The Spanning Tree Protocol

Andrea Marin

Università Ca' Foscari di Venezia
Dipartimento di Informatica
Corso di Sistemi Distribuiti

2009

Presentation outline

1. Introduction
   - Local internetworking
   - Motivations
2. Protocol description
   - High level description
   - Control protocol
3. The original formulation (1985)
   - Port states
   - Ageing and hello packets
   - Electing root
   - Managing ports

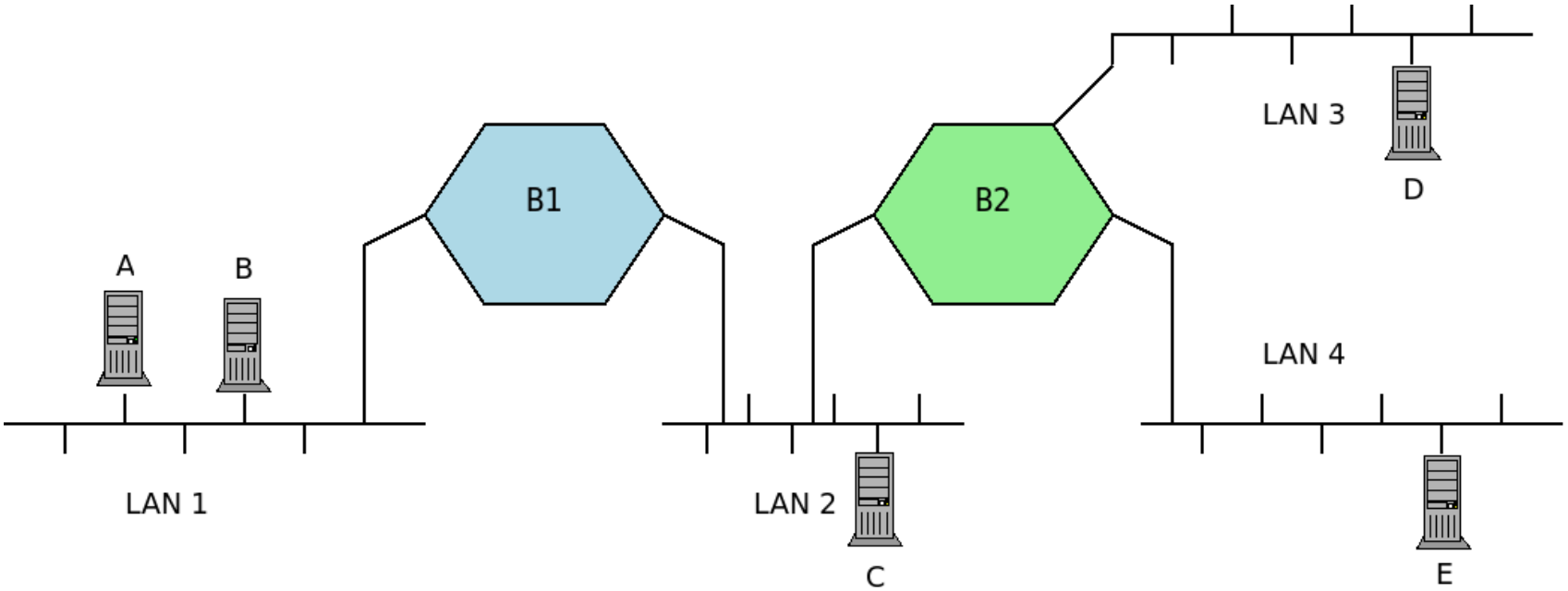
Local intertnetworking

- We have local internetworking when we want to connect a set LANs e.g. within the same company
- This can be done at network level (with routers) or at Data Link level (with Bridges)
- Connecting at Data Link level allows any protocol at Network level (IP or others)
- If the LANs use different standards several problems arise (See Tanenbaum, Computer Networks, Sec. 4.7.1)
- We consider the case of 802.3 LANs

Transparent bridges

- Transparent Bridges are used to connect different LANs
- They do not need configuration
- They *learn* the network topology by the analysis of the traffic (backward learning)
- They use a dynamic table to keep the pairs destination/output lines
- They react to changes in the network topology
- When they don't know where they have to forward a message they apply the flooding over all the lines

Example



Purposes

- In order to increment the dependability of LAN inter-connections redundant bridges are often used
- This causes loops in the LAN topology $\Rightarrow$ flooding interrupts connectivity
- Spanning tree protocol (STP)...
  - Avoids loops by reducing network connectivity
  - Reacts to changes in the network topology (either good news or bad news)
  - Is completely transparent to the user
- STP implements fault tolerance at data link level!

Preliminaries

- The network topology is seen as a graph
  - The net segments are the nodes
  - Bridge connections are the edges
  - Edge weights depends on the characteristics of the network between the bridges
- The spanning tree is sub-graph of this graph that...
  - has the same set of nodes
  - has not any loop in the topology
- The algorithm *will not* compute the minimum spanning tree
- The algorithm computes the minimum spanning tree given a root node (just like Prim's algorithm)

Phases

1. Select a root bridge (attention! Are bridges nodes or edges?)
2. Determine the least cost path from each bridge/network to the route bridge
   - Attention to ties!
3. Disable unused edges

Port states

A port in the bridge is in one of the following states:

- Blocked
- Root
- Designated

Blocked ports do not forward or receive any data message, but just control messages

Selecting the root bridge

- The system administrator assigns to each bridge a priority
- The bridge with the lowest priority is the root bridge
- In case of same priority, the bridge with the lowest MAC address is selected

Least cost from the bridge to the root

- Every bridge determines the lowest cost path to the root
- Use of the Bellman principle of optimality
- The root port of the bridge is the port connecting the bridge to the root using the lowest cost path

Least cost from the network segment to the root

- All the bridges connected to the same LAN segment decide which has the lowest cost path to the root
- The port connecting the LAN segment to the right bridge becomes the designed port for that segment
- All the ports that are not designed or root become blocked

Breaking ties

Breaking ties for root ports

- Two or more bridges give the same cost path to the root
- The root port is the port connecting to the bridge with the lowest id

Breaking ties for designated ports

- Two or more bridges give the same cost path from a network segment to the root
- The designated port is that of the bridge with the lowest id

What does a bridge know?

- When the bridge are switched on, they don't know the network topology!
- We don't want the system administrator to configure nothing but the priorities of the bridges and the cost of the LAN segments
- Need a control protocol

Bridge Protocol Data Units (BPDUs)

- BPDUs are special frames
- Source address is the source port
- Destination address is the destination port or a special multicast address 01:80:C2:00:00:00:00
- Three types of messages
  1. Configuration (C)BPDU
  2. Topology change notification (TCN)BPDU
  3. Topology change notification acknowledge (TCA)BPDU
- BPDUs are exchanged regularly (default every 2 seconds)

Adding the learning states

A port can then be in one of the following states:

- Blocking
- Listening: The switch processes BDPUs and acquires information about the network topology
- Learning: The switch port does not forward messages yet but starts to acquire addresses
- Forwarding: the port receives and sends data and BDPUs

State of the ports

- Forwarding
- Backup (blocking)
- PRE-Forwarding (not yet forwarding but ready to become)
- PRE-BACKUP (not yet backup but ready to become)

HELLO messages

- HELLO messages are transmitted by the designed bridges to all the other bridges
- They are transmitted regularly
- A HELLO message contains:
  - Transmitting bridge ID *SourceId*
  - ID of the bridge assumed to be root *RootId*
  - Distance from the assumed root *DistanceRoot*
  - Age of the HELLO (time since last information from the root arrived) *Age*
  - Link identifier (local to the transmitter)
  - MAX-AGE

Managing ages of HELLO

- Root node sends HELLO with age 0
- When a designated bridge receives a HELLO it updates its age since last communication from root with the age of the packet:

$$ThisBridge.Age \leftarrow ReceivedHello.Age$$

- $ThisBridge.Age$ is kept updated as time passes
- When $Packet.Age >$ MAX-AGE the HELLO must be discarded and contact to the root is assumed to be lost

Election of the root

- Initially a bridge assumes to be root
  $ThisBridge.Root \leftarrow ThisBridge.id$, $ThisBridge.age \leftarrow 0$
- It broadcast packets with $age \leftarrow 0$, $RoodId \leftarrow ThisBridge.id$, $DistanceRoot \leftarrow 0$
- When a bridge receives a hello packed *Hello* it compares $ThisBridge.root$ with $Hello.Root$:
  - $<$: do nothing
  - $>$: $ThisBridge.root \leftarrow Hello.Root$ and $ThisBridge.DistanceRoot \leftarrow Hello.DistanceRoot + 1$
  - $=$: if $ThisBridge.DistanceRoot > Hello.DistanceRoot + 1$ then $ThisBridge.DistanceRoot \leftarrow Hello.DistanceRoot + 1$

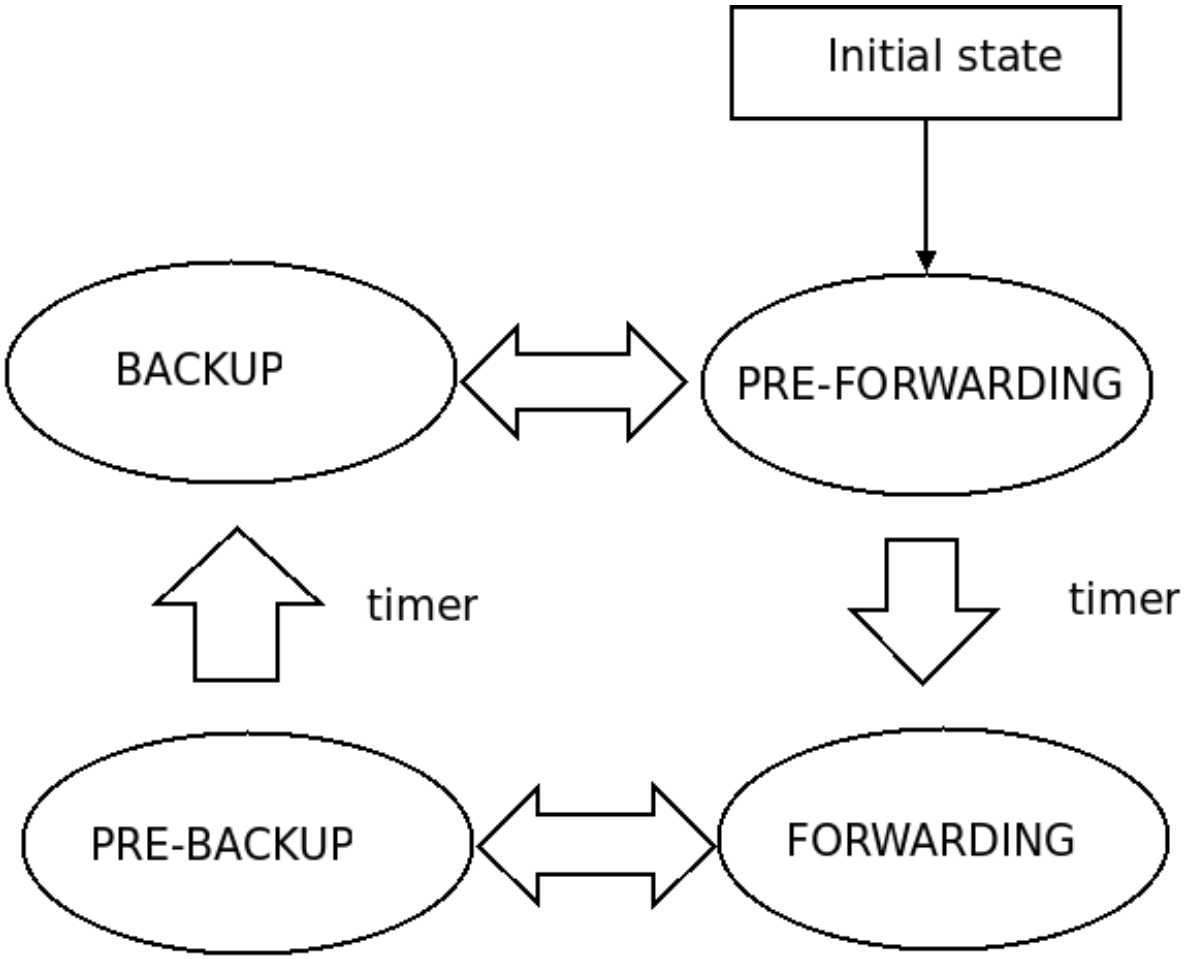
Election of the designated bridge

- A bridge assumes to be the designated bridge on each on its link
- It sends HELLO packets communicating all the other bridges on the link its known root, and the distance to it
- When a bridge receives a HELLO from a LAN segment in which a bridge claims to be closer to the root or with same distance but smaller ID, then the bridge stop sending hello packets on that LAN and renounce to be a designated bridge for that LAN

Using HELLO messages to check the topology

- The root broadcasts hello packets regularly
- They are forwarded by the bridges which also updates they *Age* status
- A designated bridge can send a HELLO in a LAN segment when it receives a HELLO packet
- HELLO in which *Age* is greater than MAX-AGE are discarded so errors in network topology can be identified

Transitions among link states



Transition events for root

- Upon startup a bridge sets all its link to PRE-FORWARDING and claims to be Root and designated bridge on every port
- When the information about root are expired the bridge tries to become root
- Arrival of a HELLO message with a good *Age* and better information about root (a new root, or a better distance)
  - Distance to root is recomputed
  - The bridge may be not anymore a designated bridge for a LAN
- After a delay transitions from PRE-FORWARDING to FORWARDING and from PRE-BACKUP to BACKUP occur

Transition events for non-root for the root-port or designated-port

- The root (designated) port has to be set to FORWARDING
- If the actual state is FORWARDING or PRE-BACKUP it is changed to FORWARDING
- If the actual state is PRE-FORWARDING nothing is done
- If the actual state if BACKUP it is changed to PRE-FORWARDING

Transition events for non-root for the other ports

- The port has to be set to BACKUP
- If the actual state is BACKUP or PRE-FORWARDING it is changed to BACKUP
- If the actual state is PRE-BACKUP nothing is done
- If the actual state is FORWARDING it is changed to PRE-BACKUP

Data traffic

- Data traffic in FORWARDING ports is treated as for standard transparent bridges
- Data traffic in PRE-FORWARDING and PRE-BACKUP is examined only for source addresses
- Data traffic is forwarded in FORWARDING and PRE-BACKUP states
- Data traffic is ignored in BACKUP state

Conclusion

- Allow interconnection of LANs with redundant bridges
- Self-configuring
- Low memory requirements and bandwidth usage
- Minimizes the probability of transient loops
- The algorithms allow the specification of parameters