Multi-hop Networks

- What’s wrong with just connecting every pair of computers with dedicated links?
- Switches orchestrate flow of information through the network, often multiplexing many logically-independent flows over a single physical link
- Packet switching: model for sharing in most current communication networks
Sharing the Network

We have many application-level communications, which we'll call "connections", that need to be mapped onto a smaller number of links.

How should we share the links between all the connections?

Two approaches possible:

- Circuit switching (isochronous)
- Packet switching (asynchronous)

Circuit Switching

- First establish a circuit between end points
  - E.g., done when you dial a phone number
  - Message propagates from caller toward callee, establishing some state in each switch
- Then, ends send data ("talk") to each other
- After call, tear down (close) circuit
  - Remove state

Multiplexing/Demultiplexing

One sharing technique: time-division multiplexing (TDM)

- Time divided into frames and frames divided into slots
  - Number of slots = number of concurrent conversations
- Relative slot position inside a frame determines which conversation the data belongs to
  - E.g., slot 0 belongs to the red conversation
  - Mapping established during setup, removed at tear down
- Forwarding step at switch: consult table

TDM Shares Link Equally, But Has Limitations

- Suppose link capacity is C bits/sec
- Each communication requires R bits/sec
- #frames in one "epoch" (one frame per communication) = C/R
- Maximum number of concurrent communications is C/R
- What happens if we have more than C/R communications?
- What happens if the communication sends less/more than R bits/sec?
  - Design is unsuitable when traffic arrives in bursts
Circuit-Switching Example: Telephone Network

Packet-Switched Networks

The pioneering research of Paul Baran in the 1960s, who envisioned a communications network that would survive a major enemy attack. The sketch shows three different network topologies described in his RAND Memorandum, "On Distributed Communications: 1. Introduction to Distributed Communications Network" (August 1964). The distributed network structure was judged to offer the best survivability.

Packet Switching

- Used in the Internet
- Data is sent in packets (header contains control info, e.g., source and destination addresses)
- Per-packet forwarding
- At each node the entire packet is received, stored, and then forwarded (store-and-forward networks)
- No capacity is allocated

Simple header example

Destination Address

Hop Limit

Source Address

Length
### IP Version 6 header

<table>
<thead>
<tr>
<th>Version</th>
<th>Traffic Class</th>
<th>Flow Label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Length</th>
<th>Next Header</th>
<th>Hop Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

#### Packet Switching: Multiplexing/Demultiplexing

- Router has a routing table that contains information about which link to use to reach a destination
- For each link, packets are maintained in a queue
  - If queue is full, packets will be dropped
- Demultiplex using information in packet header
  - Header has destination

#### Why Packet Switching Works: Statistical Multiplexing

- Aggregating multiple conversations smooths usage

Traffic in 34-101 wireless LAN during a 6.02 lecture!

Notice how bursts become smoother (but don’t completely disappear)
Queues are Essential in a Packet-Switched Network

- Queues manage packets between arrival and departure
- They are a “necessary evil”
  - Needed to absorb bursts
  - But they add delay by making packets wait until link is available
- So they shouldn’t be too big

Best Effort Delivery Model

No Guarantees!

- No guarantee of delivery at all!
  - Packets get dropped (due to corruption or congestion)
  - Use Acknowledgement/Retransmission protocol to recover
    - How to determine when to retransmit? Timeout?
- Each packet is individually routed
  - May arrive at final destination reordered from the transmit order
- No latency guarantee for delivery
  - Delays through the network vary packet-to-packet
- If packet is retransmitted too soon → duplicate

Sounds like the US Mail!
Four Sources of Delay (Latency) in Networks

- **Propagation delay**
  - Speed-of-signal (light) delay: Time to send 1 (first) bit
- **Processing delay**
  - Time spent by the hosts and switches to process packet (lookup header, compute checksums, etc.)
- **Transmission delay**
  - Time spent sending packet of size S bits over link(s)
  - On a given link of rate R bits/s, transmission delay = \( \frac{S}{R} \) sec
- **Queueing delay**
  - Time spent waiting in queue
  - Variable
  - Whose mean can be calculated from Little’s law

Little’s Law

- P packets are forwarded in time T (assume T large)
- Rate = \( \lambda = \frac{P}{T} \)
- Let \( A = \) area under the n(t) curve from 0 to T
- Mean number of packets in queue = \( N = \frac{A}{T} \)
- A is aggregate delay weighted by each packet’s time in queue.
  - So, mean delay D per packet = \( \frac{A}{P} \)
- Therefore, \( N = \lambda D \) ← Little’s Law
- For a given link rate, increasing queue size increases delay

Circuit v. Packet Switching

<table>
<thead>
<tr>
<th>Circuit switching</th>
<th>Packet Switching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guaranteed rate</td>
<td>No guarantees (best effort)</td>
</tr>
<tr>
<td>Link capacity wasted if data is bursty</td>
<td>More efficient</td>
</tr>
<tr>
<td>Before sending data establishes a path</td>
<td>Send data immediately</td>
</tr>
<tr>
<td>All data in a single flow follow one path</td>
<td>Different packets might follow different paths</td>
</tr>
<tr>
<td>No reordering; constant delay; no dropped packets</td>
<td>Packets may be reordered, delayed, or dropped</td>
</tr>
</tbody>
</table>

Plan for Rest of 6.02

- How to find paths between any two computers? (Routing)
- How can we communicate information reliably? (Transport)