Evolutions of the Spanning Tree Protocol

Fulvio Risso
Politecnico di Torino

Evolutions of the STP

- Rapid Spanning Tree protocol
  - Fast convergence
  - Appropriate for current network topologies (no hubs, only point-to-point links)

- Multiple Spanning Tree
  - Metropolitan area networks
  - Coexistence of STP and RSTP within the same domain
Rapid Spanning Tree (RSTP, 802.1w) (1)

- Modern solution for mission critical BLAN
  - Fast convergence (less than 1 second)
  - Replaces other proprietary solutions with fast convergence introduced by many vendors
- Operates only on point-to-point links
  - Direct connections (no hubs)
- Major improvements
  - Fast topology convergence
  - Fast update of the filtering database
Rapid Spanning Tree (RSTP, 802.1w) (2)

- Interoperable with IEEE 802.1D
  - Without fast convergence
  - 802.1w and 802.1t-2001 were integrated in IEEE 802.1D-2004
- Defines a set of symbols to be used in network design
Fast convergence

Requirements

- All the switches must be 802.1w
- Links that may create a mesh between different switches
  - Must be point-to-point (twisted pair or fiber), full duplex
    - STP supported also shared medium (e.g. coax)
  - No links terminated on hubs (although still possible at the edge; often marked as “shared links”)
  - These links guarantee that devices at both sides of the link detect the fault at the same time and that they initiate the convergence process at almost the same time

Convergence is usually on the order of ~ 10ms

- Faults are immediately detected in the physical layer
- New transition rules for ports helps to improve the convergence
Improvements over STP

Improvements introduced for a fast service recovery

Physical connectivity recovery
- Fast detection of failures over links
  - Detection based on the physical layer
  - Topology map to restore services
- Failure propagation to establish a new topology and port rules
  - Detection based on the protocol
  - Propagation and resolution of the new topology
- Fast transition of the ports to the forwarding state
  - Root Port transition
  - Design Port transition

Filtering Data Base recovery
- Unicast entries update
- Multicast entries update
- Removal of entries related to local changes
- Notifications and appropriate entries removal
- Fast update of local entries
- Notification of the new requirements
- Establish alternative paths before a failure occurs
Port States and Roles: STP

- STP does not distinguish appropriately the two concepts

- Port States
  - Possible operational states with respect to data frames
  - Disabled, Blocking, Listening, Learning, Forwarding
  - Which is the difference between Blocking and Listening?
    - It relates to their status according to the STP topology, but from the operational point of view they behave the same (do not forward frames, do not learn addresses)

- Port Roles
  - Role of the port within the STP topology
  - Root, Designated, Blocking
Port States and Roles: RSTP

- RSTP has a better separation of the two concepts

- Port States
  - Possible operational states with respect to data frames
  - Only 3 states left (from 5 defined in STP)
    - Discarding, Learning, Forwarding
    - Disabled, Blocking and Listening merged in the same state

- Port Roles
  - Role of the port within the RST topology
  - Root, Designated, Alternate, Backup, Edge
### Port states in RSTP

<table>
<thead>
<tr>
<th></th>
<th>Update filtering DB</th>
<th>Process frames</th>
<th>Possible RSTP role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discarding</td>
<td>NO</td>
<td>NO</td>
<td>Alternate or Backup</td>
</tr>
<tr>
<td>Learning</td>
<td>YES</td>
<td>NO</td>
<td>The port is on the way to become Root or Designated</td>
</tr>
<tr>
<td>Forwarding</td>
<td>YES</td>
<td>YES</td>
<td>Root or Designated</td>
</tr>
</tbody>
</table>

### States in STP

<table>
<thead>
<tr>
<th></th>
<th>Receive frames</th>
<th>Forward frames</th>
<th>Process received BPDU</th>
<th>Transmit BPDU</th>
<th>Update filtering DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Blocking</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Listening</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Learning</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Forwarding</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>
New Port Roles (1)

- Blocking state replaced by Alternate/Backup
- Other roles (Root, Designated) remain the same

Alternate

- Alternate port provides an alternate path to the Root Bridge
  - It can replace the root port if that port fails
New Port Roles (2)

- **Backup**
  - If we have a LAN directly connected to two ports of the same bridge, the port that is not Designated will be Backup.
  - It is called “backup” because
    - It provides a backup path for reaching that LAN from the bridge.
    - It cannot guarantee an alternate connectivity to the root bridge.
  - Backup Ports exist only where there are two or more connections from a given Bridge to a given LAN.
    - It does no longer exist on modern networks.
New Port Roles (3)

- Edge ports
  - Port that connects to a single end-station
    - No other switches are served from that point
  - Need an explicit configuration from the network admin
    - The RSTP still monitor the presence of BPDUs on that link
      - Protection against possible loops
    - In case a BPDU is received, the port will become part of the RSTP domain and will change its status according to the standard RSTP rules
New Port Roles (4)

- Edge ports
  - Edge ports work differently from other ports
    - As soon they detect the “link up” signal, they move immediately to Forwarding state without Listening e Learning state
      - No longer have to wait 30s (2 * Max Forward Delay) before having the port fully operational
    - Edge ports become immediately “Designated”
    - A port changing state do not cause a Topology Change Notification BPDU transmission through the root port
Modifications in the BPDU

- Modified
  - BPDU Type (now 2)
  - BPDU Version (now 2)
    - Old bridges can discard new BPDUs
  - Flags

- New flags used for:
  - Encode role and state of the port that generates the BPDU
  - Handle the proposal/agreement mechanism
Principles of the new algorithm

- The tree is created in the same way as STP
  - Root bridge election
  - Root port definition
  - Designated port definition

- Costs and other parameters are handled the same way

- The port not selected as root or designated will become:
  - Alternate if connected to a port on a different bridge
  - Backup if connected to a different port of the same bridge
Improvements against STP

- New BPDU Handling
  - BPDU are *sent* Every Hello-Time
  - Faster Aging of Information
  - Accepts Inferior BPDUs

- Rapid Transition to Forwarding State
  - Edge Ports
  - Link Type
(1) BPDU sent every Hello Time

- BPDU are sent every Hello Time
  - In STP, non-root bridges simply relay BPDUs when receive them from the root port
    - If the root bridge dies, nobody generates BPDUs till Max_Age expires
  - In RSTP, a bridge always generates its BPDUs every Hello Time, even if it does not receive the corresponding BPDUs from the root

- Hello Time: default 2 sec
(2) Faster Aging of Information

- If Hello not received for 3 consecutive times, the current BPDU of the root bridge is declared obsolete
  - No need to wait for Max_Age, although still valid parameter
  - Quick failure detection

- BPDU are then used as “keep-alive” between bridges

- Corollary
  - If a bridge fails to receive BPDUs from a neighbor, it is sure that there is a fault on the link to the neighbor
  - In STP, the problem might have been anywhere from that bridge to the root

- In presence of Full-Duplex links, failures are detected much faster (no need for 3*Hello Time)
  - Exploits signals coming from the physical layer (e.g. “link down”)
(3) Accepts Inferior BPDUs (1)

- If a bridge receives inferior BPDU from its Designated or Root Bridge, it immediately accepts it and replaces the one previously stored

- Inferior BPDU
  - BPDU with a worse RootID \textit{(or)}
  - BPDU with a worse root path cost
(3) Accepts Inferior BPDUs (2)

- Rational
  - If a bridge receives inferior BPDU from its Designated or Root Bridge, it means that something bad happened on its path toward the root bridge
    - E.g., a link failure that increased the root path cost
    - E.g., a loss of connectivity toward the old root bridge, and another bridge elected itself as root
  - The current bridge accepts that information without having to wait for MaxAge
    - The new BPDU will replaces the one previously stored
    - Much quicker convergence
(3) Accepts Inferior BPDUs (3)

- Being received on its Designated/Root port, the information is trustworthy
  - This does not mean that the new information will become immediately active
  - It means that the new info is “accepted” and this triggers a new re-computation of the RSTP without having to wait for the timeout
    - In STP, we had to wait MaxAge in order to accept the new RootID
- E.g., if the new RootID received is worse than the CurrentBridgeID, that bridge will start propagating itself as root
(3) Accepts Inferior BPDUs (4)

- Example
  - B1 detects loss of connectivity from the root
    - I.e., we need to elect another root
  - B1 starts promoting itself as a root bridge
    - B1 does not know the other bridge ID of the network, so it makes sense it elects itself as root
  - B2 and B3 accepts this info because it comes from the root port
    - In case B1 has better bridgeID than B2, B3, the RootID will not change
    - Otherwise, B2/B3 will reply with a better BPDU
(4) Rapid Transition to Forwarding State

- Edge ports connect end users and are not part of the ST topology
  - Edge ports transition immediately in Forwarding State
    - No listening/ learning stage
    - No delayed connectivity for the end user

- Point-to-Point links
  - Defined when the port is in Full Duplex
    - A port in Half Duplex is considered as “shared port” by default
  - Exploit the Proposal/Agreement Sequence in order to bring the port up
Proposal/Agreement Sequence

- When a bridge comes up it puts its ports 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 BPDU it sends out.
- The other bridge may either agree with its proposal (bit “ack” set) or not (it sends its own BPDU to the other party).
Proposal/Agreement Sequence: Example

1) Proposal: new Designated Port

2) Acknowledge: new Root port
Convergence with STP: example

- The Root Bridge adds a new link toward A
  - Link goes into Listening (no data)
    - No loop in the network
  - BPDU generated by root propagated down to B, C, D
  - B, C, D update their STP Topology (e.g. Root Port)
    - A, B, C were previously reachable through D
    - All links involved still in blocking
    - D will move port P1 in Blocking state
  - A, B, C and their leaves unreachable for 2 * ForwardDelay
    - Link Root-A in Forwarding state after 2* Forward Delay
Convergence with RSTP: example (1)

- The Root Bridge adds a new link toward A
  - Both ports put in Designated Blocking, and send their BPDU
  - The bridge that receives the better BPDU activates puts in “sync” all its ports:
    - Blocks all non-edge designated ports
    - Edge ports remain unchanged
    - Alternate and Backup ports remain unchanged
  - This operation is made in order to re-create the proper tree starting from the root and propagate the new topology down toward the edge
Convergence with RSTP: example (2)

- Then, A explicitly authorizes the RB to put its port in Forwarding State
  - A sends the same BPDU received, but with the “ACK” flag set
  - A puts that link in forwarding state too
  - The upstream bridge activates that link as well
- No loops can occur, since downstream ports of bridge A are still blocked
Convergence with RSTP: example (3)

- The process is repeated on B and C: they both negotiate to activate their upstream ports through a SYNC operation
  - Note that B has only edge ports, hence it does not have to block anything in order to allow A to put its lower port in forwarding mode
  - C has to block only port between C and D
- At one point, D will recognize that there are two paths toward the root and it will select the best one
Convergence with RSTP: some notes

- The explicit authorization sent in the SYNC process replaces the 2*Forward Delay of STP
- The final topology is exactly the one calculated by the STP
  - I.e., the blocked port will be exactly in the same place as before
  - Only the steps to obtain this topology have changed
- Explicit negotiation is possible only when bridges are connected by point-to-point links
  - I.e., full-duplex links unless explicit port configuration
Fault detected on the root port

- If an Alternate port exist, it is promoted to Root
  - In case more than one Alternate exist, the best one is promoted
  - This path will become the new path toward the root bridge

- If no Alternate ports exist, the Bridge starts promoting itself as a new Root Bridge
Topology Change Detection

- Only non-edge ports that move to forwarding state cause a Topology Change
  - A loss of connectivity (a port moving to Blocking) is no longer considered a Topology Change
    - It was in 802.1D

- An RTSP bridge that detects a topology change
  - Starts the TC timer (2*Hello Time) for all entries related to its Designated Ports and its Root Port
  - It flushes all MAC addresses associated to these ports
  - It propagates BPDU with the TC bit on all these ports till the TC timer expires
Topology Change Propagation (1)

- When a bridge receives a BPDU with the TC bit set from a neighbor:
  - It clears the MAC addresses learned on all its ports, except the one that receives the topology change
  - It starts the TC 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
- TCN floods very quickly across the entire network
  - No longer involved the Root Bridge (such as in the STP) in order to generate BPDUs with the TC set
Topology Change Propagation (2)

The originator of the TC directly floods this information through the network.
Compatibility with 802.1D (1)

- Interoperability possible
  - Migration delay timer (3 sec) starts when a port comes up
    - During this time, the current STP or RSTP mode associated to the port is locked
    - When 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
  - 802.1D timers (Forward Delay, Max_age) used only as a backup
- Fast convergence is lost
- Problem: an 802.1w bridge that starts operating in 802.1D compatibility mode does not turn back in 802.1w unless an explicit configuration is taken
Compatibility with 802.1D (2)

Example

- A has the Designated Port over the LAN
- A receives a 802.1D BPDU from C
  - Please note that C advertises itself as Designated Port, since it cannot understand the 802.1w BPDU generated by Bridge A
- A changes in compatibility mode and starts generating 802.1D BPDU
- If A has the best BridgeID, C will put its port in Blocking State
Conclusions

- Efficient
- Fast convergence (often < 1s)
- Can replace many proprietary protocols
- Possibility to mix devices from different vendors
- Interoperability problems with STP
  - Mostly used with Multiple Spanning Tree (MST)