Lecture 20: Bridging

Prof. Alan Mislove  (amislove@ccs.neu.edu)

Slides used with permissions from Edward W. Knightly,
T. S. Eugene Ng, Ion Stoica, Hui Zhang
Recap

Broadcast technology

- Broadcast network is a simple way to connect hosts
  - Everyone hears everything
- Need MAC protocol to control medium sharing
- Problem: Cannot scale up to connect large number of nodes
  - Too many nodes, too many collisions, goodput (throughput of useful data) goes to zero

Hub emulates a broadcast channel
Easy to add a new host
Building Large LAN Using Bridges

• Bridges connect multiple IEEE 802 LANs at layer 2
  – Datagram packet switching
  – Only forward packets to the right port
  – Reduce collision domain

• In contrast, hubs rebroadcast packets.
Transparent Bridges

• Overall design goal: Complete transparency
  • “Plug-and-play”
  • Self-configuring without hardware or software changes
  • Bridges should not impact operation of existing LANs

• Three parts to transparent bridges:
  (1) Forwarding of Frames
  (2) Learning of Addresses
  (3) Spanning Tree Algorithm
Frame Forwarding

• Each bridge maintains a **forwarding database** with entries 
  \[ \langle \text{MAC address}, \text{port}, \text{age} \rangle \]

  - **MAC address**: host address or group address
  - **port**: port number of bridge
  - **age**: aging time of entry

**interpretation:**

• a machine with **MAC address** lies in direction of the **port** number from the bridge. The entry is **age** time units old.
Frame Forwarding 2

- Assume a frame arrives on port x.

Search if MAC address of destination is listed for ports A, B, or C.

- Found?
  - Forward the frame on the appropriate port

- Not found?
  - Flood the frame, i.e., send the frame on all ports except port x.
In principle, the forwarding database could be set statically (=static routing)

In the 802.1 bridge, the process is made automatic with a simple heuristic:

The source field of a frame that arrives on a port tells which hosts are reachable from this port.
Algorithm:
• For each frame received, stores the source address in the forwarding database together with the port where the frame was received.
• An entry is deleted after some time out (default is 15 seconds).
Example

- Consider the following packets:
  \(<\text{Src}=\text{A, Dest}=\text{F}>, \ <\text{Src}=\text{C, Dest}=\text{A}>, \ <\text{Src}=\text{E, Dest}=\text{C}>\)

- What have the bridges learned?
Danger of Loops
Danger of Loops

- Consider the two LANs that are connected by two bridges.
Danger of Loops

- Consider the two LANs that are connected by two bridges.
- Assume host $n$ is transmitting a frame $F$ with unknown destination.
Danger of Loops

• Consider the two LANs that are connected by two bridges.
• Assume host $n$ is transmitting a frame $F$ with unknown destination.

What is happening?
Danger of Loops

- Consider the two LANs that are connected by two bridges.
- Assume host \( n \) is transmitting a frame \( F \) with unknown destination.

**What is happening?**
- Bridges A and B flood the frame to LAN 2.
• Consider the two LANs that are connected by two bridges.
• Assume host $n$ is transmitting a frame $F$ with unknown destination.

What is happening?
• Bridges A and B flood the frame to LAN 2.
• Bridge B sees $F$ on LAN 2 (with unknown destination), and copies the frame back to LAN 1
Danger of Loops

- Consider the two LANs that are connected by two bridges.
- Assume host $n$ is transmitting a frame $F$ with unknown destination.

**What is happening?**
- Bridges A and B flood the frame to LAN 2.
- Bridge B sees $F$ on LAN 2 (with unknown destination), and copies the frame back to LAN 1.
- Bridge A does the same.
Danger of Loops

- Consider the two LANs that are connected by two bridges.
- Assume *host n* is transmitting a frame *F* with unknown destination.

**What is happening?**
- Bridges A and B flood the frame to LAN 2.
- Bridge B sees *F* on LAN 2 (with unknown destination), and copies the frame back to LAN 1
- Bridge A does the same.
- The copying continues
Consider the two LANs that are connected by two bridges.
Assume host $n$ is transmitting a frame $F$ with unknown destination.

What is happening?
- Bridges A and B flood the frame to LAN 2.
- Bridge B sees $F$ on LAN 2 (with unknown destination), and copies the frame back to LAN 1.
- Bridge A does the same.
- The copying continues

Where’s the problem? What’s the solution?
Spanning Trees

- The solution to the loop problem is to not have loops in the topology.

- IEEE 802.1 has an algorithm that builds and maintains a spanning tree in a dynamic environment.

- Bridges exchange messages (Configuration Bridge Protocol Data Unit (BPDU)) to configure the bridge to build the tree.
What’s a Spanning Tree?

- A subset of edges of a graph that spans all the nodes without creating any cycle (i.e. a tree)
802.1 Spanning Tree Approach (Sketch)

- Elect a bridge to be the root of the tree
- Every bridge finds shortest path to the root
- Union of these paths become the spanning tree
What do the BPDU messages do?

With the help of the BPDUs, bridges can:

• Elect a single bridge as the **root bridge**.
• Calculate the distance of the shortest path to the root bridge
• Each LAN can determine a **designated bridge**, which is the bridge closest to the root. The designated bridge will forward packets towards the root bridge.
• Each bridge can determine a **root port**, the port that gives the best path to the root.
• Select ports to be included in the spanning tree.
Concepts

- Each bridge as a unique identifier:
  
  Bridge ID = <MAC address + priority level>

  Note that a bridge has several MAC addresses (one for each port), but only one ID

- Each port within a bridge has a unique identifier (port ID).

- Root Bridge: The bridge with the lowest identifier is the root of the spanning tree.

- Path Cost: Cost of the least cost path to the root from the port of a transmitting bridge; Assume it is measured in # of hops to the root.

- Root Port: Each bridge has a root port which identifies the next hop from a bridge to the root.
Concepts

- **Root Path Cost**: For each bridge, the cost of the min-cost path to the root

- **Designated Bridge, Designated Port**: Single bridge on a LAN that provides the minimal cost path to the root for this LAN:
  - if two bridges have the same cost, select the one with highest priority (smallest bridge ID)
  - if the min-cost bridge has two or more ports on the LAN, select the port with the lowest identifier

- **Note**: We assume that “cost” of a path is the number of “hops”.
A Bridged Network

Diagram of a bridged network with nodes labeled B1, B2, B3, B4, B5, B6, and B7.
Steps of Spanning Tree Algorithm

1. Determine the root bridge
2. Determine the root port on all other bridges
3. Determine the designated bridge on each LAN

• Each bridge is sending out BPDUs that contain the following information:

| rootID | cost | bridgeID/portID |

root bridge (what the sender thinks it is)
root path cost for sending bridge
Identifies sending bridge
Ordering of Messages

• We can order BPDU messages with the following ordering relation “<” (let’s call it “lower cost”):

If (R1 < R2)
   M1 < M2
elseif ((R1 == R2) and (C1 < C2))
   M1 < M2
elseif ((R1 == R2) and (C1 == C2) and (B1 < B2))
   M1 < M2
else
   M2 < M1
Determine the Root Bridge

- Initially, all bridges assume they are the root bridge.
- Each bridge B sends BPDPUs of this form on its LANs:

  ![BPDU format]

- Each bridge looks at the BPDPUs received on all its ports and its own transmitted BPDPUs.
- Root bridge is the smallest received root ID that has been received so far (Whenever a smaller ID arrives, the root is updated)
At this time: A bridge B has a belief of who the root is, say R.

Bridge B determines the Root Path Cost (Cost) as follows:

- If \( B = R \) : Cost = 0.
- If \( B \neq R \): Cost = \{Smallest Cost in any of BPDUs that were received from R\} + 1

B's root port is the port from which B received the lowest cost path to R (in terms of relation “<“).

Knowing R and Cost, B can generate its BPDU (but will not necessarily send it out):
Calculate the Root Path Cost
Determine the Root Port

• At this time: B has generated its BPDU

<table>
<thead>
<tr>
<th>R</th>
<th>Cost</th>
<th>B</th>
</tr>
</thead>
</table>

• B will send this BPDU on one of its ports, say port x, only if its BPDU is lower (via relation “<“) than any BPDU that B received from port x.

• In this case, B also assumes that it is the designated bridge for the LAN to which the port connects.
Selecting the Ports for the Spanning Tree

- **At this time**: Bridge B has calculated the root, the root path cost, and the designated bridge for each LAN.
- **Now B can decide which ports are in the spanning tree:**
  - B’s root port is part of the spanning tree
  - All ports for which B is the designated bridge are part of the spanning tree.
- B’s ports that are in the spanning tree will forward packets (=forwarding state)
- B’s ports that are not in the spanning tree will not forward packets (=blocking state)
A Bridged Network (End of Spanning Tree Computation)
Ethernet Switches

• Bridges make it possible to increase LAN capacity.
  – Packets are no longer broadcasted - they are only forwarded on selected links
  – Adds a switching flavor to the broadcast LAN

• Ethernet switch is a special case of a bridge: each bridge port is connected to a single host.
  – Can make the link full duplex (really simple protocol!)
  – Simplifies the protocol and hardware used (only two stations on the link) – no longer full CSMA/CD
  – Can have different port speeds on the same switch
    • Unlike in a hub, packets can be stored
Can the Internet be One Big Switched Ethernet?
Can the Internet be One Big Switched Ethernet?

- Inefficient
  - Too much flooding
Can the Internet be One Big Switched Ethernet?

- Inefficient
  - Too much flooding
- Explosion of forwarding table
  - Need to have one entry for every Ethernet address in the world!
- Poor performance
  - Tree topology does not have good load balancing properties
  - Hot spots
Can the Internet be One Big Switched Ethernet?

- Inefficient
  - Too much flooding
- Explosion of forwarding table
  - Need to have one entry for every Ethernet address in the world!
- Poor performance
  - Tree topology does not have good load balancing properties
  - Hot spots
- Etc…