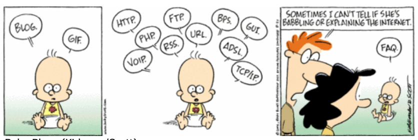
Communication Networks

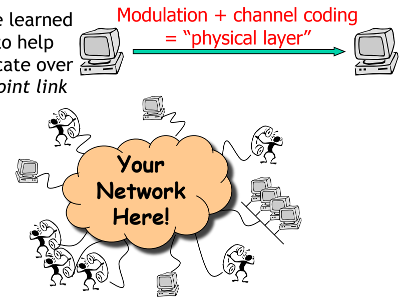
Hari Balakrishnan (hari@mit.edu)
6.02 Spring 2010, Lecture #15
March 31, 2010



Baby Blues (Kirkman/Scott)

From Links to Networks

So far, we've learned about tools to help us communicate over a *point-to-point link*

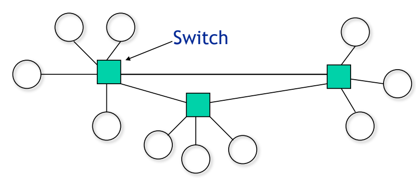


Key Idea: Sharing

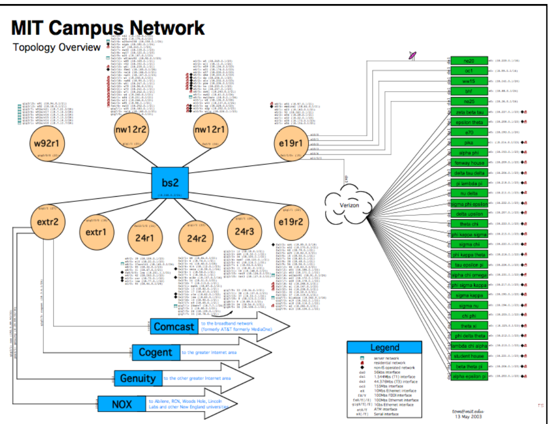
- Fundamental to all communication networks
- Occurs at multiple levels
  - **Link sharing to alleviate $O(N^2)$ scaling problem**
  - Channel sharing to allow many nodes to share common medium (often wireless)

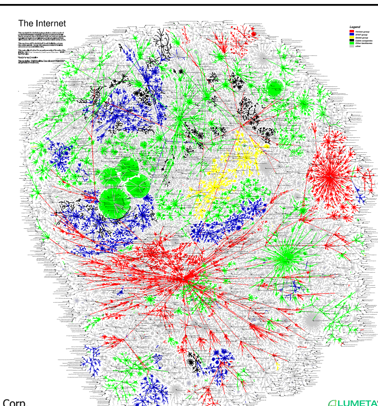
Switches Orchestrate Link Sharing

- A *switch* is a computing device that allows many concurrent communications to share the network



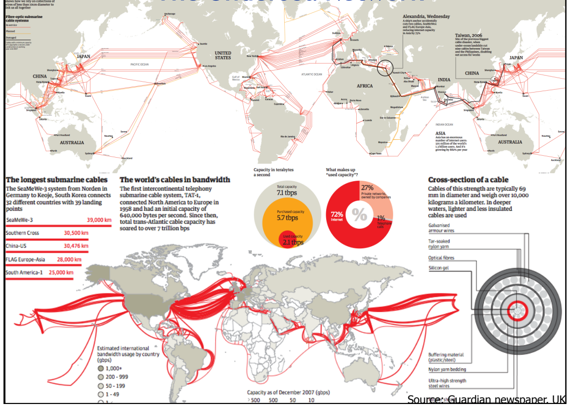
This structure is called a *network topology*





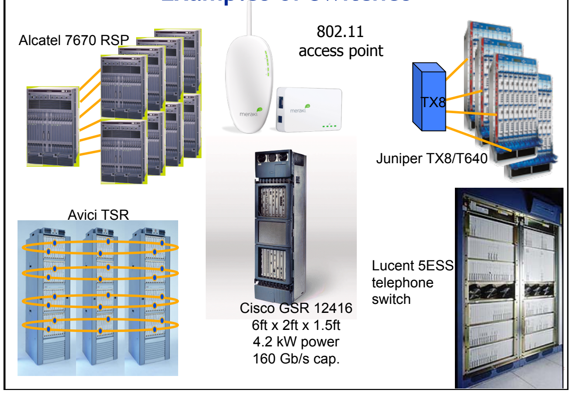
Source: Lumeta Corp.
http://www.saschameinrath.com/files/Internet_map_labels.pdf

The Undersea Network



Source: Guardian newspaper, UK

Examples of Switches

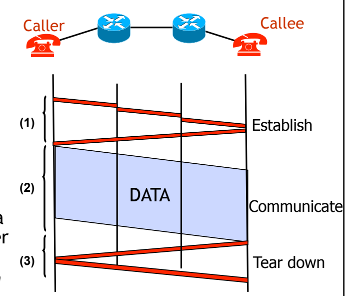


Two Very Different Ideas for Designing Switched Networks

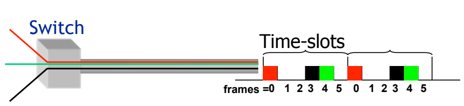
- Circuit switching
  - Used by classic telephone networks
- Packet switching
  - Used by the Internet infrastructure
  - (Phone networks also now moving to this model)

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

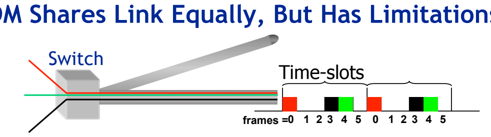


Example: Time Division Multiplexing



- Divide time into *N* frame times, each frame belonging to a different conversation (color)
- At most *N* concurrent conversations share link
- Setup: Allocate time-slot to conversation
- Add entry to table mapping (inlink,time-slot) to (outlink,time_slot)
- Forwarding step at switch: consult table
- When does this approach (not) work?

TDM Shares Link Equally, But Has Limitations

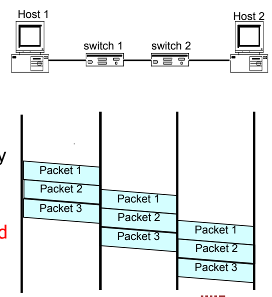


- Suppose link capacity is C bits/sec
- Each communication requires R bits/sec
- #frames in one "epoch" (one frame per communication) = C/R
- Maximum number of concurrent communications is C/R
- What happens if we have more than C/R communications?
- What happens if the communication sends less/more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

→ Design is unsuitable when traffic arrives in *bursts*

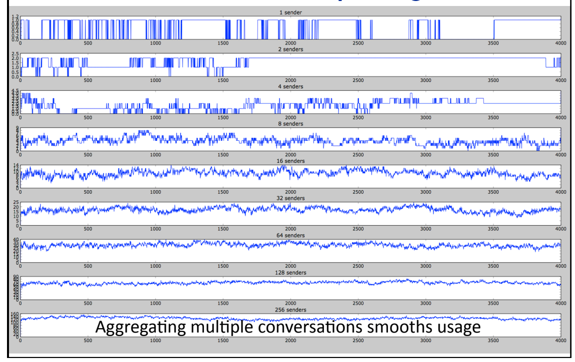
A Different Approach: Packet Switching

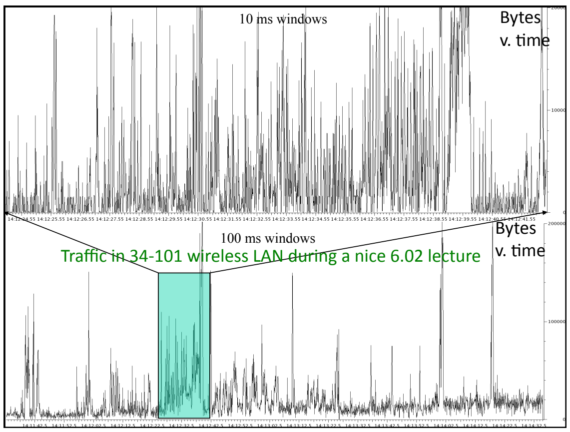
- Data is sent in **packets**
- Each packet contains *control* information in a *header*
  - Destination address
  - Source address
  - Other stuff

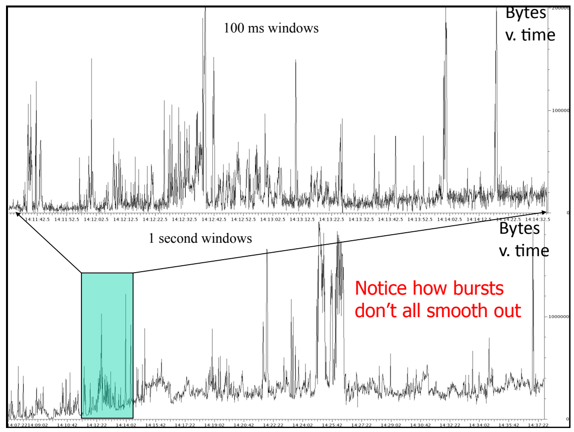


- Switch forwards each packet by looking up dest addr in a *forwarding table*
  - **Receive**, **lookup**, **store in queue** (if link busy), **forward**
- No reservation of time slot: different communications can get different rates

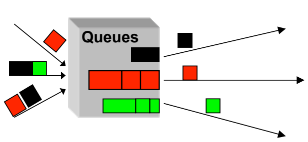
Why Packet Switching Works: Statistical Multiplexing





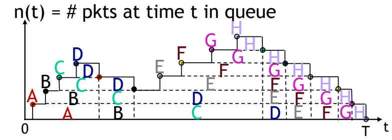


Queues are Essential



Queues absorb packet bursts
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

Little's Law



- Suppose T is large and that P packets are forwarded in that time
- Let A = area under the n(t) curve from 0 to T
- Then, rate = P/T; and mean number of pkts in queue, E[n] = A/T
- How to calculate mean delay per packet?
  - A is aggregate delay weighted by each packet's time in queue (why?)
  - So, mean delay per packet sent = A/P
- Therefore, **E[n] = rate * (mean delay)**
- For a given link rate, increasing queue size increases delay