From Links to Networks

So far, we’ve learned about tools to help us communicate over a point-to-point link.
How to Build a Network

- Idea: Compose many point-to-point links
  - Simplest approach: link between each pair of nodes

- Why not?
  - Too expensive! How does #links scale?

Solution: Sharing

- Fundamental to how networks are designed
- A switch is a computing device that allows many concurrent communications to share the network

This structure is called the network topology
Muscle-Powered Communications

- Human messengers on foot or horseback
  - “Command and Control” between capital and the field
  - 14 AD: Roman relays—50 miles per day for regular mail, 100 miles per day for express mail
  - 1280 AD: Kublai Khan—200-250 mi per day
  - “Poste Haste”— “Fast Post” —riders signal by horns
“Let us turn now to the system of post-horses by which the Great Khan sends his dispatches. You must know that the city of Khan-balik is a centre from which many roads radiate to many provinces, one to each, and every road bears the name of the province to which it runs. ... When one of the Great Khan’s messengers sets out along any of these roads, he has only to go twenty-five miles and there he finds a posting station, which in their language is called yamb and in our language may be rendered as ‘horse post’. ... Here the messengers find no less that 400 horses, stationed here by the Great Khan’s orders and always kept in readiness for his messengers ...”

“By this means the Great Khan’s messengers travel throughout his dominions and have lodgings and horses fully accoutred for every stage. ... The whole organization is so stupendous and so costly that it baffles speech and writing.”

-- Marco Polo (1290)

[Extracted from “Empire of the Air: The Men Who Made Radio”, by Tom Lewis]

---

An early “switched” network: The optical telegraph

- Chappe (1763-1805), a “defense contractor”; 1st message successfully sent in 1794
- 1799: Napoleon seizes power; sends “Paris is quiet, and the good citizens are content.”
- 1814: Extends from Paris to Belgium & Italy
- 1840: 4000 miles, 556 stations, 8 main lines, 11 sublines, each hop ~10 km
- Many modern techniques: framing, codes, redundant relays, message acks, priority messages, error notification, primitive encryption!
And switches today...

- Alcatel 7670 RSP
- Avici TSR
- Juniper TX8/T640
- Cisco GSR 12416
  - 6ft x 2ft x 1.5ft
  - 4.2 kW power
  - 160 Gb/s cap.
- Lucent 5ESS telephone switch

Two Very Different Ideas for Designing Switched Networks

- Circuit switching
  - Used by telephone networks

- Packet switching
  - Used by the Internet infrastructure
  - (Phone networks also now moving to this model)
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 that was created

Key Idea: Sharing by Multiplexing

- One way for sharing a link is time-division multiplexing (TDM)
  - Divide time into $N$ frame times, each frame belonging to a different conversation (colors in pic)
  - At most $N$ concurrent conversations can share link
  - During set-up, allocate time-slot to conversation
  - Add entry to table mapping time-slot to output link, output_link(time_slot)
- Forwarding step at switch
  - If frame arrives in time-slot $k$, send packet on output_link($k$)
- Other multiplexing schemes: freq. division (FDM), wavelength-division (WDM), code-division (CDM)
TDM Shares Link Equally, But Has Limitations

- Suppose link capacity is $C$ bits/sec
- Each communication requires $R$ bits/sec
- Number of 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?
  $\rightarrow$ Design is unsuitable when traffic arrives in bursts
A Different Approach: Packet Switching

- Data is sent in **packets**
- Each packet contains **control** information in a **header**
  - Destination address
  - Source address
  - Other stuff

- Switch forwards each packet by looking up dest addr in a **forwarding table**
  - Receive, process (lookup), forward

- No reservation of time slot: different communications can get different rates

Packet-Switched Networks Need Queues to Absorb Bursts

- **Multiplex** using a queue
  - Switch uses memory to buffer packets

- **Demultiplex** using information in packet header
  - Header has destination (“datagram header”)
  - Switch has a forwarding table that contains information about which link to use to reach a destination
  - Switches build forwarding tables using **routing protocols**
Why Does Packet Switching Work?

Statistical Multiplexing

When you aggregate bursts, you get some smoothing.
Pic shows time-aggregation; similar for aggregating across many sources.

Exponential is ideal.
Real traffic is Pareto.

Queues

- Queues are a necessary evil
  - Needed to absorb bursts
  - But add delay by making packets wait

- Two important and far-reaching queueing theory results
  - Little’s law
  - Delay as a function of utilization (load/capacity)
    - (In next lecture)
Little’s Law

- Mean number of packets (jobs) in queue = Rate of processing (capacity) * Mean response time per job
- Or, \( N = \text{rate} \times \text{delay} \)
- Or, #stages in pipeline = throughput*latency
- What’s the intuition?

Plan

- Packet-switched networks are “best-effort”: packets dropped, corrupted, delayed, reordered,…
- Layering as a way of coping with “best effort” properties
- Physical layer: already studied (modulation, channel coding)
- Link layer: Framing, channel access (ALOHA) lec+lab
- Network layer: addressing, forwarding, routing protocols (lab)
- End-to-end (transport) layer: reliability (lab)