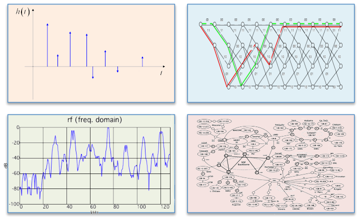


INTRODUCTION TO EECS II
DIGITAL COMMUNICATION SYSTEMS

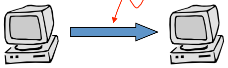
6.02 Spring 2012
Lecture #19

- Multi-hop networks
- Packet switching
- Queues and Little's Law

From Links to Networks

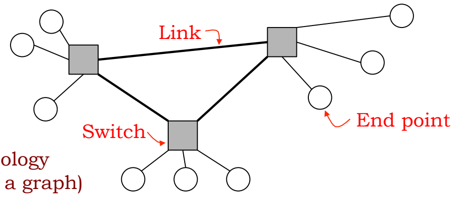
- Have: digital point-to-point

We've studied channel coding, modulation, and MAC protocols



- Want: many interconnected points

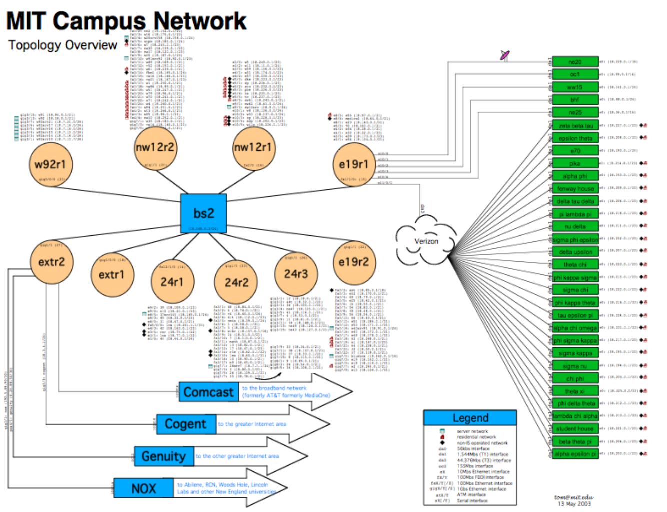
Multi-hop Networks



Network topology
(modeled as a graph)

- 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

MIT Campus Network
Topology Overview



Sharing the Network

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

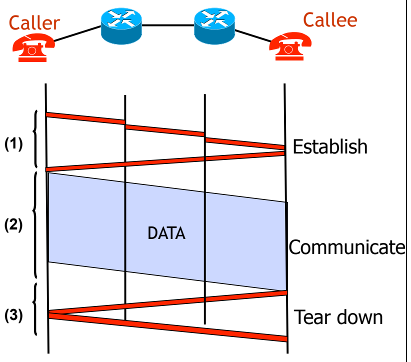
How should we share the links between all the connections?

Two approaches possible:

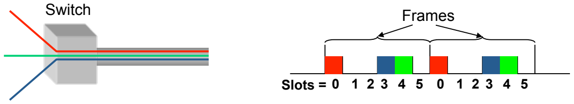
Circuit switching (isochronous)
Packet switching (asynchronous)

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



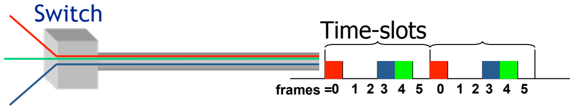
Multiplexing/Demultiplexing



One sharing technique: time-division multiplexing (TDM)

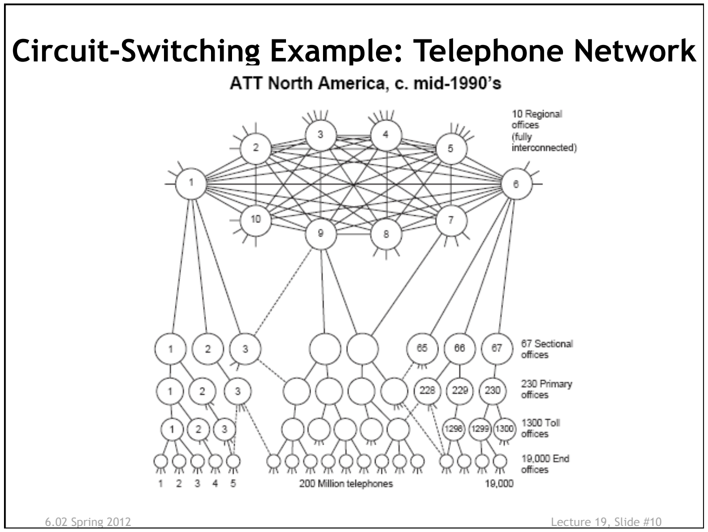
- Time divided into frames and frames divided into slots
  - Number of slots = number of concurrent conversations
- Relative slot position inside a frame determines which conversation the data belongs to
  - E.g., slot 0 belongs to the red conversation
  - Mapping established during setup, removed at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

TDM Shares Link Equally, But Has Limitations



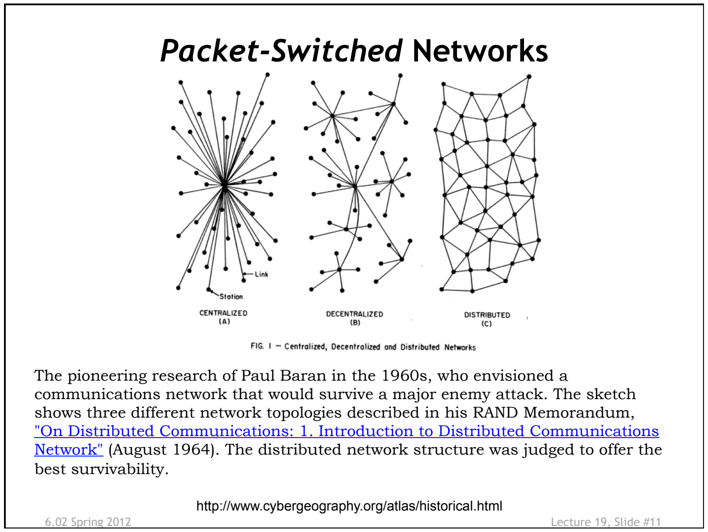
- Suppose link capacity is C bits/sec
- Each communication requires R bits/sec
- #frames in one "epoch" (one frame per communication) = C/R
- Maximum number of concurrent communications is C/R
- What happens if we have more than C/R communications?
- What happens if the communication sends less/more than R bits/sec?

→ Design is unsuitable when traffic arrives in *bursts*



Circuit-Switching Example: Telephone Network

ATT North America, c. mid-1990's

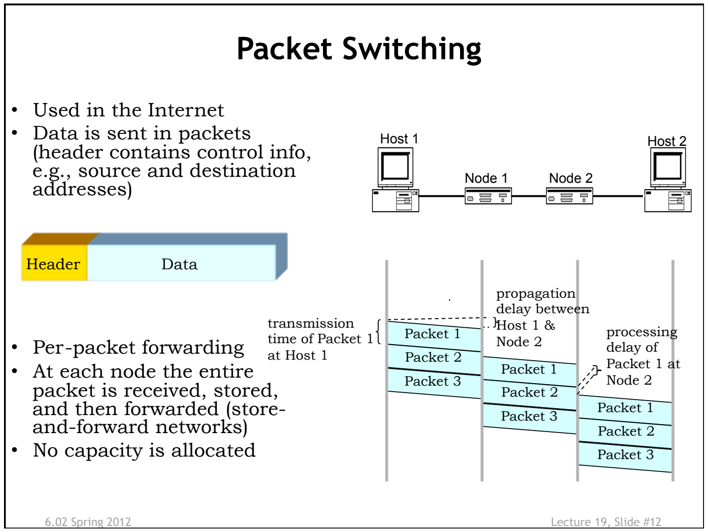


Packet-Switched Networks

FIG. I — Centralized, Decentralized and Distributed 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.

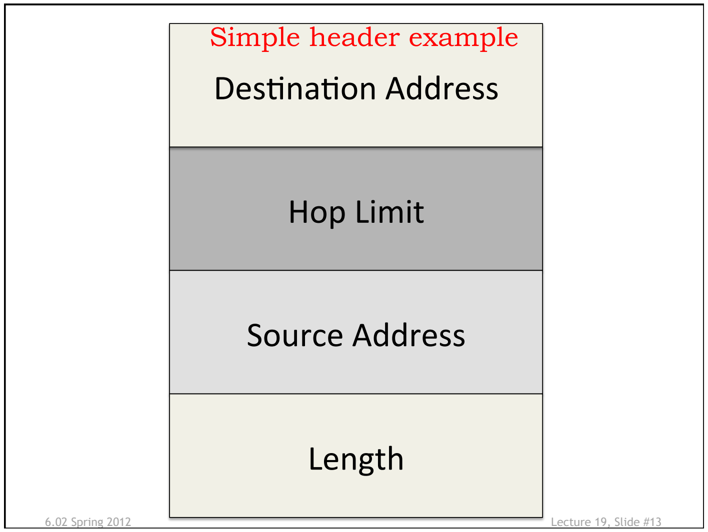
http://www.cybergeography.org/atlas/historical.html



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



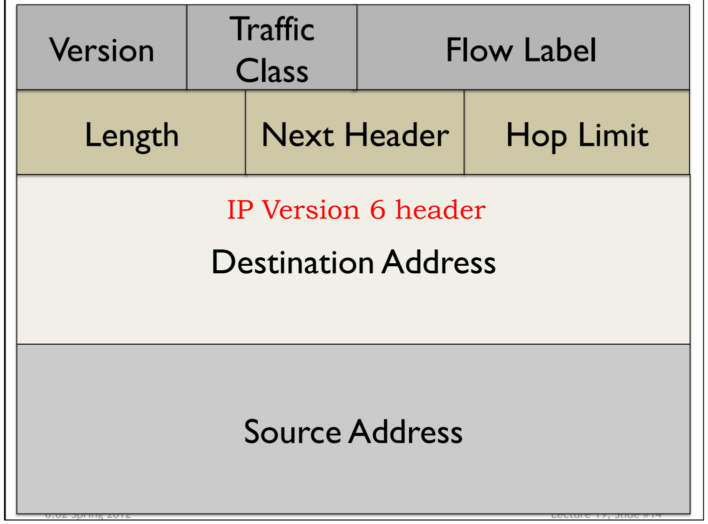
Simple header example

Destination Address

Hop Limit

Source Address

Length



Version | Traffic Class | Flow Label

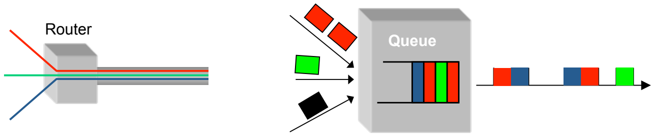
Length | Next Header | Hop Limit

IP Version 6 header

Destination Address

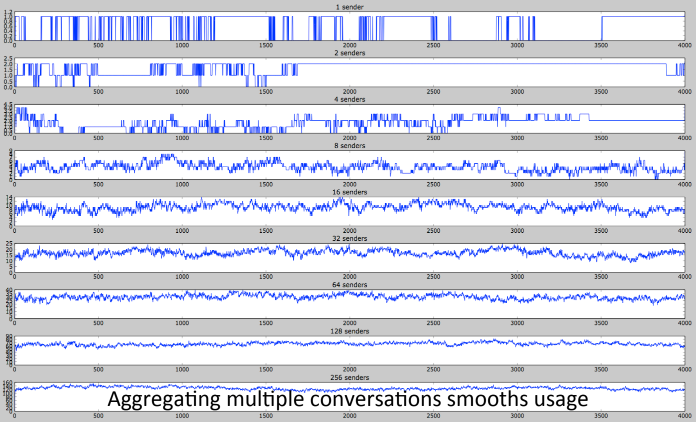
Source Address

Packet Switching: Multiplexing/Demultiplexing

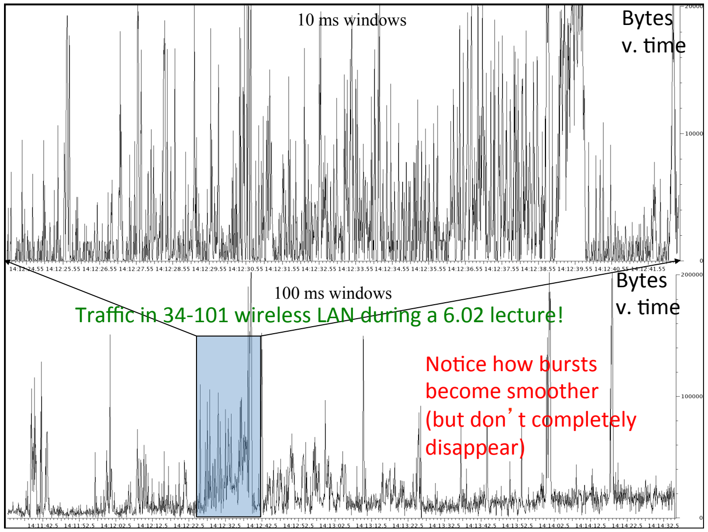


- Router has a routing table that contains information about which link to use to reach a destination
- For each link, packets are maintained in a queue
  - If queue is full, packets will be dropped
- Demultiplex using information in packet header
  - Header has destination

Why Packet Switching Works: Statistical Multiplexing

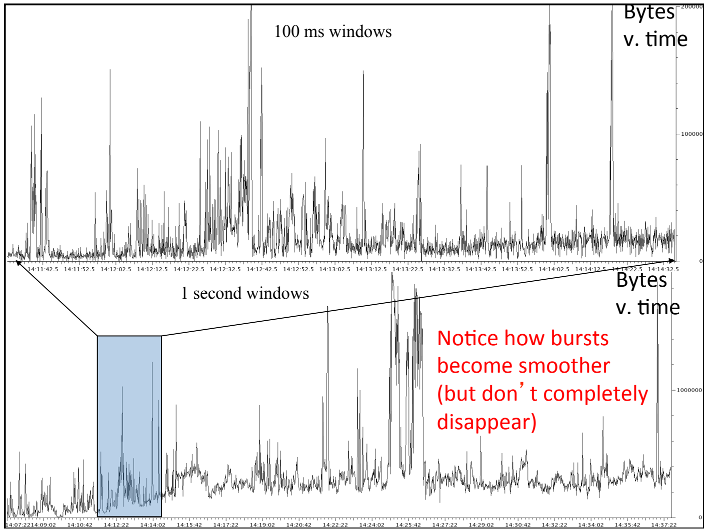


Aggregating multiple conversations smooths usage

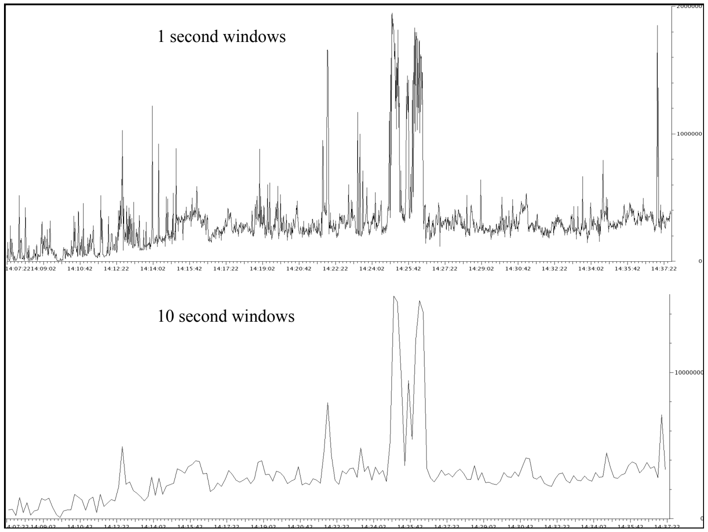


Traffic in 34-101 wireless LAN during a 6.02 lecture!

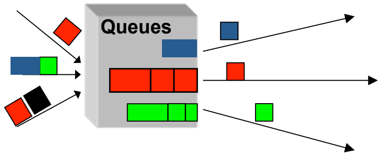
Notice how bursts become smoother (but don't completely disappear)



Notice how bursts become smoother (but don’t completely disappear)



Queues are Essential in a Packet-Switched Network



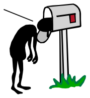
- Queues manage packets between arrival and departure
- They are a “necessary evil”
  - Needed to absorb bursts
  - But they add delay by making packets wait until link is available
- *So they shouldn’t be too big*

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

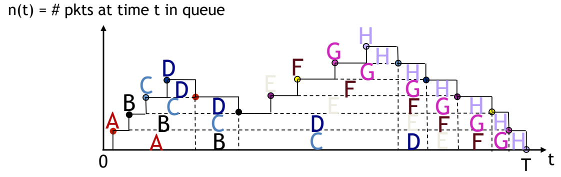
Sounds like the US Mail!



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 = S/R sec
- *Queueing* delay
  - Time spent waiting in queue
  - Variable
  - Whose mean can be calculated from **Little's law**

Little's Law

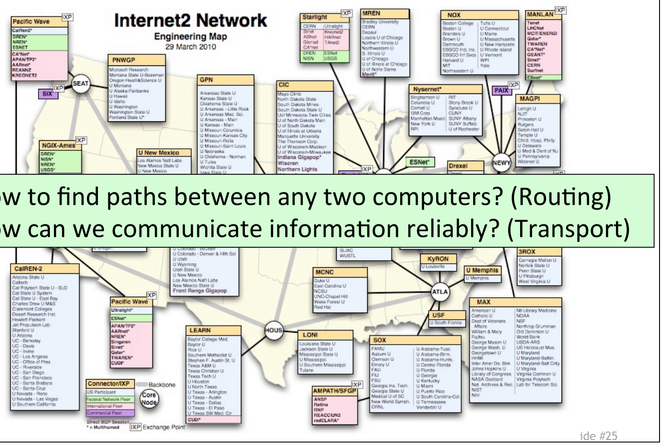


- P packets are forwarded in time T (assume T large)
- Rate = λ = P/T
- Let A = area under the n(t) curve from 0 to T
- Mean number of packets in queue = N = A/T
- A is aggregate delay weighted by each packet's time in queue. So, mean delay D per packet = A/P
- Therefore, **$N = \lambda D$** ← Little's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

Circuit v. Packet Switching

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

Plan for Rest of 6.02



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