FDDI – Example of a reservation based protocol

- Stations organized in a ring; packets travel one way only
- If you have token, transmit; delayed vs. immediate token release
- Would like to support bursty behavior – timed token algorithm
  - Allow a station to transmit multiple frames under low loads
  - But need a way to regulate periods between seeing the token under high load
- Token maintenance is an issue
- Should be better than Ethernet at high loads?

Reservation

- Try to prevent collisions by coordinating access with neighbors
  - Problem: coordination may be expensive
  - Especially true under low load
- We looked at different ways of reducing the cost of coordination
- Started talking about possible ways to allow contention-like behavior to reservation protocols

Homework due today
Last Time:
- Lots of MAC
- Started switched Networks
Today
- More switched networks
- Bridges/LAN Switches (link level datagram approach)
Discussion/Summary

- What is possible is a function of the physical properties of the link we are trying to share
- Big decision between contention and reservation
  - But sometimes need to be creative so that we address their weaknesses
- Wrap up discussion of issues in directly connected nodes

CSMA/Collision Avoidance (CA)

- Sender Broadcasts Request to Send (RTS)
- Receiver responds with Clear To Send (CTS), including the length of transmission
- Sender sends message if it receives CTS
- Any nodes receiving the CTS do not transmit for the transmission length
- Nodes receiving RTS, but not CTS know they are far enough and can transmit
Switching and Heterogeneity

- Supporting heterogeneous technologies is one of the design objectives of the Internet.
- Different addresses, bandwidths, latencies, loss rates, service guarantees, switching approach.

Virtual Circuit Switching

- Switch translates directly between the different technologies (Bridge, LAN switch, Level 2 switch, or simply switch).
- Use a universal format that abstracts away the technology.

Datagrams

- Connection-Oriented: explicit connection setup and tear-down.
- No connection exists: no connection setup phase.
- Each switch maintains a routing table.
- Connection number (and tear-down) needed.

Connection-Oriented Model - each packet forwarded independently

Connection-Oriented Model - each packet forwarded independently

Switching and Heterogeneity

- Switch has buffers, it operates using store and forward.
- Physical limitations apply to each connection independently.
- Connection set-up/tear-down.

Connection set-up/tear-down.

Switch vs. Repeater

- A switch simply translates the signal it receives.
- A repeater amplifies the signal it receives.
- Input and output must be exactly matched.
- No buffering.

Switching and Heterogeneity
VC Setup

- Need to establish a path to destination
- To create a path from A to B on the example topology:
  - Setup message is sent from A towards B
    - Still need the routing table
  - When an intermediate switch gets the setup message
    - It creates an entry in its “VC table”
    - Forwards the setup message to the next hop
- What should be kept in this table entry?
  - Must the VC number be global? Sounds like a hard problem

Examples

- Internet Protocol (IP) is connectionless (what about TCP?)
- Asynchronous Transfer Mode (ATM) is connection oriented – we will overview later
- X.25 – connection oriented protocol
  - Buffer allocated for each VC
    - Circuit is rejected if not enough buffer space
    - Sliding window is run between hops
- Frame Relay – “X.25 light”, no flow control or reliability guarantees; faster
- Why “virtual” in VC; how is this different from circuit switching in phone company?

Virtual Circuit vs. Datagram

- Virtual Circuit
  - Need to wait a full RTT for connection setup before sending data
  - Connection request contains full address; subsequent data packets contain only circuit identifier
  - A link or switch fails? tough, need to re-establish path
  - Connection setup provides opportunity to reserve resources
- Datagram:
  - No connection setup – can start communication immediately
  - No way to tell if a path exists
  - Packets are routed independently, it is possible to route around failures
  - Every packet must carry full destination information

Source Routing

- Header includes sequence of ports from source to destination
- Not as widely used as the other two models. Useful in certain situations
  - IP includes a source routing option (disabled due to security concerns)
  - Dynamic Source Routing (DSR) is a popular protocol for Wireless Ad hoc networks
  - Used in some high-performance LANs (e.g., myrinet)
Learning Bridges

- Idea: if a bridge sees a frame sent by node X coming on port y, it learns that the way to X is through port y
  - What if it sees a frame destined to X appear on y?
  - What if it sees a frame for X appear on a port z?
  - What if it sees a frame it does not know?
- Dynamically constructs a routing table
- Any problems with this scheme?

Bridges/LAN Switches

- Suppose we want to build an ethernet network bigger than 1024 hosts or using more than 3 populated segments in a row
- Can we use a LAN switch?
  - Also known as bridges
- A bridge/LAN switch that operates on link-layer frames (as opposed to network layer packets)
- Idea of bridge is to transparently extend the LAN
  - Is this another way of saying repeater?

Example

- Example: two ethernet LANs connected by a bridge, compared to two ethernet segments connected by a repeater
  - The bridge appears as an additional host with its own ethernet address on each segment
  - Two communications can be going on the two different segments without collision
  - What if the bridge simply forwards all incoming frames to all outgoing segments?
  - Can forwarding be done in a better way?
  - Network administrator sets up forwarding table?
Problem – Loops

- Problem – the network can contain loops
  - Learning can go wrong
  - Frames can get stuck indefinitely in the network
- What can be done?
  - Rely on administrator to statically eliminate loops?

Spanning Tree Algorithm in IEEE 802.1

IEEE 802.1 is the standard for "LAN/MAN Bridging and Management"

Idea: bridges disable some ports to eliminate cycles

General Algorithm:
- Bridges elect a "tree root" bridge
  - Each bridge calculates shortest path to root
    - Bridges on each LAN elect a "designated bridge" such that
      - It is the closest bridge to the root
      - Break ties using bridge id

Implementation

- All bridges initially assume they are root
- They send out configuration messages on all ports
  - Config. message: (bridge id, distance, root)
- When bridge receives a config message with a better config:
  - It updates its interface information
  - Adds 1 to distance and sends new config. message on all other ports
- A configuration “A” is better than “B” if
  - It identifies a root with a smaller id
  - The root has an equal id, but shorter distance
  - Root and distance are equal but bridge id is smaller
- Bridge only forwards messages on interfaces if the configuration lists them as designated bridge

Example

- B3 receives (2, 0, 2) from 2; accepts 2 as root (2 < 3)
- B3 sends (3, 1, 2) to 5
- B5 receives (1,0,1) from 1 and sends (5,1,1) to 3
- B3 accepts root as 1, realizes B2 and B5 are closer, stops forwarding on both interfaces
Discussion

- Easy to administer, but optimal?
- Bridge ids have to be unique
- What happens if a bridge or link fails?
  - Heartbeat messages/TTL
- What if two LAN segments are different speed? (e.g., 10Mbit ethernet and 100Mbit ethernet)
- What if two segments use different Link protocols (e.g., FDDI and Ethernet)?
- What to do with broadcast and multicast packets?
  - Can bridge learn multicast groups?
- Transparency (an extended LAN appearing as a regular LAN) is dangerous – why?
- Why not build the Internet out of bridges/switched LANs?