
Within a very short time (e.g. 1 second) the whole LAN is overloaded with broadcast frames and nobody could transmit any useful frame anymore.

Broadcast Storm (2)



For simplicity we only follow one path

The picture above shows the amplification effect mentioned on the previous page.



A relatively seldom known problem is the mutual table rewriting phenomena. This problem occurs with unicast frames!

Assume that host A sends an unicast frame to destination B, both bridges learn the location of host A and host B, but suddenly B is detached. However, both bridges keep the entry for B for five minutes.

During this time the following happens:

1) After the bridges forward the frame from the above segment to the bottom segment this frame is not consumed by any host B, and therefore the bridges forward this frame back to the top segment.

2) At this moment the bridges rewrites their table as host A appears to be located on the bottom segment.

3) Again the bridge forward the frame to the bottom segment, hereby rewriting the port address for this source address...ad infinitum!

The Spanning Tree

IEEE 802.1D-2004

Spanning Tree

- Invented by *Radia Perlman* as general "mesh-to-tree" algorithm
- A must in bridged networks with redundant paths
- Only one purpose: Cut off redundant paths with highest costs
- Special STP frames: Bridge Protocol Data Units (BPDUs)

Now we have learned that active parallel paths lead to severe problems in a switched (i.e. bridged) network. Therefore we can only overcome this problem by deactivating any redundant path. This should be performed automatically in order to call Ethernet bridging still "Transparent" bridging.

The inventor of bridging, Radia Perlman, also created an easy solution for the redundancy problem: The Spanning Tree Protocol (STP).

The STP is implemented in bridges only (not in hosts) and has only one purpose: To determine any redundant paths and cut them off! Hereby cost values are considered for each path in order to maintain the best paths.

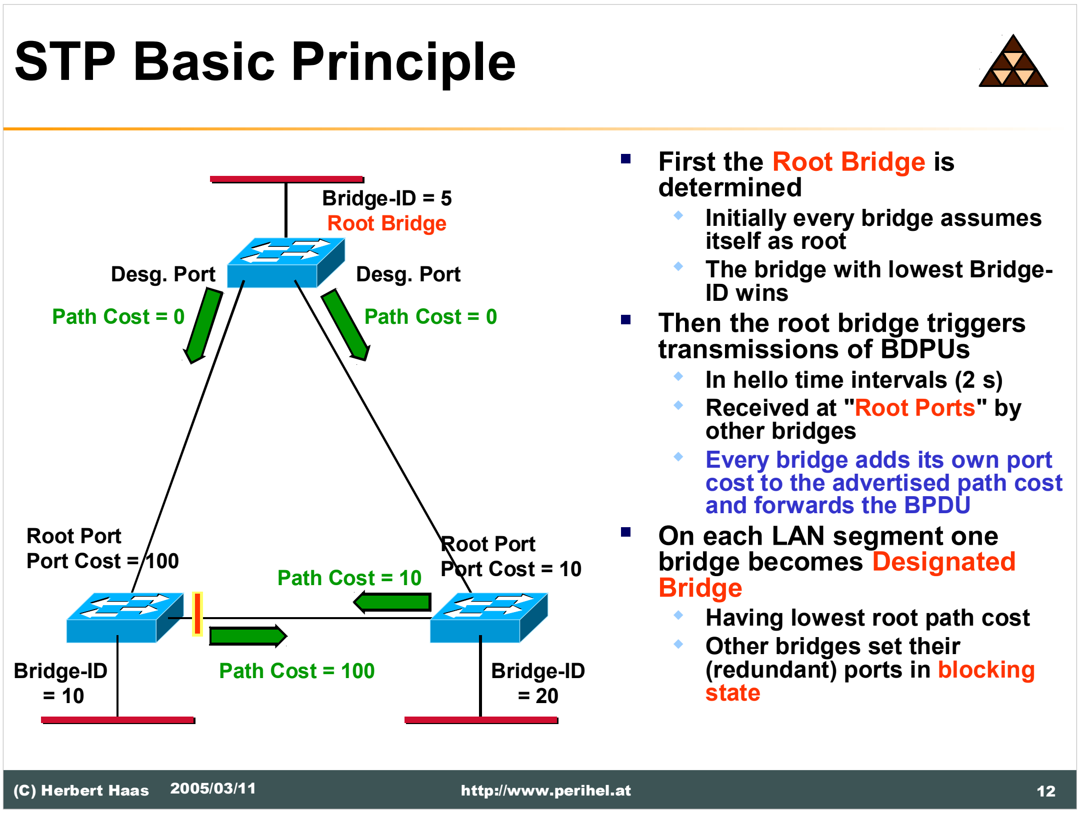
Three STP Parameters

- 8 byte **Bridge-ID** for each bridge
  - Consists of 2 byte Priority value (default 32768) and 6 byte (lowest) MAC address
  - Used to determine root bridge and as tie-breaker to when determing designated port
- 4 byte **Port Cost** for each port
  - Old (still used) standard method: 1000 / Port_BW_in_Mbits
    - E. g. 10 Mbit/s ➔ Cost=100
  - Used to calculate **Root Path Cost** to determine root port and designated port
- 2 byte **Port-ID** for each port
  - Consists of 1 byte Priority value (default 128) and 1 byte port number
  - Only used as tie-breaker if the same Bridge-ID and the same Path Cost is received on multiple ports

What do we need for STP to work? First of all this protocol needs a special messaging means, realized in so-called **Bridge Protocol Data Units (BPDUs).** BPDUs are simple messages contained in Ethernet frames containing several parameters described below.

Each bridge is assigned one unique **Bridge-ID** which is a combination of a 16 bit priority number and the lowest MAC address found on any port on this bridge. The Bridge-ID is determined automatically using the default priority 32768.

Each port is assigned a **Port Cost**. Again this value is determined automatically using the simple formula Port Cost = 1000 / BW, where BW is the bandwidth in Mbit/s. Of course the Port Cost can be configured manually.



We give only a basic explanation here of how the STP works. First a **Root Bridge** is determined by choosing the bridge with the **lowest** Bridge-ID. This is simply done by sending BDUs containing the presumed Root Bridge. At first each bridge assumes to be the Root Bridge itself. After any bridge has sent his "opinion" the root bridge is determined.

Then the **Root Ports** are determined by each bridge. The Root Bridge sends BPDUs periodically (every 2 seconds by default) "downstream" to the "leaves" of the tree which is currently created. Each bridge adds its own port costs to the Root Path Cost parameter in the BPDU and forwards this BPDU over all other ports. This way each bridge learns the best path to the root.

Finally on each LAN segment the bridge having best Root Port becomes **Designated Bridge**. Its port on this LAN segment is called Designated Port (DP). Root Ports and Designated Ports are in a forwarding state. All other ports are in a blocking state.

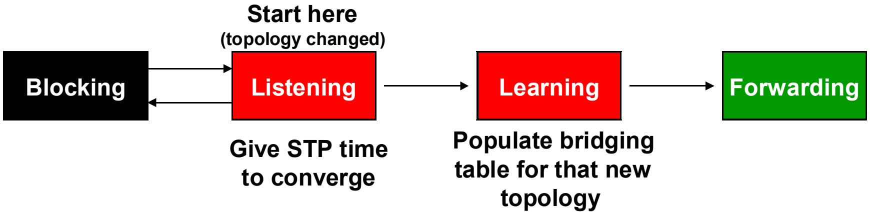
But the best (and shortest) description comes from Radia Perlman's poem:

First the root must be selected
by ID it is elected.
least cost paths to root are traced,
and in the tree these paths are place.

Final situation

- Root switch
  - Has only Designated Ports
  - All in forwarding state
- Other switches have
  - Exactly one Root Port (upstream)
  - Zero or more Designated Ports (downstream)
  - Zero or more Nondesignated Ports (blocked)

Port States



- At each time, a port is in one of the following states:
  - Blocking, Listening, Learning, Forwarding, or Disabled
- Only Blocking or Forwarding are final states (for enabled ports)
- Transition states
  - 15 s Listening state is used to converge STP
  - 15 s Learning state is used to learn MAC addresses for the new topology
- Therefore it lasts 30 seconds until a port is placed in forwarding state

Note

- **Redundant links remain in active stand-by mode**
  - **If root port fails, other root port becomes active**
- **Only 7 bridges per path allowed according standard (!)**
  - **Because of 15 seconds listening state and 2 seconds hello timers**

Still it is reasonable to establish parallel paths in a switched network in order to utilize this redundancy in an event of failure. The STP automatically activates redundant paths if the active path is broken. Note that BPDUs are always sent or received on blocking ports.

Note that (very-) low price switches might not support the STP and should not be used in high performance and redundant condigurations.

For performance reasons the IEEE standard 802.1d only allows 7 bridges for each path. Some vendors allow to change this value.

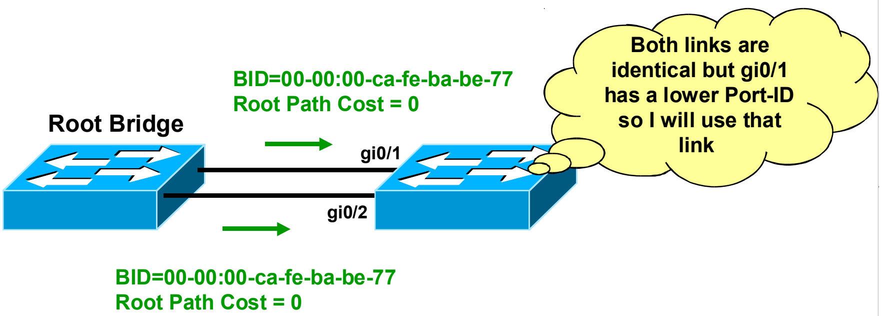
Only for your interest, here are the Ethernet parameters for BPDUs:

Multicast address 0180 C200 0000 hex

LLC DSAP=SSAP= 42 hex

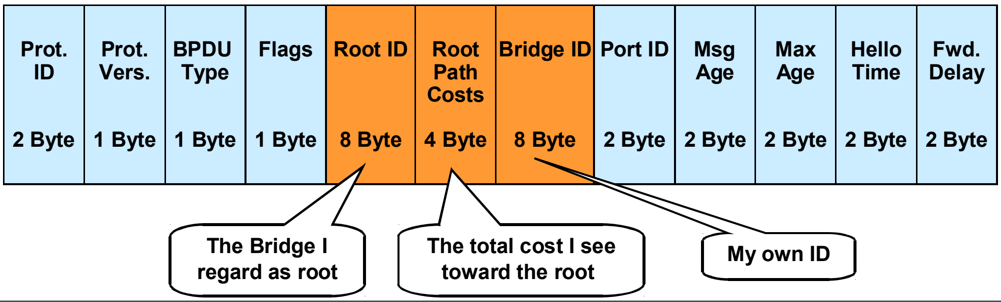
Usage for a Port-ID

- The Port-ID is only used as last tie-breaker
- Typical situation in highly redundant topologies: Multiple links between each two switches
  - Same BID and Costs announced on each link
  - Only local Port-ID can choose a single link



BPDU Format

- **Each bridge sends periodically BPDUs carried in Ethernet multicast frames**
  - **Hello time default: 2 seconds**
- **Contains all information necessary for building Spanning Tree**



Just for your interest, the above picture shows the structure of BPDUs. You see, there is no magic in here, and the protocol is very simple. There are no complicated protocol procedures. BPDUs are sent periodically and contain all involved parameters. Each bridge enters its own "opinion" there or adds its root path costs to the appropriate field. Note that some parameters are transient and others are not.

The other parameters not explained here are not so important to understand the basic principle.

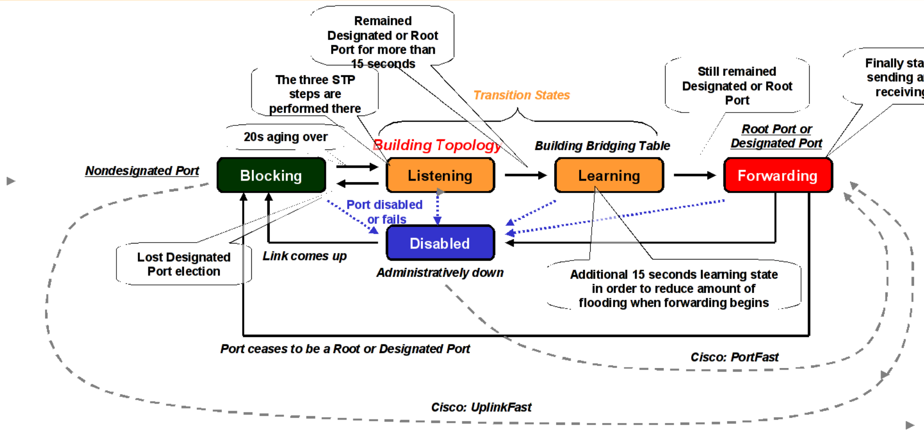
Importance of details…

- Many people think STP is a simple thing – until they encounter practical problems in real networks
- Important Details
  - STP State Machine
  - BPDU format details
  - TCN mechanism
  - RSTP
  - MSTP

Note: STP is a port-based algorithm

- Only the root-bridge election is done on the bridge-level
- All other processing is port-based
  - To establish the spanning tree, each enabled port is either forwarding or blocking
  - Additionally two transition states have been defined

STP State Machine: Port Transition Rules



- STP is completely performed in the Listening state
  - Blocking ports still receive BPDUs (but don't send)
- Default convergence time is 30-50 s
  - 20s aging, (15+15)s transition time
- Timer tuning: Better don't do it !
  - Only modify timers of the root bridge
  - Don't forget values on supposed backup root bridge

802.1d defines port roles and states:

Port Roles	Port States
Root	Disabled
Designated	Blocking
Nondesignated	Listening
	Learning
	Forwarding

A specific port role is a long-term "destiny" for a port, while port states denote transient situations. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration.

From the 802.1D-1998 standard:

If the Bridge times out the information held for a Port, it will attempt to become the Designated Bridge for the LAN to which that Port is attached, and will transmit protocol information received from the Root on its Root Port on to that LAN.

If the Root Port of the Bridge is timed out, then another Port may be selected as the Root Port. The information transmitted on LANs for which the Bridge is the Designated Bridge will then be calculated on the basis of information received on the new Root Port.

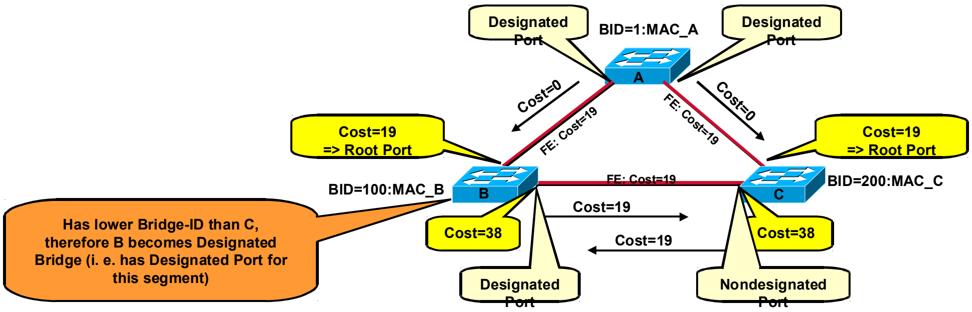
Another Example

Three steps to create spanning tree:

1. **Elect Root Bridge** (Each L2-network has exactly one Root Bridge)
2. **Elect Root Ports** (Each non-root bridge has exactly one Root Port)
3. **Elect Designated Ports** (Each segment has exactly one Designated Port)

To determine root port and designated port:

1. **Determine lowest (cumulative) Path Cost to Root Bridge**
2. **Determine lowest Bridge ID**
3. **Determine lowest Port ID**



Each segment has exactly one Designated Port. This simple rule actually breaks any loops.

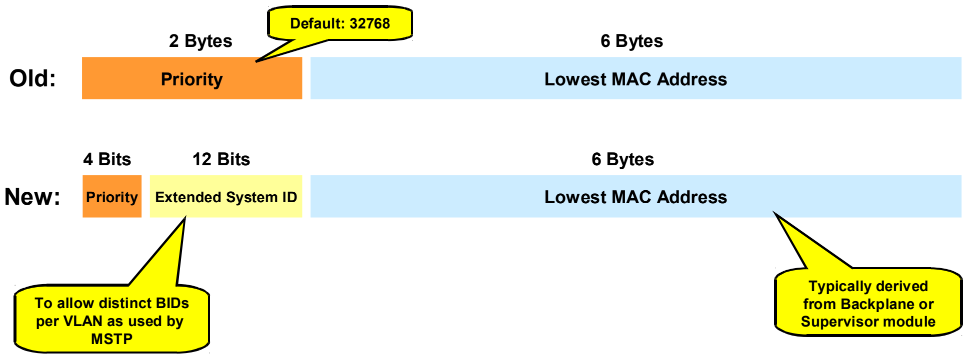
A nondesignated port receives a more useful BPDU than the one it would send out on its segment. Therefore it remains in the so-called blocking state.

Port ID - Contains a unique value for every port. Port 1/1 contains the value 0x8001, whereas Port 1/2 contains 0x8002. (Or in decimal: 128.1, 128.2, …)

From the 802.1D-1998 standard:

Each Configuration BPDU contains, among other parameters, the unique identifier of the Bridge that the transmitting Bridge believes to be the Root, the cost of the path to the Root from the transmitting Port, the identifier of the transmitting Bridge, and the identifier of the transmitting Port. This information is sufficient to allow a receiving Bridge to determine whether the transmitting Port has a better claim to be the Designated Port on the LAN on which the Configuration BPDU was received than the Port currently believed to be the Designated Port, and to determine whether the receiving Port should become the Root Port for the Bridge if it is not already.

Components of the Bridge-ID



- **The recent 802.1D-2004 standard requires only 4-bits for priority and 12 bits to distinguish multiple STP instances**
  - **Typically used for MSTP, where each set of VLANs has its own STP topology**
- **Therefore, ascending priority values are 0, 4096, 8192, …**
  - **Typically still configured as 0, 1, 2, 3 …**

802.1T spanning-tree extensions, and some of the bits previously used for the switch priority are now used for the extended system ID (VLAN identifier for the per-VLAN spanning-tree plus [PVST+] and for rapid PVST+ or an instance identifier for the multiple spanning tree [MST]).

Before this, spanning tree used one MAC address per VLAN to make the bridge ID unique for each VLAN.

Extended system IDs are VLAN IDs between 1025 and 4096. Releases 12.1(14)E1and later releases support a 12-bit extended system ID field as part of the bridge ID.

```
Switch(config)# spanning-tree extend system-id
```

STP Port Cost

Speed [Mbit/s]	Old Cost (1000/Speed)	New Cost	802.1T
10	100	100	2,000,000
100	10	19	200,000
155	6	14	(129032 ?)
622	1	6	(32154 ?)
1000	1	4	20,000
10000	1	2	2,000

- **Also different cost values might be used**
  - **See recommendations in the IEEE 802.1D-2004 standard to comply with RSTP and MSTP**

802.1T Excerpt

Link Speed	Recommended value	Recommended range	Range
<=100 Kb/s	200 000 000*	20 000 000–200 000 000	1–200 000 000
1 Mb/s	20 000 000[a]	2 000 000–200 000 000	1–200 000 000
10 Mb/s	2 000 000[a]	200 000–20 000 000	1–200 000 000
100 Mb/s	200 000[a]	20 000–2 000 000	1–200 000 000
1 Gb/s	20 000	2 000–200 000	1–200 000 000
10 Gb/s	2 000	200–20 000	1–200 000 000
100 Gb/s	200	20–2 000	1–200 000 000
1 Tb/s	20	2–200	1–200 000 000
10 Tb/s	2	1–20	1–200 000 000

*Bridges conformant to IEEE Std 802.1D, 1998 Edition, i.e., that support only 16-bit values for Path Cost, should use 65 535 as the Path Cost for these link speeds when used in conjunction with Bridges that support 32-bit Path Cost values.

Detailed BPDU Format

	Bytes	
Protocol ID	2	Always zero
Version	1	Always zero
Message Type	1	Configuration (0x00) or TCN BPDU (0x80)
Flags	1	LSB = Topology change flag (TC), MSB = TC Ack flag (TCA)
Root ID	8	Who is Root Bridge?
Root Path Cost	4	How far away is Root Bridge?
Bridge ID	8	ID of bridge that sent this BPDU
Port ID	2	Port-ID of sending bridge (unique: Port1/1=0x8001, 1/2=0x8002, ...)
Message Age	2	Time since Root generated this BPDU
Maximum Age = 20	2	BPDU is discarded if older than this value (default: 20 seconds)
Hello Time = 2	2	Broadcast interval of BPDUs (default: 2 seconds)
Forward Delay = 15	2	Time spent in learning and listening states (default: 15 seconds)

A TCN-BPDU only consists of these 3 fields !!! (Protocol ID, Version, Message Type)

When first booted, Root-ID == BID

If value increases, then the originating bridge lost connectivity to Root Bridge (Message Age)

Maximum Age, Hello Time, Forward Delay:
- Predetermined by root bridge
- Affect convergence time
- Misconfigurations cause loops

- BPDUs are sent in 802.3 frames
  - DA = 01-80-C2-00-00-00
  - LLC has DSAP=SSAP = 0x42 ("the answer")
- Configuration BPDUs
  - Originated by Root Bridge periodically (2 sec Hello Time), flow downstream

In normal stable operation, the regular transmission of Configuration Messages by the Root ensures that topology information is not timed out. To allow for reconfiguration of the Bridged LAN when components are removed or when management changes are made to parameters determining the topology, the topology information propagated throughout the Bridged LAN has a limited lifetime. This is effected by transmitting the age of the information conveyed (the time elapsed since the Configuration Message originated from the Root) in each Configuration BPDU. Every Bridge stores the information from the Designated Port on each of the LANs to which its Ports are connected, and monitors the age of that information.

Topology Change Notification (TCN)

- Special BPDUs, used as alert by any bridge
  - Flow upstream (through Root Port)
  - Only consists of the first three standard header fields!
- Sent upon
  - Transition of a port into Forwarding state and at least one Designated Port exists
  - Transition of a port into Blocking state (from either Forwarding or Learning state)
- Sent until acknowledged by TC Acknowledge (TCA)

Topology Change Notification (TCN)

- **Only the Designated Ports of upstream bridges processes TCN-BPDUs and send TC-Ack (TCA) downstream**
- **Finally the Root Bridge receives the TC and sends Configuration BPDUs with the TC flag set to 1 (=TCA) downstream for (Forward Delay + Max Age = 35) seconds**
  - **This instructs all bridges to reduce the default bridging table aging (300 s) to the current Forward Delay value (15 s)**
  - **Thus bridging tables can adapt to the new topology**

Main idea: To avoid 5 minute age timer upon topology change! Some destinations may not be reachable any more!

Normally, all Configuration BPDUs are (periodically) sent by the root bridge. Other bridges never send out a BPDU toward the root bridge!

Therefore dedicated TCN messages have been defined to allow a non-root bridge to announce topology changes.

TCN BPDUs are sent on the root port until acknowledged by the upstream bridge (BPDU with the topology change acknowledgement (TCA) bit set).

The TCN is sent every hello_time which is a locally configured value (not the hello_time specified in configuration BPDUs)

Reasons to send TCNs:

1. When a port changes from "Forwarding" to any other state
2. When a port transitions to forwarding and the bridge has a designated port (that is the bridge is not standalone).

Then a TCN is sent upstream to the root bridge (i. e. only sent through the root port) which 'broadcasts' this information downstream to all other bridges.

o These downstream TCNs are not acknowledged

o The TC bit is set by the root for a period of max_age + forward_delay seconds, which is 20+15=35 seconds by default.

o Every bridge now reduces the aging time of every existing bridging table entry to 15 seconds (more precisely: the actual value of forward_delay) This is done (also for new entries) for the duration of 35 seconds (more precisely: max_age + forward_delay).

Configuration on Cisco switches

```
Switch(config)# spanning-tree vlan 200
```

Enable SPT on a specific VLAN

```
Switch(config)# spanning-tree vlan 200 priority 0
```

Enforcing Root Bridge

```
Switch(config-if)# spanning-tree cost 18
```

Manipulate Port Costs

```
Switch(config-if)# spanning-tree vlan 200 cost 15
```

Manipulate Port Costs for a specific VLAN

```
Switch# show spanning-tree vlan 200

VLAN0200
  Spanning tree enabled protocol ieee
  Root ID     Priority     49352
              Address      0008.2199.2bc0
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority     49352  (priority 49152 sys-id-ext 200)
              Address      0008.2199.2bc0
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300
  Uplinkfast enabled

Interface        Port ID                      Designated                 Port ID
Name             Prio.Nbr      Cost Sts       Cost Bridge ID             Prio.Nbr
---------------- -------- --------- --- --------- -------------------- --------
Fa0/1            128.1         3019 LIS         0 49352 0008.2199.2bc0 128.1
Fa0/2            128.2         3019 LIS         0 49352 0008.2199.2bc0 128.2
```


Enable spanning tree on a per-VLAN basis.

Old commands:

set spantree priority

set spantree root

show spantree

STP Optimizations

Port Fast
Uplink Fast
Backbone Fast

Port Fast

- Optimizes switch ports connected to end-station devices
  - Usually, if PC boots, NIC establishes L2-link, and switch port goes from Disabled=>Blocking=>Listening=>Learning=>Forwarding state ...30 seconds!!!
- Port Fast allows a port to immediately enter the Forwarding state
  - STP is NOT disabled on that port!

Any connectivity problems after cold booting a PC in the morning but NOT after warm-booting during the day?

Port Fast

- Port Fast only works once after link comes up!
  - If port is then forced into Blocking state and later returns into Forwarding state, then the normal transition takes place!
  - Ignored on trunk ports
- Alternatives:
  - Disable STP (often a bad idea)
  - Use a hub in between => switch port is always active

PortFast Configuration

```
Switch(config-if)# spanning-tree portfast
```

Enables PortFast on an interface

```
Switch#show running-config interface fastethernet 5/8
Building configuration...
Current configuration:
!
interface FastEthernet5/8
 no ip address
 switchport
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast
end
```

Verify PortFast

STP Optimizations

Port Fast
Uplink Fast
Backbone Fast

Uplink Fast

- Accelerates STP to converge within 1-3 seconds
  - Cisco patent
  - Marks some blocking ports as backup uplink
- Typically used on access layer switches
  - Only works on non-root bridges
  - Requires some blocked ports
  - Enabled for entire switch (and not for individual VLANs)

UplinkFast is actually a root port optimization.

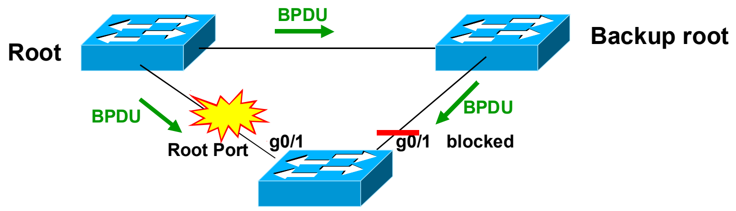
The standard Cisco mcast address 01-00-0C-CC-CC-CC, which is used for CDP, VTP, DTP, and DISL cannot be used, because all Cisco devices are programmed to not flood these frames (rather consume it).

Note that only MACs not learned over the uplinks are flooded.

show spantree uplinkfast

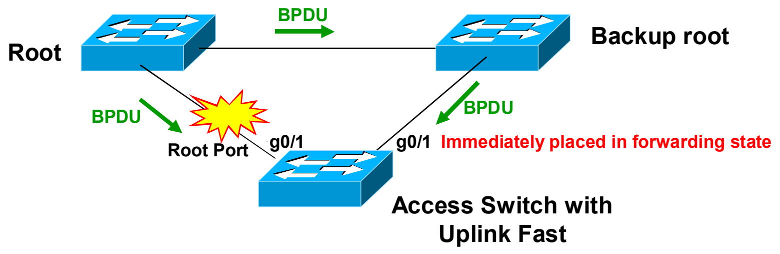
Problem

- **When link to root bridge fails, STP requires (at least) 30 seconds until alternate root port becomes active**



Idea of Uplink Fast

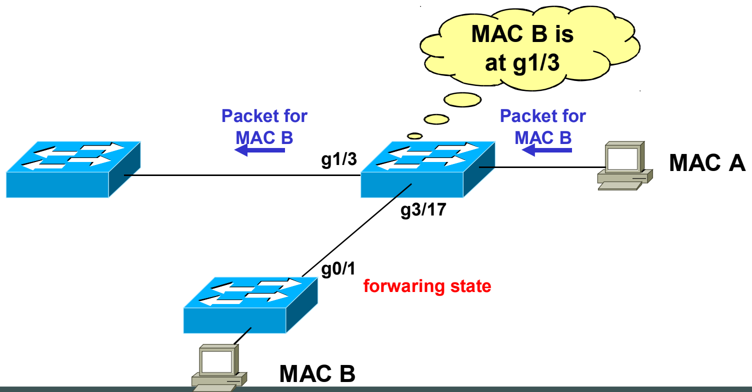
- When a port receives a BPDU, we know that it has a path to the root bridge
  - Put all root port candidates to a so-called "Uplink Group"
- Upon uplink failure, immediately put best port of Uplink group into forwarding state
  - There cannot be a loop because previous uplink is still down



The UplinkFast feature is based on the definition of an uplink group. On a given switch, the uplink group consists in the root port and all the ports that provide an alternate connection to the root bridge. If the root port fails, which means if the primary uplink fails, a port with next lowest cost from the uplink group is selected to immediately replace it.

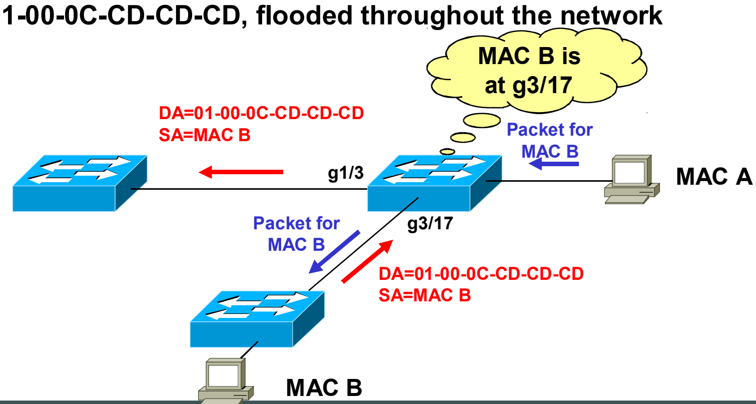
Incorrect Bridging Tables

- But upstream bridges still require 30 s to learn new topology
- Bridging table entries in upstream bridges may be incorrect



Actively correct tables

- Uplink Fast corrects the bridging tables of upstream bridges
- Sends 15 multicast frames (one every 100 ms) for each MAC address in its bridging table (i. e. for each downstream hosts)
  - Using SA=MAC: All other bridges quickly reconfigure their tables; dead links are no longer used
  - DA=01-00-0C-CD-CD-CD, flooded throughout the network



Addional Details

- When broken link becomes up again, Uplink Fast waits until traffic is seen
  - That is, 30 seconds plus 5 seconds to support other protocols to converge (e. g. Etherchannel, DTP, …)
- Flapping links would trigger uplink fast too often which causes too much additional traffic
  - Therefore the port is "hold down" for another 35 seconds before Uplink Fast mechanism is available for that port again
- Several STP parameters are modified automatically
  - Bridge Priority = 49152 (don't want to be root)
  - All Port Costs += 3000 (don't want to be designated port)

1100xxxx xxxxxxxx = 49152=2^15+2^14

UplinkFast - Configuration

```
Switch(config)# spanning-tree uplinkfast [max-update-rate max_update_rate]
```

```
Switch# show spanning-tree uplinkfast
UplinkFast is enabled
Station update rate set to 150 packets/sec.
UplinkFast statistics
-----------------------
Number of transitions via uplinkFast (all VLANs)            :9
Number of proxy multicast addresses transmitted (all VLANs) :5308
Name                 Interface List
-------------------- ------------------------------------
VLAN1                Fa6/9(fwd), Gi5/7
VLAN2                Gi5/7(fwd)
VLAN3                Gi5/7(fwd)
VLAN4
VLAN5
VLAN1002             Gi5/7(fwd)
VLAN1003             Gi5/7(fwd)
VLAN1004             Gi5/7(fwd)
VLAN1005             Gi5/7(fwd)
```

STP Optimizations

Port Fast
Uplink Fast
Backbone Fast

Backbone Fast

- **Complementary to Uplink Fast**
- **Safes 20 seconds when recovering from <u>indirect link failures</u> in core area**
  - **Issues Max Age timer expiration**
  - **Reduce failover performance from 50 to 30 seconds**
  - **Cannot eliminate Forwarding Delay**
- **Should be enabled on every switch!**

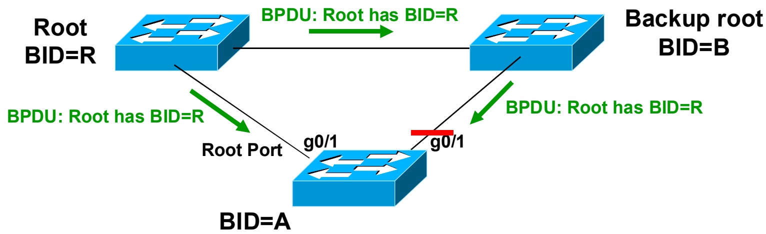
BackboneFast is actually a Max Age optimization.

Upon Root Port failure, a switch assumes it Root role and generates own Configuration BPDUs, which are treated as "inferior" BPDUs, because most switches might still receive the BPDUs from the original Root Bridge.

The request/response mechanism involves a so-called Root Link Query (RLQ) protocol, that is, RLQ-requests are sent to upstream bridges to check whether their connection to the Root Bridge is stable. Upstream bridges reply with RLQ-responses. If the upstream bridge does not know about any problems, it forwards the RLQ-request further upwards, until the problem is solved. If the RLQ-response is received by the downstream bridge on a non-Root Port, then this bridge knows, that it has lost its connection to the Root Bridge and can immediately expire the Max Age timer.

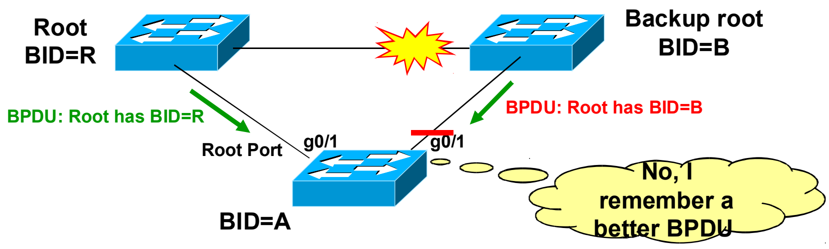
Problem

- Consider initial situation
- Note that blocked port (g0/1) always remembers "best seen" BPDU – which has best (=lowest) Root-BID



Problem (cont.)

- Now backup-root bridge looses connectivity to root bridge and assumes root role
- Port g0/1 does not see the BPDUs from the original root bridge any more
- But for MaxAge=20 seconds, any inferior BPDU is ignored



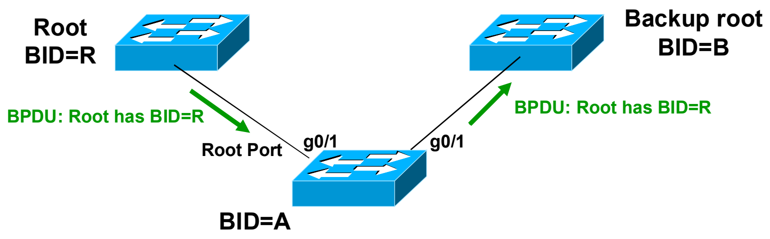
<u>**Note that the key problem is this:**</u>

1) **Direct** link failures **would immediately set the bridge in listening mode** (i. e. all of its ports).

2) But **indirect** link failures **always includes the max-age timer** (20 s) before entering the listening state.

Problem (cont.)

- Only after 20 seconds port g0/1 enters listening state again
- Finally, bridge A unblocks g0/1 and forwards the better BPDUs to bridge B
- Total process lasts 20+15+15 seconds



Solution

- If an inferior BPDU is originated from the local segment's Designated Bridge, then this probably indicates an indirect failure
  - (Bridge B was Designated Bridge in our example)
- To be sure, we ask other Designated Bridges (over our ***other*** blocked ports and the root port) what they think which bridge the root is
  - Using Root Link Query (RLQ) BPDU
- If at least one reply contains the "old" root bridge, we know that an indirect link failure occurred
  - Immediately expire Max Age timer and enter Listening state

BackboneFast - Configuration

```
Switch(config)# spanning-tree backbonefast
```

```
Switch# show spanning-tree backbonefast
BackboneFast is enabled

BackboneFast statistics
-----------------------
Number of transition via backboneFast (all VLANs)  : 0
Number of inferior BPDUs received (all VLANs)      : 0
Number of RLQ request PDUs received (all VLANs)    : 0
Number of RLQ response PDUs received (all VLANs)   : 0
Number of RLQ request PDUs sent (all VLANs)        : 0
Number of RLQ response PDUs sent (all VLANs)       : 0
```

Other STP Tuning Options

- BPDU Guard
  - Shuts down PortFast-configured interfaces that receive BPDUs, preventing a potential bridging loop
- Root Guard
  - Forces an interface to become a designated port to prevent surrounding switches from becoming the root switch
- BPDU Filter
- BPDU Skew Detection
  - Report late BPDUs via Syslog
  - Indicate STP stability issues, usually due to CPU problems
- Unidirectional Link Detection (UDLD)
  - Detects and shuts down unidirectional links
- Loop Guard

Rapid Spanning Tree (RSTP)

IEEE 802.1D – 2004
(Formerly known as 802.1w)

Introduction

- RSTP is now an add-on to the IEEE 802.1D-2004 standard
  - Contains contributions from Cisco
- Computation of the Spanning Tree is identical between STP and RSTP
  - Conf-BPDU and TCN-BPDU still remain
  - New BPDU type "RSTP" has been added
    - Version=2, type=2
- RSTP BPDUs can be used to negotiate port roles on a particular link
  - Only done if neighbor bridge supports RSTP (otherwise only Conf-BPDUs are sent
  - Using a Proposal/Agreement handshake

Major Features

- BPDUs are no longer triggered by root bridge
  - Instead, each bridge can generate BPDUs independently and immediately (on-demand)
- Much faster convergence
  - Few seconds
- Better scalability
  - No network diameter limit

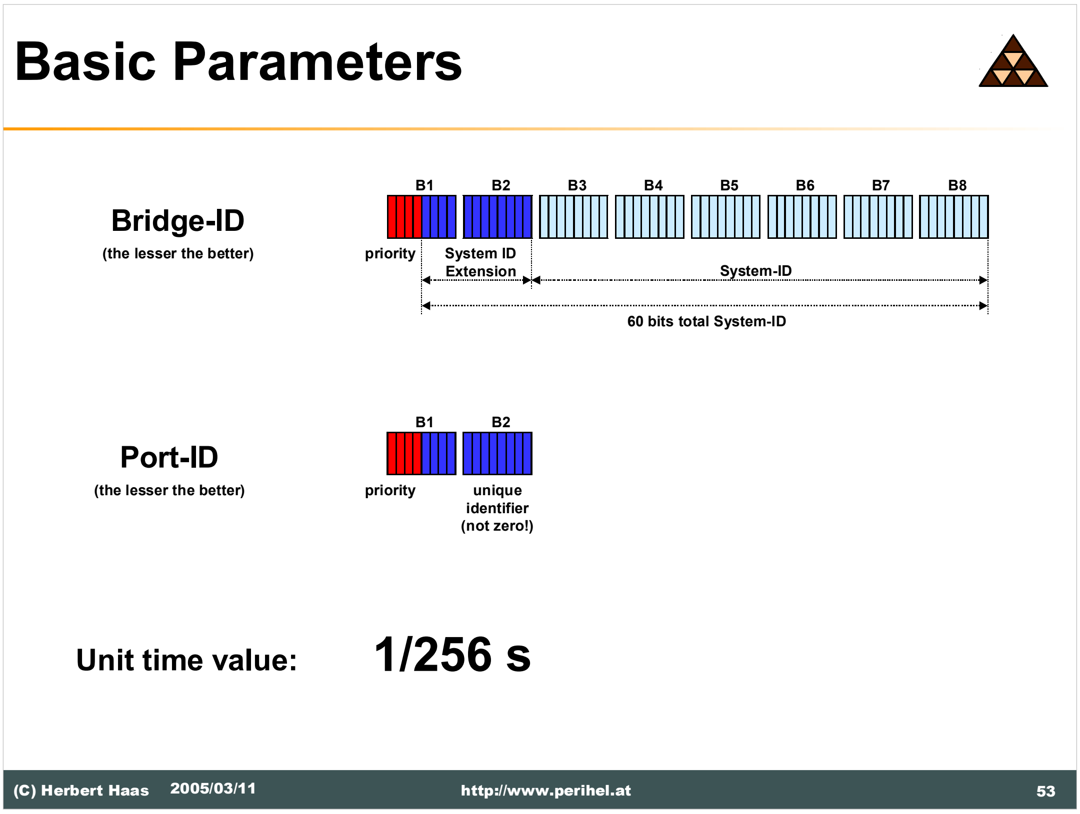
Compatibility

- RSTP is designed to be compatible and interoperable with the traditional STP – without additional management requirements!
- If an RSTP-enabled bridge is connected to an STP bridge, only Configuration-BPDUs and Topology-Change BPDUs are sent
  - (No port role negotiation)
- Memory requirements per bridge port independent of number of bridges

An RSTP Bridge Port automatically adjusts to provide interoperability, if it is attached to the same LAN as an STP Bridge. Protocol operation on other ports is unchanged. Configuration and Topology Change Notification BPDUs are transmitted instead of RST BPDUs which are not recognized by STP Bridges. Port state transition timer values are increased to ensure that temporary loops are not created through the STP Bridge. Topology changes are propagated for longer to support the different Filtering

Database flushing paradigm used by STP. It is possible that RSTP's rapid state transitions will increase rates of frame duplication and misordering.

BPDUs convey Configuration and Topology Change Notification (TCN) Messages. A Configuration Message can be encoded and transmitted as a Configuration BPDU or as an RST BPDU. A TCN Message can be encoded as a TCN BPDU or as an RST BPDU with the TC flag set. The Port Protocol Migration state machine determines the BPDU types used.



Bridge-ID:

12-bit System-ID Extension allows to have a different BID for every VLAN (MST, 802.1Q). For backwards compatibility, old STP implementations could use a 16-bit priority value but may only set the 4 most significant bits, remaining 12 must be zero:

MSByte1 2 3 ...

MSB LSB

xxxx 0000 0000 0000

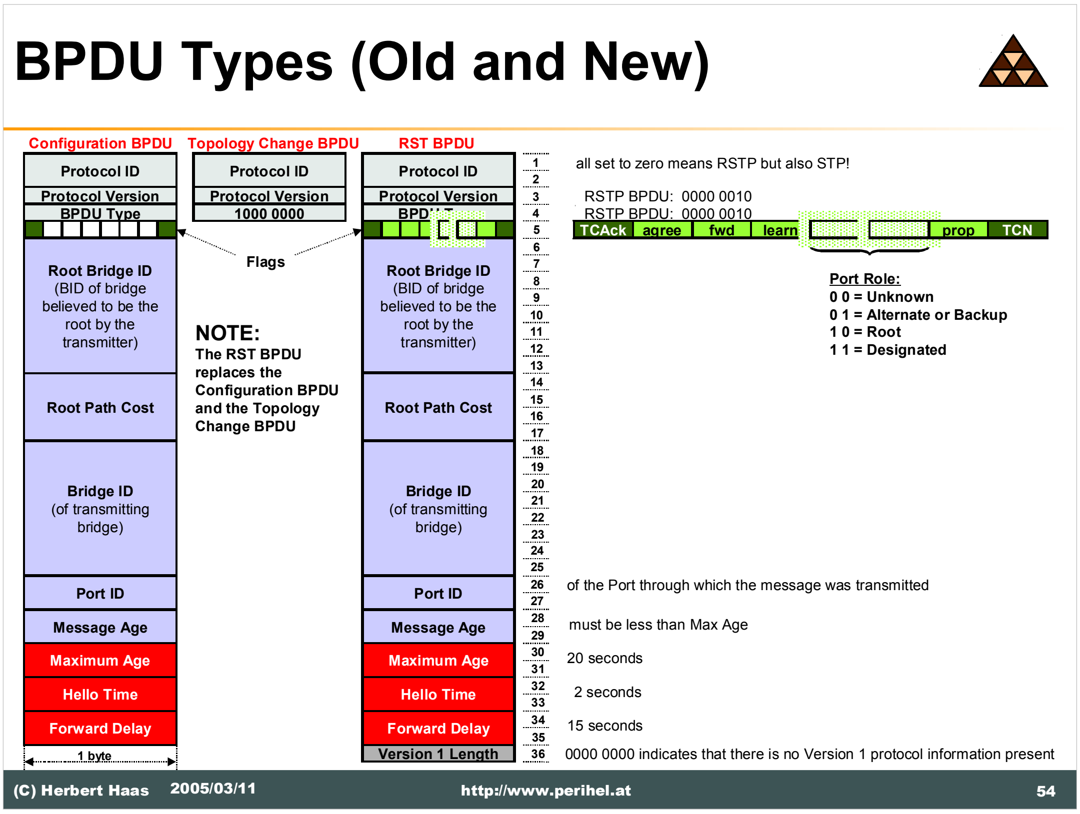
Allowed values: 0, 4096, 8192, ... , 61440, but I think the little Endian interpretation 0..15 will be used(?)

Port-ID:

In the old standard 8 bits priority + 8 bit unique identifier were used.

Unit time value

for all timer values (2 bytes) is 1/256 second, which allows a range from 0 to 65535*1/256=256.



Flags:

- TCN (bit 1)
- Proposal (bit 2)
- Port Role (bits 3, 4)
- Learning (bit 5)
- Forwarding (bit 6)
- Agreement (bit 7)
- Topology Change Acknowledgment (bit 8)

Note: A Configuration BPDU has same structure than a RSTP BPDU with the following exceptions:

1) A Configuration BPDU is only 35 byte long, that is, there is no "Version 1 length" field

2) A Configuration BPDU only uses two flags, that is, TCAck (bit 7) and TCN (bit 0)

NOTE: If the Unknown value of the Port Role parameter is received, the state machines will effectively treat the RST

BPDU as if it were a Configuration BPDU.

Same simple basic rules

- Bridge with lowest BID becomes **Root Bridge**
  - Has only Designated Ports
- Every other bridge has exactly one **Root Port**
  - Providing a least cost path to the Root Bridge
  - Local tie-breaker is the Port Identifier
- A **Designated Bridge** provides the lowest Root Path Cost for a LAN
  - Tie-breaker between multiple bridges is BID
  - Local tie-breaker is the Port Identifier

Every Bridge has a Root Path Cost associated with it. For the Root Bridge this is zero. For all other Bridges, it is the sum of the Port Path Costs on the least cost path to the Root Bridge.

If a Bridge has two or more ports with the same Root Path Cost, then the port with the best Port Identifier is selected as the Root Port.

The Bridge providing the lowest Root Path Cost for a LAN is called the Designated Bridge for that LAN. If there are two or more Bridges with the same Root Path Cost, then the Bridge with the best priority (least numerical value) is selected as the Designated Bridge.

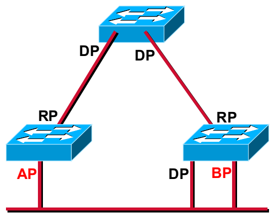
Since each Bridge provides connectivity between its Root Port and its Designated Ports, the resulting active topology connects all LANs (is “spanning”) and will be loop free (is a “tree”).

Any operational Bridge Port that is not a Root or Designated Port is a Backup Port if that Bridge is the Designated Bridge for the attached LAN, and an Alternate Port otherwise. Backup Ports exist only where there are two or more connections from a given Bridge to a given LAN.

Backup and Alternate Ports

- If a port is neither Root Port nor Designated Port
  - It is a **Backup Port** – if this bridge is a Designated Bridge for that LAN
  - Or an **Alternate Port** otherwise

Backup and Alternate Ports:



Port Types

- Shared Ports
  - Are not supported (ambiguous negotiations)
  - Uses standard STP here
- Point-to-point ports
  - Usual and required port types
  - Supports proposal-agreement process
- Edge Port
  - Hosts resides here
  - Transitions directly to the Forwarding Port State, since there is no possibility of it participating in a loop
  - May change their role as soon as a BPDU is seen

Algorithm Overview

- **Designated Ports transmit Configuration BPDUs periodically to detect and repair failures**
  - **Blocking (aka Discarding) ports send Conf-BPDUs only upon topology change**
- **Every Bridge accepts "better" BPDUs from any Bridge on a LAN or revised information from the prior Designated Bridge for that LAN**
- **To ensure that old information does not endlessly circulate through redundant paths in the network and prevent propagation of new information, each Configuration Message includes a message age and a maximum age**
- **Transitions to Forwarding is now confirmed by downstream bridge – therefore no Forward-Delay necessary!**

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.

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.

Main Differences to STP (1)

- **The three 802.1d states *disabled, blocking*, and *listening* have been merged into a unique 802.1w discarding state**
- **Non-designated ports on a LAN segment are split into *alternate* ports and *backup* ports**
  - **A backup port receives better BPDUs from the same switch**
  - **An alternate port receives better BPDUs from another switch**

In most cases, RSTP performs better than Cisco's proprietary extensions without any additional configuration. 802.1w is also capable of reverting back to 802.1d in order to interoperate with legacy bridges (thus dropping the benefits it introduces) on a per-port basis.

There is no difference between a port in blocking state and a port in listening state; they 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 will be either a designated or root and is on its way to the forwarding state. Unfortunately, once in forwarding state, there is no way to infer from the port state whether the port is root or designated, which contributes to demonstrating the failure of this state-based terminology. RSTP addresses this by decoupling the role and the state of a port.

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 now split into the backup and alternate port roles.

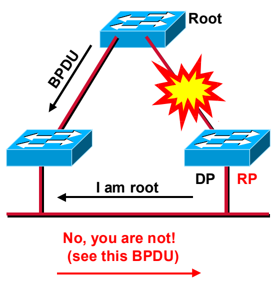
A non-designated port is a blocked port that receives a more useful BPDU than the one it would send out on its segment. The "more useful BPDU" can be received from the same switch (on another port on the same LAN segment) or from another switch (also on the same LAN segment). The first is called a **backup** port, the latter an **alternate** port.

The name *blocking* is used for the *discarding state* in Cisco implementation.

Main Differences to STP (2)

- BPDUs are sent every hello-time, and not simply relayed anymore
  - Immediate aging if three consecutive BPDUs are missing
- When a bridge receives better information ("I am root") from its DB, it immediately accepts it and replaces the one previously stored
  - But if the RB is still alive, this bridge will notify the other via BPDUs

BackboneFast-like behavior:



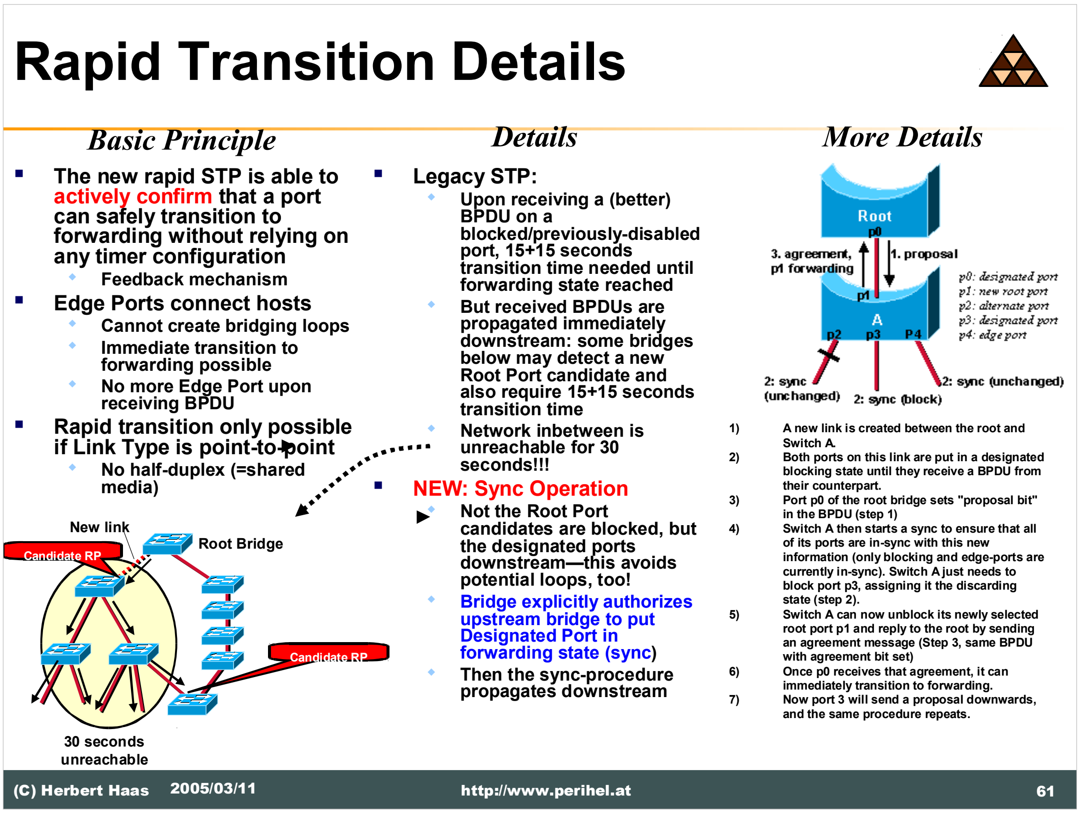
In most cases, RSTP performs better than Cisco's proprietary extensions without any additional configuration. 802.1w is also capable of reverting back to 802.1d in order to interoperate with legacy bridges (thus dropping the benefits it introduces) on a per-port basis.

There is no difference between a port in blocking state and a port in listening state; they 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 will be either a designated or root and is on its way to the forwarding state. Unfortunately, once in forwarding state, there is no way to infer from the port state whether the port is root or designated, which contributes to demonstrating the failure of this state-based terminology. RSTP addresses this by decoupling the role and the state of a port.

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 now split into the backup and alternate port roles.

A non-designated port is a blocked port that receives a more useful BPDU than the one it would send out on its segment. The "more useful BPDU" can be received from the same switch (on another port on the same LAN segment) or from another switch (also on the same LAN segment). The first is called a **backup** port, the latter an **alternate** port.

The name *blocking* is used for the *discarding state* in Cisco implementation.



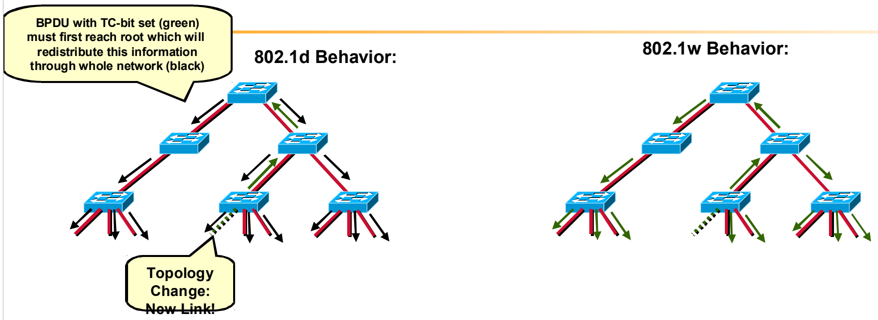
The edge port concept is already well known from Cisco's PortFast feature. Neither edge ports nor PortFast enabled ports generate topology changes when the link toggles. Unlike PortFast, an edge port that receives a BPDU immediately loses its edge port status and becomes a normal spanning tree port.

Note: Cisco's implementation maintains the *PortFast* keyword be used for edge port configuration, thus making the transition to RSTP simpler.

RSTP can only achieve rapid transition to forwarding on edge ports and on point-to-point links. A port operating in full-duplex will be assumed to be point-to-point, while a half-duplex port will be considered as a shared port by default.

Sync Operation: The final network topology is reached just in the time necessary for the new BPDUs to travel down the tree. No timer has been 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 forwarding, bypassing the twice-the-forward-delay long listening and learning stages.

Topology Change

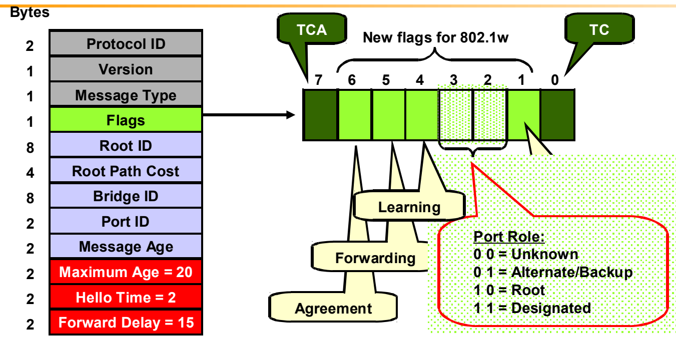


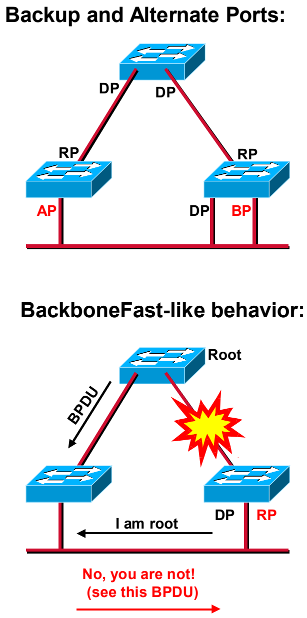
- **802.1d: When a bridge detects a topology change**
  - **A TCN is sent to towards the root**
  - **Root sends Conf-BPDU with TC-bit downstream (for 10 BPDUs)**
  - **All other bridges can receive it and will reduce their bridging-table aging time to *forward_delay* seconds, ensuring a relatively quick flushing of stale information**
- **RSTP: Only non-edge ports moving to the forwarding state cause a TCN**
  - **Loss of connectivity NOT regarded as topology change any more**
  - **TCN is immediately flooded throughout whole domain**
  - **Every bridge flushes MAC addresses and sends TCN upstream (RP) and downstream (DPs)**
  - **Other bridges do the same: Now, the TCN-process is a one-step procedure, as the TCNs do not need to reach the root first and require the root for re-origination downstream**

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

RSTP is able to interoperate with legacy STP protocols. However, it is important to note that 802.1w's inherent fast convergence benefits are lost when interacting with legacy bridges. Each port maintains a variable defining the protocol to run on the corresponding segment. A migration delay timer of three seconds is also started when the port comes up. When this timer is running, the current (STP or RSTP) mode associated to the port is locked. As soon as the migration delay has expired, the port will adapt to the mode corresponding to the next BPDU it receives. If the port changes its operating mode as a result of receiving a BPDU, the migration delay is restarted, limiting the possible mode change frequency.

RSTP Summary





- IEEE 802.1w is an improvement of 802.1d
  - Vendor-independent (Cisco's Uplink Fast, Backbone Fast, and Port Fast are proprietary)
- The three 802.1d states *disabled*, *blocking*, and *listening* have been merged into a unique 802.1w discarding state
- Nondesignated ports on a LAN segment are split into *alternate* ports and *backup* ports
  - A backup port receives better BPDUs from the same switch
  - An alternate port receives better BPDUs from another switch
- Other changes:
  - BPDU are sent every hello-time, and not simply relayed anymore.
  - Immediate aging if three consecutive BPDUs are missing
  - When a bridge receives inferior information ("I am root") from its DB, it immediately accepts it and replaces the one previously stored. If the RB is still alive, this bridge will notify the other via BPDUs.

(C) Herbert Haas 2005/03/11 http://www.perihel.at 63

In most cases, RSTP performs better than Cisco's proprietary extensions without any additional configuration. 802.1w is also capable of reverting back to 802.1d in order to interoperate with legacy bridges (thus dropping the benefits it introduces) on a per-port basis.

There is no difference between a port in blocking state and a port in listening state; they 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 will be either a designated or root and is on its way to the forwarding state. Unfortunately, once in forwarding state, there is no way to infer from the port state whether the port is root or designated, which contributes to demonstrating the failure of this state-based terminology. RSTP addresses this by decoupling the role and the state of a port.

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 now split into the backup and alternate port roles.

A non-designated port is a blocked port that receives a more useful BPDU than the one it would send out on its segment. The "more useful BPDU" can be received from the same switch (on another port on the same LAN segment) or from another switch (also on the same LAN segment). The first is called a **backup** port, the latter an **alternate** port.

The name *blocking* is used for the *discarding state* in Cisco implementation.

Other

- There is no 15-sec forwarding delay anymore
  - TCN ensures that all tables are immediately flushed
- Protection against misordering and duplication
  - Port state transitions to Learning and Forwarding are delayed
  - Ports can temporarily transition to the Discarding state
- RSTP provides rapid recovery to minimize frame loss

Note

- A bridge must first receive a BPDU from the Root Bridge until BPDUs from Non-Root-Bridges can be forwarded
- Every bridge sends BPDUs periodically (by default every 2 seconds) and the neighbor bridge is declared dead when three subsequent BPDUs are missing
- Upon a topology change (e. g. neighbor dead) the bridge sends BPDUs with the Proposal Bit set which triggers a recalculation of the STP

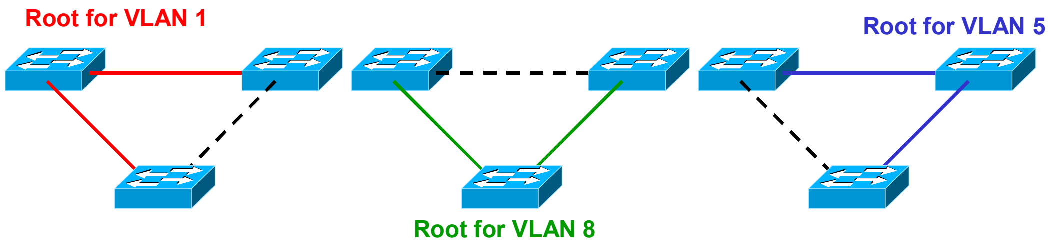
Cisco Extensions: PVST(+)

Per-VLAN Spanning Tree

About

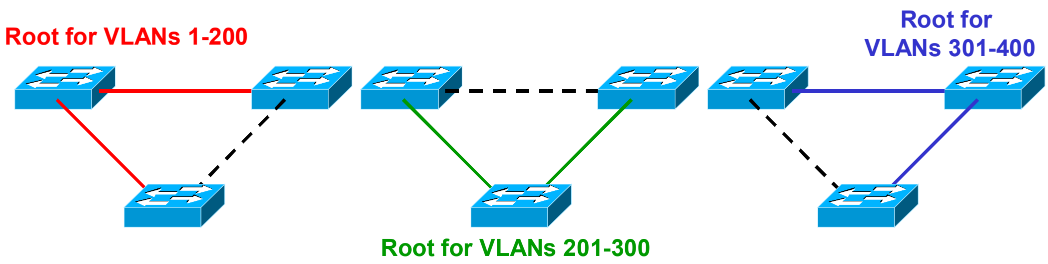
- In over 70% of all enterprise networks you will encounter Cisco switches
- Cisco extended STP and RSTP with a per-VLAN approach: "Per-VLAN Spanning Tree"
- Advantages:
  - Better (per-VLAN) topologies possible
  - STP-Attacks only affect current VLAN
- Disadvantages:
  - Interoperability problems might occur
  - Resource consumption (800 VLANs means 800 STP instances)

Example



- **Remember that root bridge should realize the center of the LAN**
  - Attracts all traffic
  - Typically servers or Internet-connectivty resides there
- **Different VLANs might have different cores**
- **PVST+ allows for different topologies**
  - Admin should at least configure ideal root bridge BID manually

Scalability Problem



- Typically the number of VLANs is much larger than the number of switches
- Results in many identical topologies
- In the above example we have 400 VLANs but only three different logical topologies
  - 400 Spanning Tree instances
  - 400 times more BPDUs running over the network

PVST (Classical, OLD!)

- Cisco proprietary (of course)
- Interoperability problems when also standard CST is used in the network (different trunking requirements)
- Provides dedicated STP for every VLAN
- Requires ISL
  - Inter Switch Link (Cisco's alternative to 802.1Q)

PVST+

- Today standard in Cisco switches
  - Default mode
  - Interoperable with CST
- The PVST BPDUs are also called SSTP BPDUs
- The messages are identical to the 802.1d BPDU but uses SNAP instead of LLC plus a special TLV at the end

PVST+ Protocol Details

- For native VLAN on trunk, normal (untagged) 802.1d BPDUs are sent
  - Also to the IEEE destination address 0180.c200.0000
- For tagged VLANs, PVST+ BPDUs use
  - SNAP, OID=00:00:0C, and EtherType 0x010B
  - Destination address 01-00-0c-cc-cc-cd
  - Plus 802.1Q tag
- Additionally a "PVID" TLV field is added at the end of the frame
  - This PVID TLV identifies the VLAN ID of the source port
  - The TLV has the format:
    - type (2 bytes) = 0x00 0x34
    - length (2 bytes) = 0x00 0x02
    - VLAN ID (2 bytes)
    - Also usually some padding is appended

PVST+ Compatibility Issues

- PVST+ switches can act as translators between groups of Cisco PVST switches (using ISL) and groups of CST switches
  - Sent untagged over the native 802.1Q VLAN)
  - BPDUs of PVST-based VLANs are practically 'tunneled' over the CST-based switches using a special multicast address (the CST based switches will forward but not interpret these frames)
- Not important anymore…

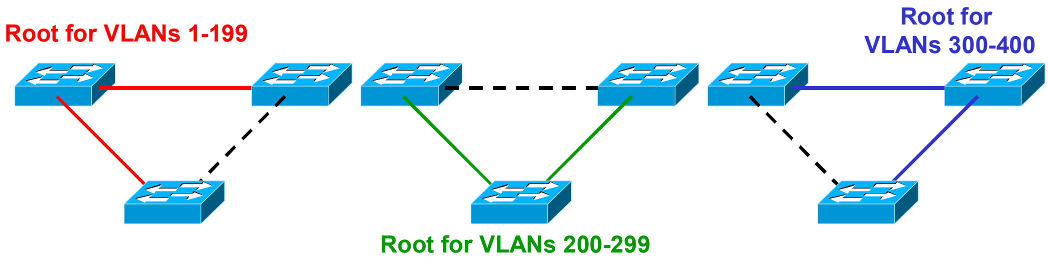
MSTP

Text durch Klicken hinzufügen

Overview

- Also the MSTP standard contains contributions from Cisco
- Solves the cardinality mismatch between the number of VLANs and the number of useful topologies
- Switches are organized in Regions
- In each Region sets of VLANs can be independently assigned to one out of 16 Spanning Tree Instances
- Each Instance has its own Spanning Tree topology

Example



- Compared to PVST+ only three Spanning Tree Topologies (=Instances) required
- Each STP instance has assigned 200 VLANs
  - Each VLAN can only be member of one instance of course

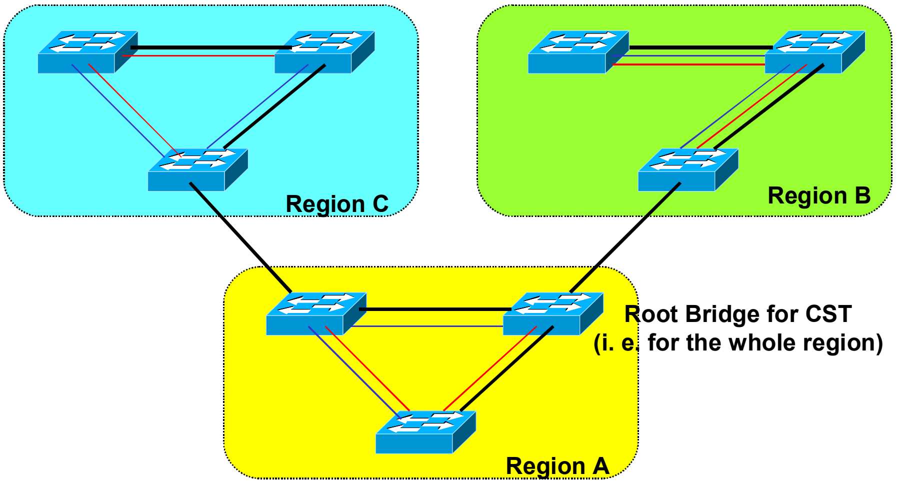
MSTP Details

- Each switch maintains its own MSTP configuration which contains the following mandatory attributes:
  - The Configuration name (32 chars),
  - The revision number (0..65535),
  - The element table which specifies the VLAN to Instance mapping
- All switches in a Region must have the same attributes

Regions

- The bridges checks attribute equivalence via a digest contained in the BPDUs
  - Note that the attributes must be configured manually and are NOT communicated via the BPDUs
- If digest does not match then we have a region boundary port
- Regions are only interconnected by the Common Spanning Tree (CST)
  - Instance 0
  - Uses traditional 802.1d STP

Region Example



- Only the logical STP topologies are shown (not the physical links)
- Each region has internal STP instances (red and blue)
- One CST instance interconnects all regions (black)

Note

- When enabling MSTP, per default the CST (instance zero) has all VLANs assigned
- Each region must be MSTP-aware
  - Since only a subset of VLANs is assigned to the CST
  - Old-STP switched always create a general (all-VLAN) topology
  - Don't let MSTP-unaware switch become root bridge

Any Questions?

THE ANSWER IS … FORTY-TWO!

The choice of 0x42 as the LLC SAP value for BPDUs has an interesting history. First, the chair and editor of the IEEE 802.1D Task Force (Mick Seaman) was British, and 42 is "The Answer to the Ultimate Question of Life, the Universe, and Everything" in *The Hitchhiker's Guide to the Galaxy*, a popular British book, radio, and television series [by Douglas Adams] at the time of the development of the original standard.

Even in the United States, the series was so popular that the original Digital Equipment Corp. bridge architecture specification was titled eXtended LAN Interface Interconnect, or XLII, the Roman representation of 42.

From Rich Seifert's Switch Book

Rich Seifert also continues:

"Finally, 0x42 is a palindrome; it has the same binary pattern regardless of whether one transmits the most–significant bit first or the least-significant bit first —01000010. This eliminates any confusion regarding bit ordering of the field when transmitted on Little Endian (e. g. Ethernet) versus Big Endian (e. g. Token Ring) networks, although this side benefit was not recognized until after the value was assigned."