
IEEE 802.1D Bridging

IEEE 802.1D

- Transparent Bridging
- Expedited Traffic Capabilities
- Selective Multicasting

IEEE 802.1D

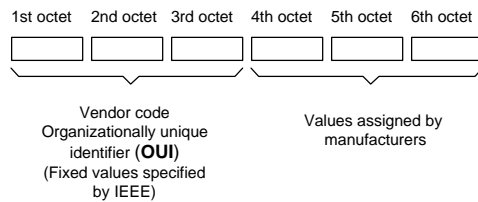
Transparent Bridging

LAN Addressing

- IEEE 802 standardized length of address field
 - 16-bit addresses
 - 48-bit addresses
- 16-bit addresses are sufficient for single LANS, provided LAN manager is capable of assigning addresses to the stations.
- 48-bit addresses allow stations to be provided with globally unique identifiers - (Plug and Play).
- 16-bit address option has not caught on and may be ignored.

Globally Unique LAN Addresses

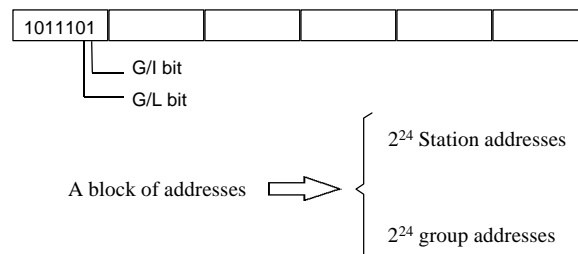
- IEEE is the official authority responsible for handing out blocks of addresses.
 - \$1,250 for a block of addresses



- 802.1 committee's canonical form
 - e.g. 00 - 90 - AC - 59 - 31 - 51
 - Assumes least-significant-bit-first order

Group/Individual (G/I) Bit

- G/I bit is defined as the *first bit on the wire*
 - If G/I bit is 0, address refers to a particular station
 - If G/I bit is 1, address refers to a logical group of stations (or multicast address)
 - Special case: all 1's address means broadcast.

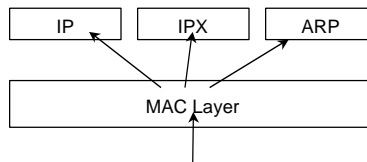


Global/Local (G/L) Bit

- Blocks of addresses purchased from IEEE have G/L bit set to 0.
- Addresses with G/L bit set to 1 are freely available. It is up to network manager to assign these addresses & make sure that there are no address collisions.

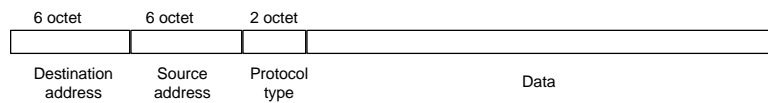
Protocol Type Multiplexing Field

- It is possible for multiple higher-layer protocols to be implemented in the same station.
- There must be a way to determine which protocol should receive the packet
- There must be a way to determine which protocol constructed the packet.



Protocol Type Multiplexing

- In original Ethernet, 2-octet protocol type field currently administered by the IEEE.

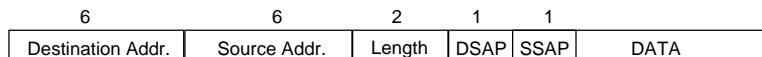


- More information can be found at:

<http://standards.ieee.org/regauth/ethertype/index.html>

Service Access Point Multiplexing (1)

- More flexible to have separate fields for source & destination
 - May assign number to protocols differently in each machine



1 0 1 0 1 0 1 0

M bit (always 0; 1 is reserved)
Global/Local bit

All 1's means "all SAP's".

All 0's reserved for the data link layer itself.

6 bits for globally assigned individual SAP numbers (by IEEE).

Strict Rules: Only protocols designed by a standard body approved by IEEE may be assigned a number

SAP Multiplexing (2)

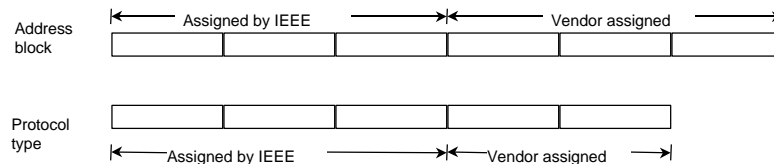
- Protocols that are not assigned global SAP values could use locally assigned SAP numbers.
 - managers must insure that each protocol had a unique # within that system.
 - conversation start-up is difficult.

Subnetwork Access Protocol SNAP SAP (1)

- SNAP SAP is a single globally assigned SAP value
 - 10101010 (AA in hex)
- When DSAP contains the SNAP SAP, this implies header is expanded to include a “protocol type” field.
- Protocol type field could then be large enough so that a global authority could assure that every protocol is assigned a number.

SNAP SAP (2)

- Administration of protocol type is then linked to administration of addresses.

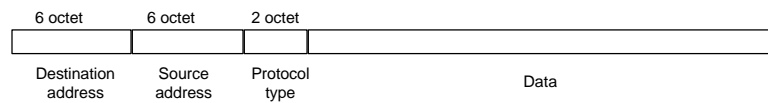


Transmission Bit Order

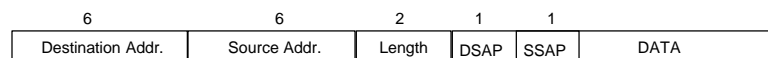
- 802.3 and 802.4
 - Least significant bit is transmitted first.
- 802.5 and FDDI
 - Most significant bit is transmitted first.
- *Issue:* G/I bit defined as the 1st bit on the wire.
Therefore, bits in the address fields must be shuffled when forwarding between 802.5 (and FDDI) and other LANS

802.3 and Ethernet Frame Formats

- Ethernet frame format:



- 802.3 frame format

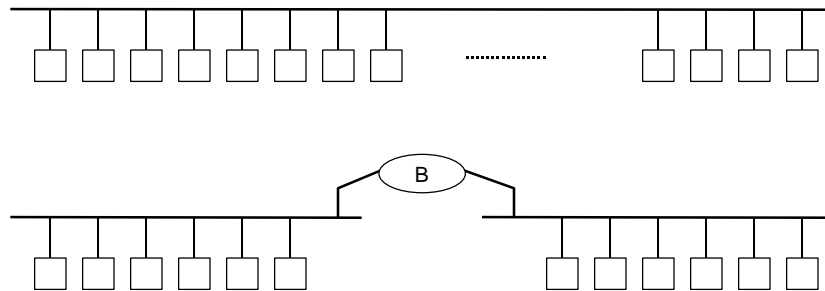


- Formats can co-exist; lengths are always less than 1536 bytes (0600 hex); types start at 0600.

Reasons for Bridges

- Limited number of stations per segment.
- Limited size of a segment.
- Limited bandwidth per segment.

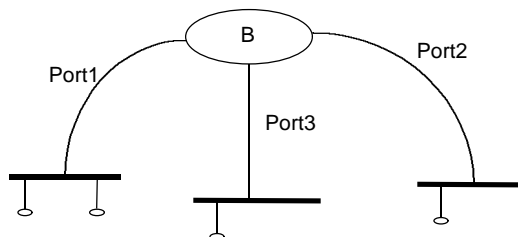
Transparent Bridging



As far as the stations are concerned, these two topologies should be the same; the bridge is “invisible” (transparent).

Transparent Bridge Functions

- Functions:
 - Promiscuous listen;
 - Store and forward (based on a forwarding database)
 - Filtering



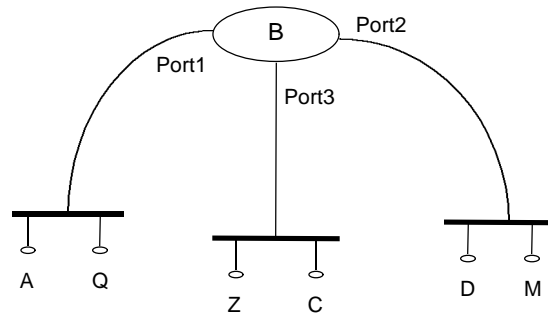
Creating and Maintaining the Forwarding Database

- The bridge listens promiscuously, receiving every frame transmitted.
- For each frame received, the bridge stores the address in the frame's source address field in the forwarding database, together with the port on which it was received.
- The Bridge ages each entry in the station cache and deletes it after a period of time (aging time) in which no traffic is received with the address as the source address.

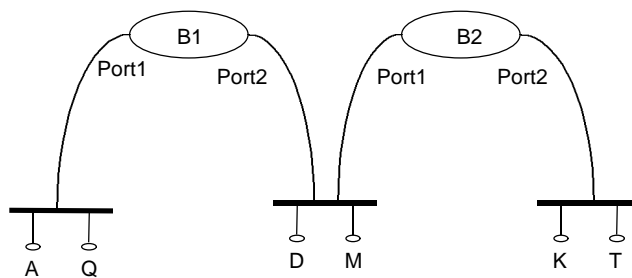
Forwarding Frames

- For each frame received, the bridge looks at the frame's destination address:
 - If the address is a multicast address or the broadcast address (all 1's) then the bridge forwards the frame onto all its interfaces except the one on which the frame was received.
 - If the address is a unicast address, then the bridge looks through its forwarding database:
 - If the address is found in the forwarding database, the bridge only forwards the frame onto the interface specified in the table; If the specified interface is the one from which the frame was received, the frame is dropped.
 - Otherwise, if the destination address is not found in the forwarding database, the bridge forwards the frame onto all its interfaces except the one from which it was received.

Example 1

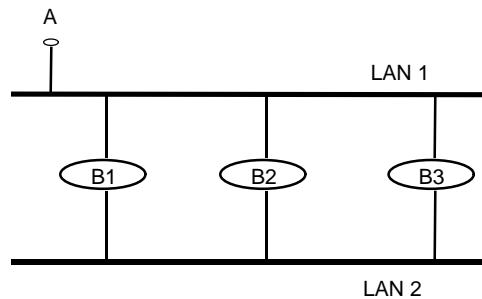


Example 2



Topologies with Loops (1)

- Problems:
 - Frames proliferate
 - Learning process unstable
 - Multicast traffic loops forever



Topologies with Loops (2)

- Solutions:
 - 1) Require that topologies be loop-free; through careful deployment of LAN segments and bridges;
 - 2) Design bridges to detect existence of loops in the topology and complain.
 - 3) Design into the bridge an algorithm called the spanning tree algorithm, that prunes the topology into a loop-free subset (a spanning tree), and automatically adapts to topology changes.

Reconfiguration Algorithm Requirements

- Configures an arbitrary topology into a spanning tree.
- Automatic reconfiguration in case of topology changes, with no transient loops.
- Entire topology should stabilize for any size LAN. Should stabilize with high probability within a short, bounded time.
- Active topology should be reproducible and manageable.
- Transparent to end-stations.
- Must not use a lot of bandwidth.

Spanning Tree Algorithm (1)

- A distributed Algorithm which:
 - 1) Elects a single bridge to be the root bridge;
 - 2) Calculates the distance of the shortest path from each bridge to the root bridge;
 - 3) For each LAN, elects a “designated bridge” from among the bridges residing on that LAN; whereby, the designated bridge is the one closest to the root bridge.

Spanning Tree Algorithm (2)

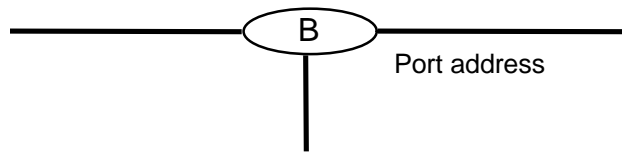
- 4) For each bridge
 - Select ports to be included in the spanning tree.
 - The ports selected are
 - The root port: the port that gives the best path from the bridge to the root bridge;
 - ports on which the bridge has been selected as the designated bridge
 - Ports that are selected are placed in the *forwarding* state
 - All other ports are placed in the *blocked* state

Forwarding Packets Along the Spanning Tree Forward and Blocked States for Ports

- Data Traffic is forwarded to and from ports selected in the spanning tree.
- Data traffic is always discarded upon receipt from ports in the blocked state data traffic is never forwarded onto ports that are in the blocked state

Bridge ID

- Each port has a separate unique LAN address (48-bit address). The bridge is given a single bridge-wide ID:
 - A unique 48-bit address
 - e.g., LAN address on one of the ports



- Root bridge to be selected is the bridge with the numerically lowest bridge ID.

Path Length (Cost)

- Path length in number of hops from a bridge to the root bridge.
- Interested in the *least cost path* to the root.

Configuration Message

- Configuration Bridge Protocol Data Unit (BPDU's)
 - Transmitted by bridges to implement the spanning tree algorithm
 - Ordinary LAN data link layer header

destination	source	length	DSAP	SSAP	Configuration message
-------------	--------	--------	------	------	-----------------------

- Destination address : Special multicast address ("to all bridges")
(01-80-C2-00-00-00)
- Source address: Address of port of the bridge transmitting the configuration message
- DSAP = SSAP = 01000010

Configuration Message Content

- Root ID:
 - ID of the bridge assumed so far to be the root
- Cost:
 - cost of the least cost path to the root from the transmitting bridge (known so far).
- Transmitting Bridge ID:
 - ID of the bridge transmitting.
- Representation:
 - <Root ID>.<Cost>.<Transmitting bridge ID>

Transmission and Processing of Configuration Messages

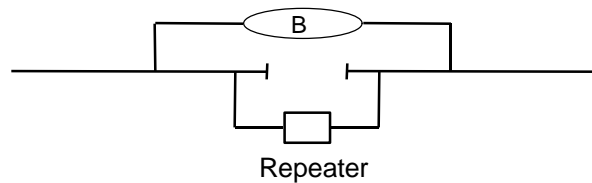
- Initially,
 - each bridge assumes it is the root
 - Transmits on each of its ports:
 - <Transmitting Bridge ID>.<O>.<Transmitting Bridge ID>
- A Bridge receives configuration messages on each of its ports.
- For each port, it saves the “best” configuration message among all messages received on that port *and* the message transmitted on that port.

Comparing Two Configuration Messages

- Given two configuration messages C1 and C2
 - C1 is better than C2 if root ID in C1 is numerically lower than root ID in C2;
 - If root ID's are equal, then C1 better than C2 if the cost listed in C1 is numerically lower than the cost in C2;
 - If root ID & cost are equal, C1 is better than C2 if the transmitting bridge ID in C1 is numerically lower than the transmitting bridge ID listed in C2.

Port Identifier

- Port Identifier
 - Each bridge has an internal numbering of its own ports.
 - Port identifiers are used as a tie-breaker.
 - Useful if bridge has multiple ports attached to the same LAN segment



Example of Configuration Messages Received on a Port

	C1			C2		
	Root ID	Cost	Transmitter	Root ID	Cost	Transmitter
a	29	15	35	31	12	32
b	35	80	39	35	80	40
c	35	15	80	35	18	38

In all cases, a, b, c, C1 is better than C2

Designated Bridge for a LAN

- If a bridge receives a “better” configuration message on a LAN than the one it would transmit, it no longer transmits configuration messages.
- Therefore, when the algorithm stabilizes, only one bridge on each LAN (the designated bridge for that LAN) transmits configuration messages on that LAN.

Determination of Root ID and Cost to Root

- Assume Bridge B's ID is 18.
- Best configuration message received on each of its ports

	Root	Cost	Transmitter
Port 1	12	93	51
Port 2	12	85	47
Port 3	81	0	81
Port 4	15	31	27

⇒ Best root is 12, (since $12 < 18$)
 Best cost is $85+1 = 86$
 Port 2 is its root port

⇒ Its own configuration message will be 12.86.18.
 Better than those received on ports 1, 3, and 4. It is the designated bridge for ports 1,3,4 and will transmit configuration message on those ports.

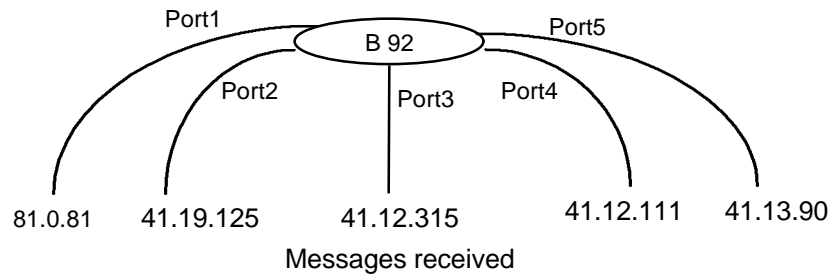
Determining Designated Bridge for Each LAN

- Once B has determined
 - the identity of the root
 - its own distance to the root
- B's configuration message becomes:
 - <root ID>.<distance to root>.<B's ID>
- For each port: B compares above with best configuration message received on that port.
- If B's configuration message is better,
 - i. B assumes for now that it is the designated bridge for that port;
 - ii. will transmit configuration message on that port.
- Otherwise, it is not, and stops sending configuration messages on that port.

Selecting Spanning Tree Ports

- The port chosen by B as its preferred path to the root (B's root port)
- All ports for which B is the designated bridge
- Ports selected in the spanning tree are placed in the *forwarding* state
 - B will forward packets to and from those ports
- All other ports are placed in the *blocking* state
 - B will not forward packets to and from those ports

Example (1)



Best Known root is 41

Best cost to root is $12 + 1 = 13$ (ports 3 or 4) - selects port 4

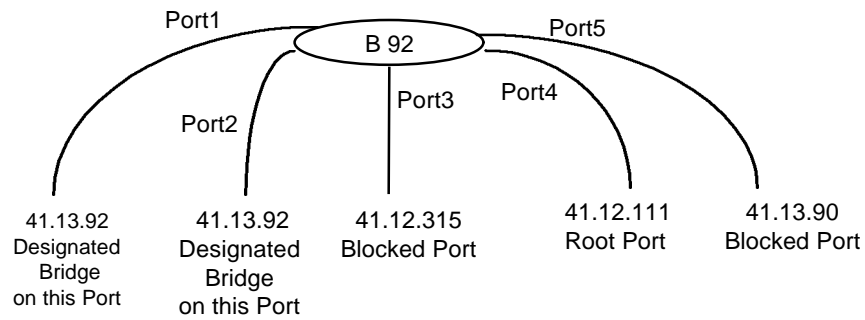
⇒ Configuration message for B92 is now 41.13.92

Example (2)

Port 1: 41.13.92 < 81.0.81	B92 is designated bridge
Port 2: 41.13.92 < 41.19.125	B92 is designated bridge
Port 3: 41.13.92 > 41.12.315	Switch to blocking state
Port 4:	Root port
Port 5: 41.13.92 > 41.13.90	Switch to blocking state

- Blocking state:
B92 will continue to run the spanning tree algorithm on those ports. But will not:
 - receive data messages from those ports
 - learn location of station addresses from them
 - forward traffic onto them

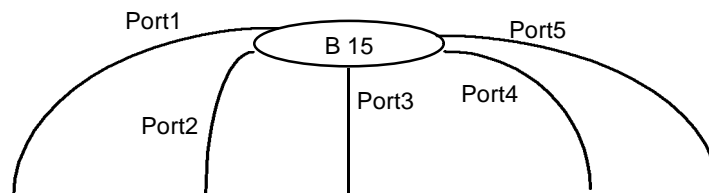
Example (3)



B92 overwrites messages on ports 1 and 2.

Example (4)

- What if B92 was actually B15?



Maintaining the Spanning Tree Age Field (1)

- Configuration message for each port is stored
 - along with message age field
 - incremented by 1 after each unit of time
- When age = “max age”, configuration message is discarded and bridge must recalculate as if it had never received a configuration message from that port.

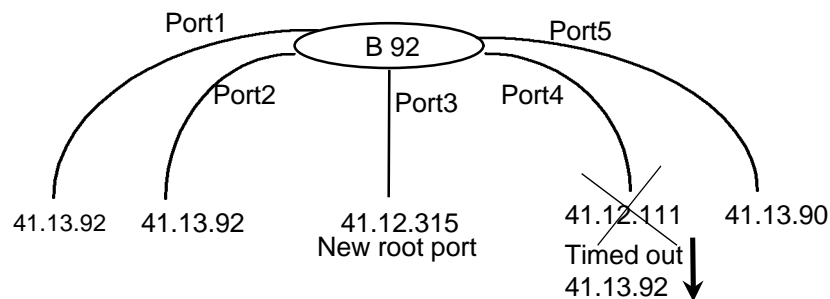
Maintaining the Spanning Tree Age Field (2)

- Root bridge generates & transmits configuration messages periodically
 - every “hello time”.
 - with message age field = 0.
- When a 1st tier bridge receives the root’s message, it transmits a configuration message on each port for which it is the designated bridge with message age = 0.
- Likewise for bridges downstream ...

Maintaining the Spanning Tree Failures (1)

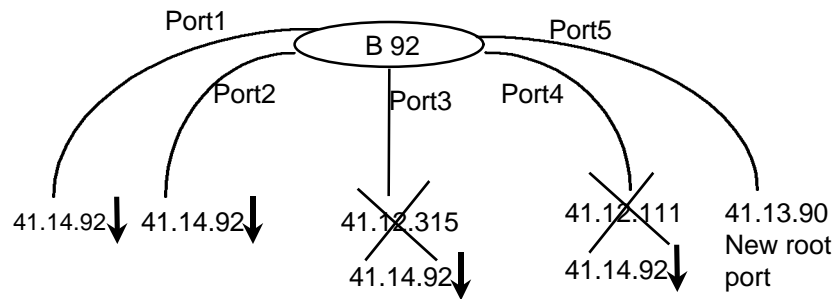
- If root bridge or any other component on the path between a bridge and the root fails, the bridge stops receiving “fresh” messages, and eventually discards the stored configuration message.
- Bridge then recalculates root, root path cost, and root port.

Maintaining the Spanning Tree Failures (2): Example -a



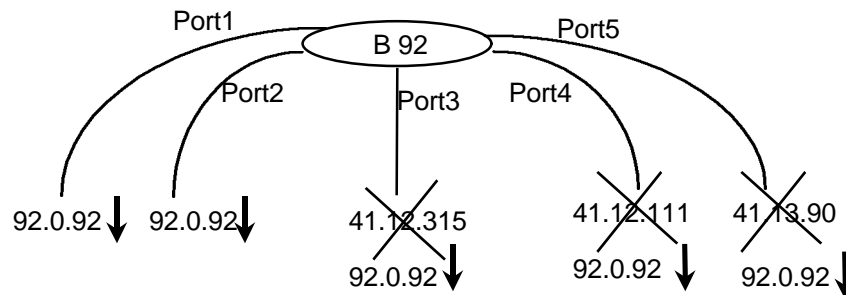
- B92 switches root port from port 4 to port 3

Maintaining the Spanning Tree Failures (2): Example -b

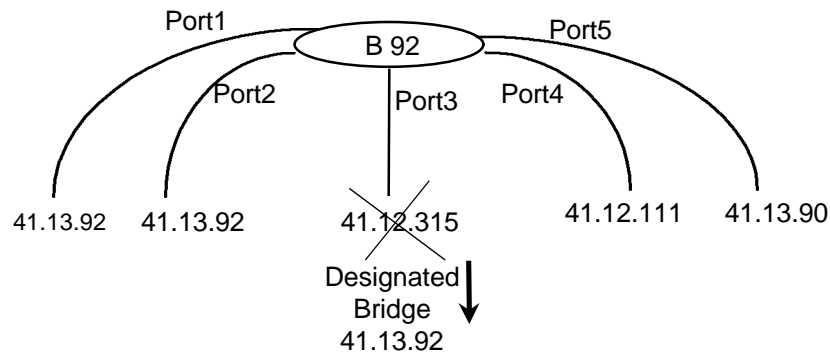


- B92 chooses port 5 as its root port.

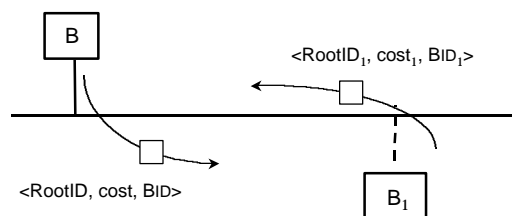
Maintaining the Spanning Tree Failures (2): Example -c



Maintaining the Spanning Tree Failures (2): Example -d



Maintaining the Spanning Tree Addition of a Bridge (1)



- Assume B 's configuration message on some port for which it is designated bridge has age X .
- Assume that bridge B_1 is added to the LAN.
 - B_1 issues its own configuration message

Maintaining the Spanning Tree Addition of a Bridge (2)

- If B's configuration message is better than that of B_1
 - B should not ignore B_1 's message.
 - B transmits its configuration message with age X. (Omitting X, or considering age to be 0 would slow down discovery of failures).
 - If B_1 's configuration message is better than that of B
 - B should overwrite its own configuration message and undertake spanning tree recalculation

In Summary: Events Causing Spanning Tree Recalculation

- Receipt of a configuration message on port P.
 - Bridge compares received message with stored message.
 - If received is "better" or the same but has a smaller age, then stored configuration message is overwritten.
- Timer Tick: Age reaches max Age on a port.

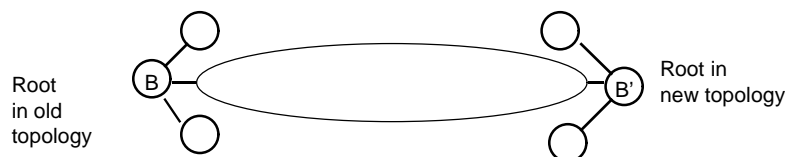
Avoiding Temporary Loops (1)

- Topological changes causing reconfiguration may lead to:
 - Temporary loops:
 - A bridge port that was in the forwarding state in the old topology hasn't yet found out that it needs to be in the blocked state in the new topology.
 - Temporary loss of connectivity:
 - A bridge port that was in the blocked state in old the topology hasn't yet found out that it should be in the forwarding state in the new topology.

In bridged networks: *Temporary loss of connectivity is better than temporary loops.*

Avoiding Temporary Loops (2)

- Force ports in Blocked state to wait some amount of time before they transition to the Forwarding state.
- Wait period should be large enough for information regarding new topology spreads:
 - Twice the maximum transit time across the network.

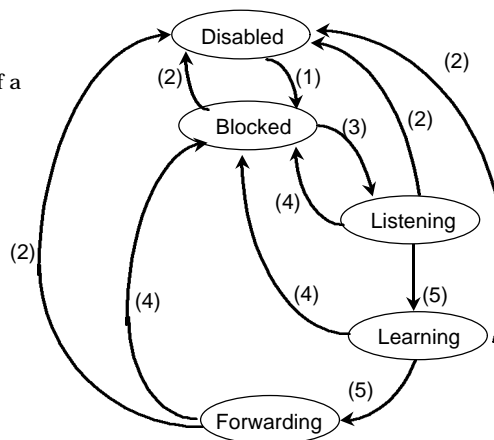


Avoiding Temporary Loops (3)

- IEEE 802.1 standard defines 2 intermediate states:
 - Listening intermediate state
 - Bridge does not learn station addresses
 - Learning intermediate state
 - Bridge starts learning about station addresses.
 - Does not forward packets over that port.

Avoiding Temporary Loops (4)

- (1) Port enabled by receipt of a management BPDU
- (2) Port disabled either by receipt of a management BPDU or failure
- (3) Port selected as a root or designated port
- (4) Port ceases to be a root or designated port
- (5) Forwarding timer expires



Forwarding Database Time-out Values (1)

- Stations locations might change
- Forwarding Database entries must be refreshed
- Choosing a suitable time out period:
 - too long --> Traffic is lost for unreasonably long time (sent out on the wrong port)
 - too short --> Traffic is forwarded unnecessarily (broadcast)
- Two circumstances requiring time-outs:
 - Station moving (15 minutes)
 - Network getting reconfigured (15 seconds)

Forwarding Database Time-out Values (2)

- Network - management - settable:
- 2 values:
 - a long value, used in the usual case;
 - a short value, used following spanning tree reconfiguration
- Spanning tree algorithm enhanced to reliably advise all bridges that the spanning tree has reconfigured
 - Notify the root;
 - Root sets a flag in its configuration message: “topology change flag”

Bridge Settable Parameters (1)

- Bridge Priority:
 - 2 - octet values that allows network manager to influence choice of a root bridge and designated bridge.
- Port Priority:
 - 1 - octet value that allows the network manager to influence the choice of port
- Hello time:
 - Time between generation of configuration messages by the bridge when it is the root bridge.
 - Recommended time: 2 seconds

Bridge Settable Parameters (2)

- Max age:
 - message age value at which a stored configuration message is judged “too old” and is discarded.
 - conservative value is to assume a delay variance of 2 seconds per hop.
 - IEEE 802.1D recommends 20 seconds (assuming a 10 hop network).
- Forward delay:
 - Time during which a bridge is prevented from starting to forward packets to and from a link (to allow news of a topology change to spread to all ports of a bridged network).
 - IEEE 802.1D recommends 15 seconds.

Bridge Settable Parameters (3)

- Long Forwarding Database timer:
 - Default recommended in IEEE 802.1D is 5 minutes.
- Path cost:
 - Value individually settable on each port.
 - Cost to be added to the root path cost field in a configuration message received on a port to determine the cost of the path to the root through that port.
 - Large value \Rightarrow port more likely to be a leaf

Network Wide Parameters

- Max Age:
 - time at which a configuration message is discarded
- Hello Time:
 - time interval between issuing configuration messages
- Forward Delay:
 - amount of time in “listening” and “learning” states.
- Root bridge includes these parameters in its configuration messages
- A bridge that is the designated bridge for some port copies the values it receives from the root into configuration messages it transmits.

Bridge Message Format (1)

# of octets	Configuration Message			Flags
2	Protocol Identifier			
1	Version			
1	Message type			
1	TCA	reserved	TC	
8	Root ID			
4	Cost of path to root			
8	Bridge ID			
2	Port ID			
2	Message age			
2	Max age			
2	Hello time			
2	Forward delay			

Bridge Message Format (2)

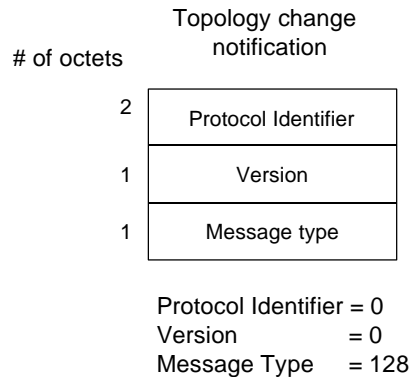
1. Protocol Identifier: 0
2. Version: 0
3. Message type: 0
4. Flags:

TC: topology change notification
when received on root port

Use forward
delay
Use short cache
timer

TCA: topology change notification
acknowledgment.

Topology Change Notification (1)



Topology Change Notification (2)

- When a bridge notices the spanning tree algorithm has caused it to transition a port into or out of the blocking state, it transmits a topology change notification message on its root port
- A bridge that receives the topology change notification on a port for which it is the designated bridge:
 - Sets the TCA flag in the next configuration message it transmits on that port
 - Transmits a topology change notification on its own root port

Topology Change Notification (3)

- When the root bridge receives a topology change notification, it sets the TC flag in its configuration messages for a time period equal to the sum of the forward delay and max age
- Bridges receiving configuration messages with the TC flag set should use the short forwarding database timer instead of the long forwarding database timer

Configuration Filtering

- Sometimes desirable to keep certain kinds of traffic confined to portions of the topology
- 802.1 standard specifies that network management have the ability to set certain addresses as being permanently in the filtering database of a bridge with instructions as to which ports the bridge should allow the packets to traverse. (ordinarily, this would be used for multicast address.)