Bridges and LAN Switches

Q. What can be used to share data between two shared-media LAN’s?

A. LAN switch (or bridge) – like a host in promiscuous mode

LANs connected with >= 1 bridge are called

**Extended LANs**

- Problem – what about when node A wants to send node B a message, what happens?

Learning Bridges

- Do not forward when unnecessary
- Maintain forwarding

<table>
<thead>
<tr>
<th>Host</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>Y</td>
<td>2</td>
</tr>
<tr>
<td>X</td>
<td>2</td>
</tr>
</tbody>
</table>

- Learn table entries based on source address
- Table is an optimization; need not be complete
- Always forward broadcast frames
Spanning Tree Algorithm

• Problem: loops

• Bridges run a distributed spanning tree algorithm
  - select which bridges actively forward
  - developed by Radia Perlman
  - now IEEE 802.1 specification

What is a Spanning Tree?

The problem:
1) Tree - is connected graph with no cycles.

2) A Spanning Tree of G is a tree which contains all vertices in G.
   Example: Given a graph G

Is G a Spanning Tree?

Yes

No

Note: Connected graph with n vertices and exactly n – 1 edges is Spanning Tree.
Spanning Tree Example

Example:

\[ G \]

1

\[ \begin{array}{c}
2 \\
3 \\
4 \\
5 \\
6 \\
7 \\
8 \\
\end{array} \]

Centralized Spanning Tree Algorithm

- DFS (Depth First Search)
Centralized Spanning Tree Algorithm

- BFS (Breadth First Search)

Distributed Spanning Tree Algorithm - Overview
- Each bridge has unique id (e.g., B₁, B₂, B₃)
- Select bridge with smallest id as root
- Select bridge on each LAN closest to root as the designated bridge (use id to break ties)
Distributed Spanning Tree Algorithm - Detail

- Bridges exchange configuration messages
  - id for bridge sending the message
  - id for what the sending bridge believes to be root bridge
  - distance (hops) from sending bridge to root bridge
- Each bridge records current “best” configuration message for each port
- Initially, each bridge believes it is the root and sends messages out on all its ports (distance to root = 0)
- When learn not root, stop generating configuration messages
  - in steady state, only root generates configuration messages
- When learn not designated bridge, stop forwarding config messages (disconnected)
  - in steady state, only designated bridges forward config messages
- Root continues to periodically send config messages
- If any bridge does not receive config message after a period of time, it starts generating config messages claiming to be the root

Distributed Spanning Tree Algorithm - Overview

- Ports which are not selected (disconnected) by the Distributed Spanning Tree Algorithm

[Diagram showing network with bridges and ports marked as disconnected.]
Limitations of Bridges

• Do not scale
  – Spanning tree algorithm does not scale (no hierarchy)
  – Broadcast does not scale (congestion)
• Do not accommodate heterogeneity
  (Ethernet-to-Ethernet, but not others such as ATM)
• Caution: beware of transparency
  – If a host is configured for single-LAN use, unexpected results can come about
    • Bridges might drop frames (congestion)
    • Latency differences
    • Frame reordering

Asynchronous Transfer Mode (ATM)

• Connection-oriented packet-switched network
  (virtual circuits)
• Used in both WAN and LAN settings
• Signaling (connection setup) Protocol: Q.2931
• Packets are called cells
  – 5-byte header + 48-byte payload (fixed length)
• Commonly transmitted over SONET
  – Other physical layers possible
Big vs Small Packets

- Small improves Queue behavior
  - Finger-grained pre-emption point for scheduling link
    - Maximum packet = 4 KB
    - Link speed = 100Mbps
    - Transmission time = \(4096 \times \frac{8}{100} = 327.68\mu s\)
    - High priority packet may sit in the queue 327.68 \(\mu s\)
    - In contrast, 53 \(\times \frac{8}{100} = 4.24\ \mu s\) for ATM
  - Near cut-through behavior
    - Two 4KB packets arrive at same time
    - Link idle for 327.68 \(\mu s\) while both arrive
    - At end of 327.68 \(\mu s\), still have 8KB to transmit
    - In contrast, can transmit first cell after 4.24 \(\mu s\)
    - At end of 327.68 \(\mu s\), just over 4KB left in queue

Variable vs Fixed-Length Packets

- No Optimal length
  - If small: high header-to-data overhead
  - If large: low utilization for small messages

- Fixed-Length Easier to Switch in Hardware
  - Simpler
  - Enables parallelism
  - Telephone company supported!
Big vs Small Packets

- Small improves Latency (for voice)
  - Voice digitally encoded at 64Kbps (8-bit samples at 8 KHz)
  - Need full cell’s worth of samples before sending cell
  - Example: 1000-byte cells implies 125ms per cell (too long)
  - Smaller latency implies no need for echo cancellors

- ATM Compromise: 48 bytes = (32 + 64) / 2
  - Excellent case study of standardization
  - U.S. wanted 64-byte cell & Europe wanted 32-byte cell

Cell Format

- User-Network Interface (UNI)

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>3</th>
<th>1</th>
<th>8</th>
<th>384 (48 bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFC</td>
<td>VPI</td>
<td>VCI</td>
<td>Type</td>
<td>CLP</td>
<td>HEC (CRC-8)</td>
<td>Payload</td>
<td></td>
</tr>
</tbody>
</table>

- Host-to-switch format
- GFC: Generic Flow Control – arbitrate access to link
- VCI: Virtual Circuit Identifier
- VPI: Virtual Path Identifier
- Type: management, congestion control, AAL5 (later 0
- CLP: Cell Loss Priority
- HEC: Header Error Check (CRC-8)

- Network-Network Interface (NNI) – between telcos
  - Switch-to-switch format
  - GFC becomes part of VPI field
Segmentation and Reassembly (SAR)

- Really *Fragmentation* and Reassembly
  - High level protocols hand packets down to lower-level protocols with headers added
  - In ATM, Packets often too large
  - Packets are split up, sent on, and reassembled
  - Protocol layer added - ATM adaptation Layer (AAL)
    - Sits between ATM and IP
    - Contains information needed by receiver for reassembly

Segmentation and Reassembly

- Four possible ATM Adaptation Layers (AAL)
  - AAL 1 and 2 designed for applications that need guaranteed rate (e.g., voice, video)
  - AAL 3 / 4 designed for packet data (connectionless)
  - AAL 5 is an alternative standard for packet data