

CSCI-1680 - Computer Networks

Rodrigo Fonseca (rfonseca)

<http://www.cs.brown.edu/courses/cs168>



Cast

- **Instructor: Rodrigo Fonseca (rfonseca)**
- **HTA: Sungseob Whang**
- **Grad TA: Michael Markovitch**
- **TA: Yi Zhang**
- **How to reach us: Piazza**

<https://piazza.com/brown/fall2017/cs168>



Overview

- **Goal: learn concepts underlying networks**
 - How do networks work? What can one do with them?
 - Gain a basic understanding of the Internet
 - Gain experience writing *protocols*
 - Tools to understand new protocols and applications
- “From 2 communicating machines to the entire Internet”***



Prerequisites

- **CSCI-0330 (or equivalent).**
 - We assume basic OS concepts (kernel/user, threads/processes, I/O, scheduling)
- **Low-level programming or be willing to learn quickly**
 - threads, locking, explicit memory management, ...
- **We allow any* language**
 - No high-level networking APIs, though (unless you write them yourself)
 - You will be bit twiddling and byte packing...



Administrivia

- **All assignments will be on the course page**
<http://www.cs.brown.edu/courses/cs168/f17>
- **Texts (not required):**
 - Peterson and Davie, Computer Networks - A Systems Approach, 4th or 5th editions *or*
 - Kurose and Ross, 'Computer Networking: A Top-Down Approach (6th or 7th editions)
- **You are responsible to check the web page and piazza!**
 - All announcements will be there
 - Textbook chapters corresponding to lectures: read them before class
 - Handouts, due dates, programming resources, *etc...*
 - *Subject to change* (reload before checking assignments)



What do you do?

- **“Written” component**
 - Exams: Midterm (15%) and Final (25%)
 - Homework: 3 written assignments (15%)
 - Short answer and design questions
- **4 Programming Projects (45%)**
 - Snowcast: streaming music server
 - IP, as an overlay, on top of UDP
 - TCP, on top of *your* IP
 - Final (short, fun, to be decided)
- **Must pass two components individually**



Mechanics

- **Content:**
 - Lecture slides are the authoritative content
 - Only what we cover in class will be tested
- **Tools:**
 - Discussions: Piazza
 - HW submission/grading + exam grading: Gradescope
 - Project development and submission: Github
 - Sign up for these (see HW0)
- **Groups**
 - Snowcast is individual, other projects in pairs



Networks

- **What is a network?**
 - System of lines/channels that interconnect
 - *E.g.*, railroad, highway, plumbing, postal, telephone, social, **computer**
- **Computer Network**
 - Moves information
 - Nodes: general-purpose computers (most nodes)
 - Links: wires, fiber optics, EM spectrum, composite...



Why are computer networks cooler?

- **Many nodes are general-purpose computers**
- **Very easy to innovate and develop new uses of the network: *you* can program the nodes**
- **Contrast with the ossified Telephone network:**
 - Can't program most phones
 - Intelligence in the network, control by parties vested in the *status quo*, ...

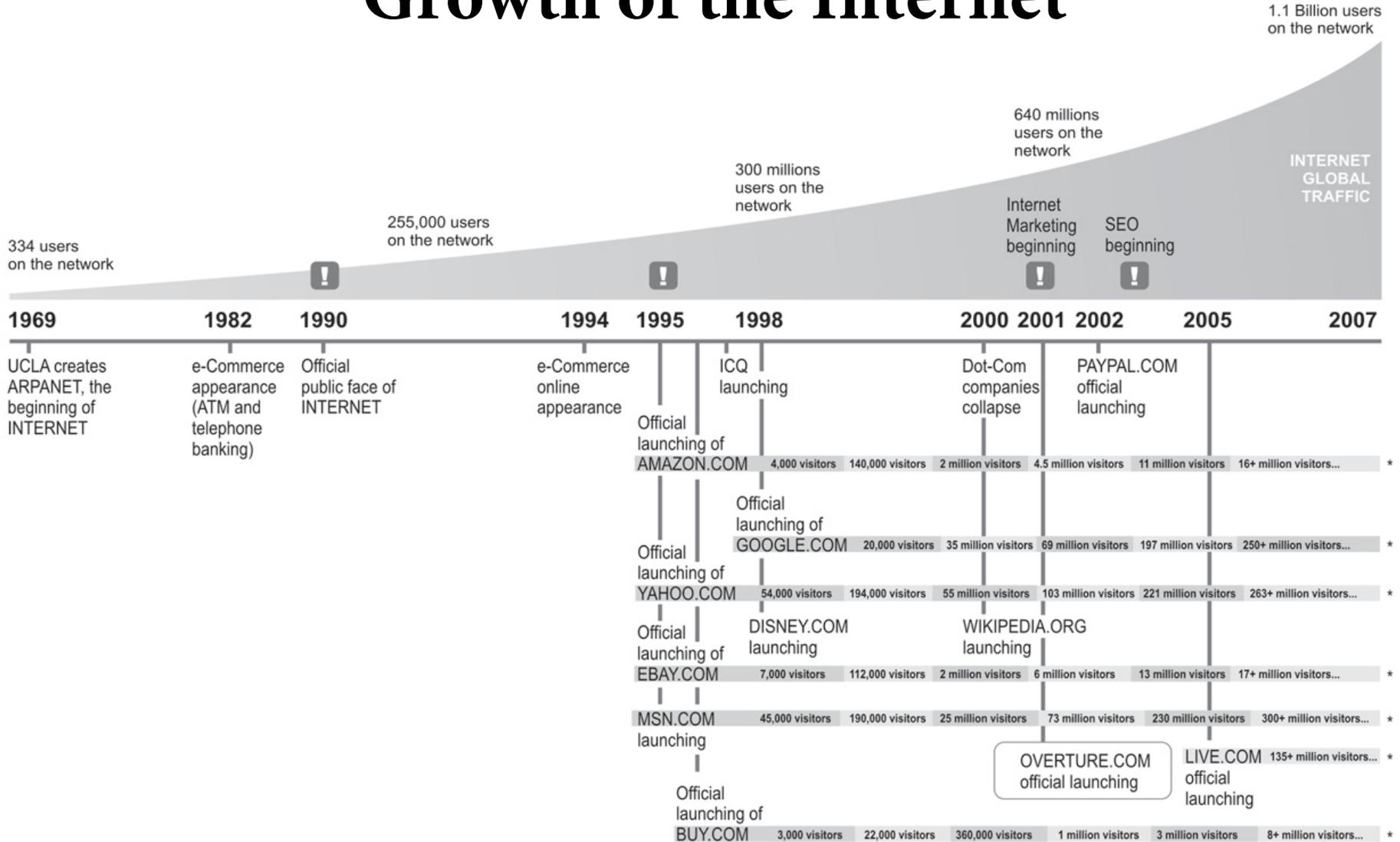


Examples of Innovation

- **WhatsApp: as of Jan 2016, reached 1B monthly active users in 7 years**
 - 57 engineers by then!
- **Uber disrupted transportation**
 - Connectivity allowed a global dispatch service
- **Mirai Botnet (!)**
 - ~50,000 IoT Devices (cameras, DVRs, routers)
 - 100s of Gbps attacks in late 2016



Growth of the Internet



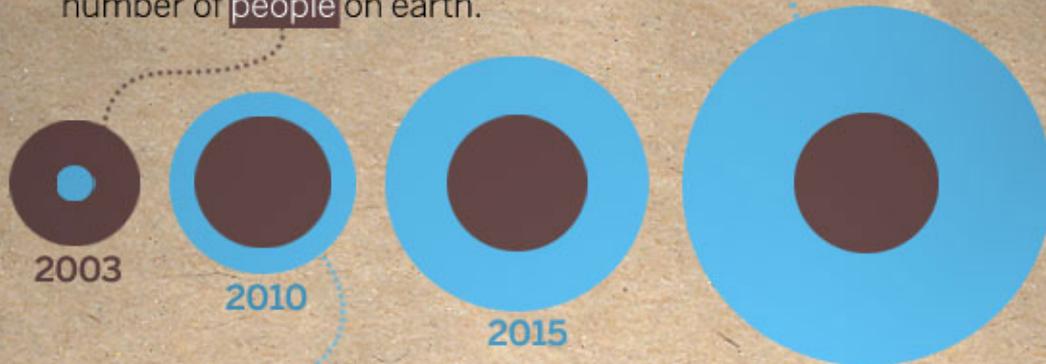
* User traffic calculation per day



INTERNET *of* THINGS



During 2008, the number of **things** connected to the Internet exceeded the number of **people** on earth.



By **2020** there will be **50 billion**.

These **things** are not just smartphones and tablets.

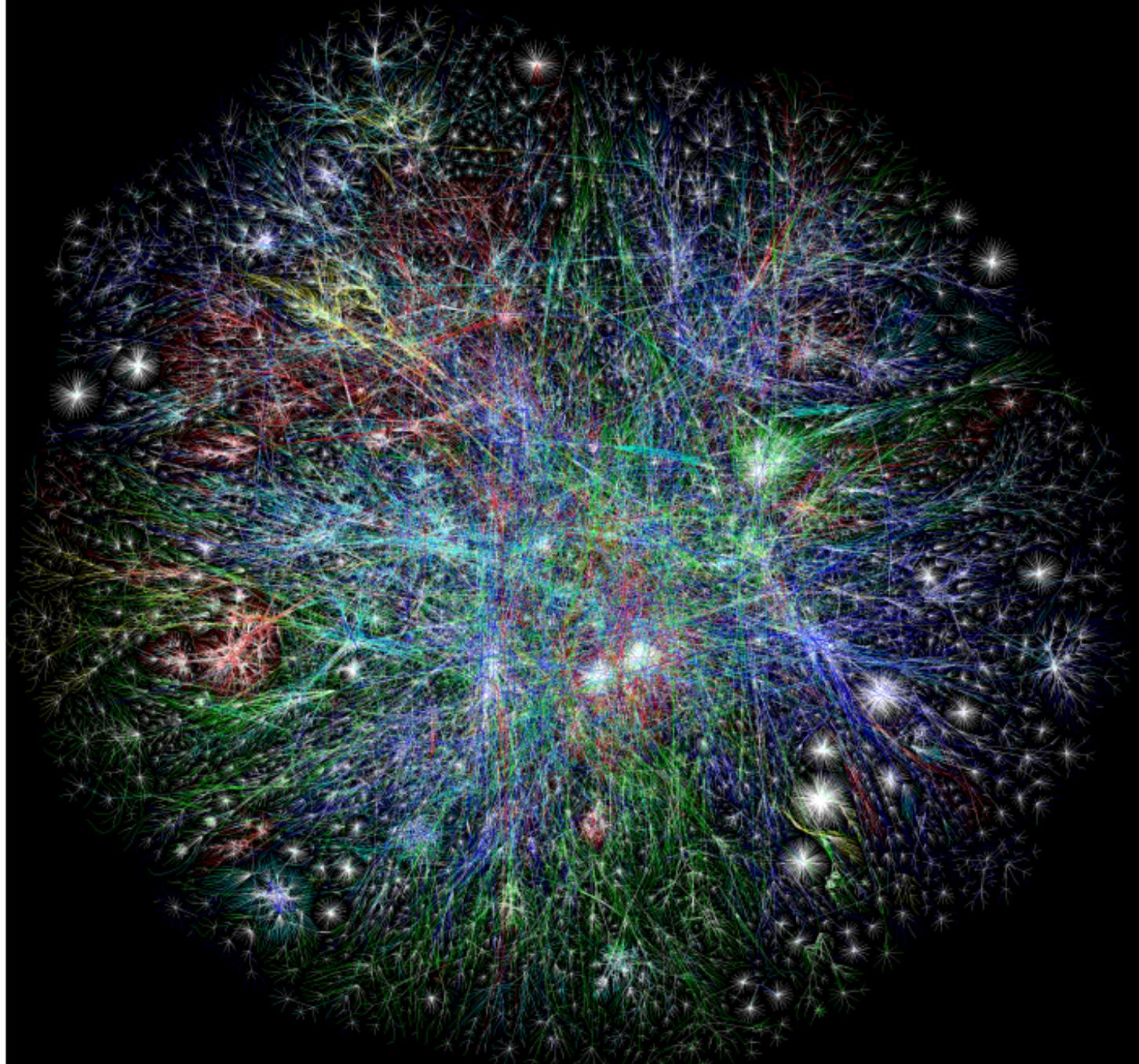
They're every **thing**.

A Dutch startup, **Sparked**, is using wireless sensors on **cattle**.





Source: Facebook



Traceroute map of the Internet, ~5 million edges, circa 2003. opte.org



Why should you take this course?

- **Impact**
 - Social, economic, political, educational, ...
 - Why should you care about NetNeutrality?
 - What does it mean to run out of IP addresses?
 - How could Egypt shut down the Internet internally?
 - How could Pakistan shut down Youtube *globally*?
- **Continuously changing and evolving**
 - Incredible complexity
 - Any *fact* you learn will be inevitably out of date
 - Learn general underlying *principles*
- **Learn to program the network**
- **Networks are cool!**



Roadmap

- **Assignments: learn by implementing**
 - Warm up: Snowcast, a networked music server
 - Get a feel for how applications use the network
- **Build knowledge from the ground up**
 - Link individual nodes
 - Local networks with multiple nodes
 - IP: Connect hosts across several networks
 - Transport: Connect processes on different hosts
 - Applications
- **A few cross-cutting issues**
 - Security, multimedia, overlay networks, P2P...



Recurring Themes

- **How to find who to talk to**
 - Addresses and names, discovery, routing
- **Decide *how* to talk to them**
 - Encodings, Protocols
- **Make sure communication is correct, only among intended parties**
 - Error correction, encryption, ...
- **How to do this at scale**
 - Planetary scale (or beyond)



Stretch

- (and I won't look if you are shopping and want to flee)

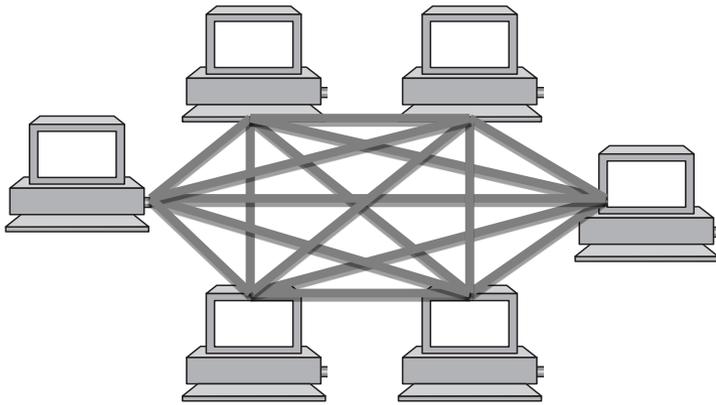


Building Blocks

- **Nodes: Computers (hosts), dedicated routers, ...**
- **Links: Coax, twisted pair, fiber, radio, ...**

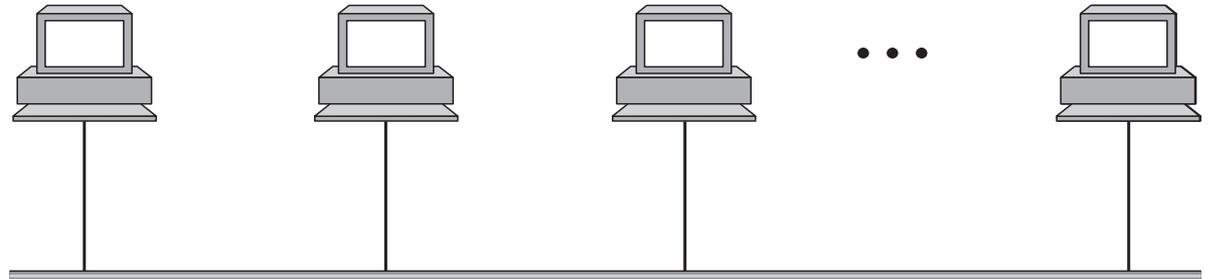


How to connect more nodes?

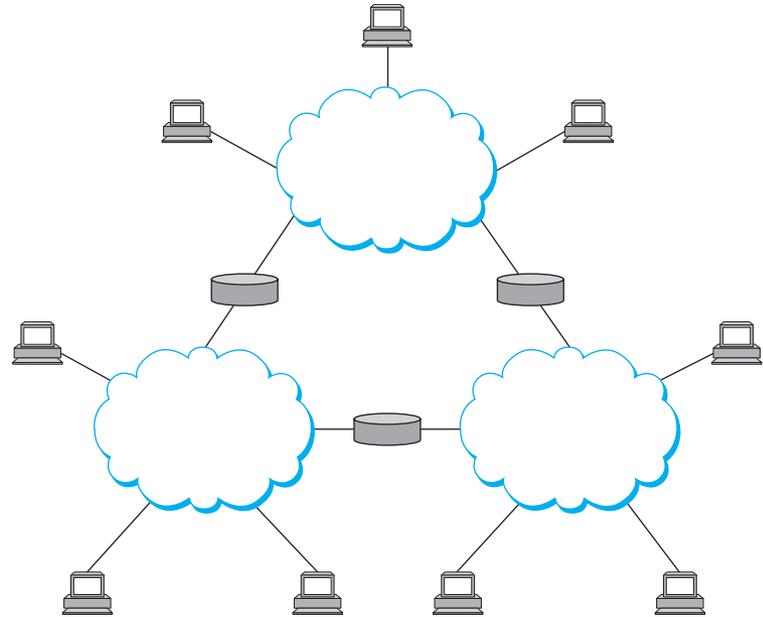
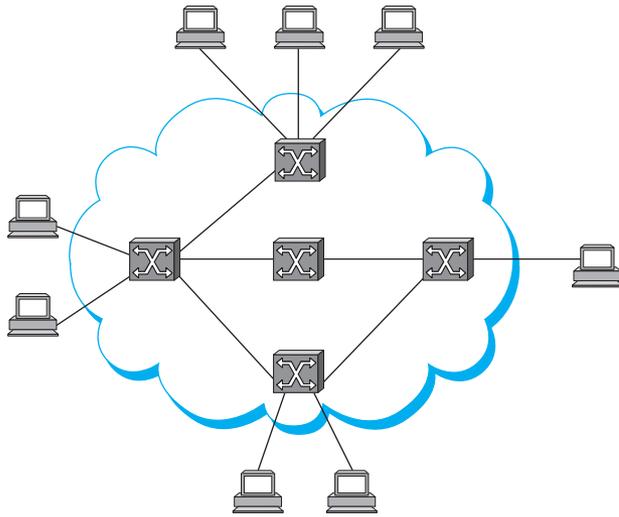


Multiple wires

Shared medium



From Links to Networks



- **To scale to more nodes, use *switching***
 - Nodes can connect to multiple other nodes
 - Recursively, one node can connect to multiple networks



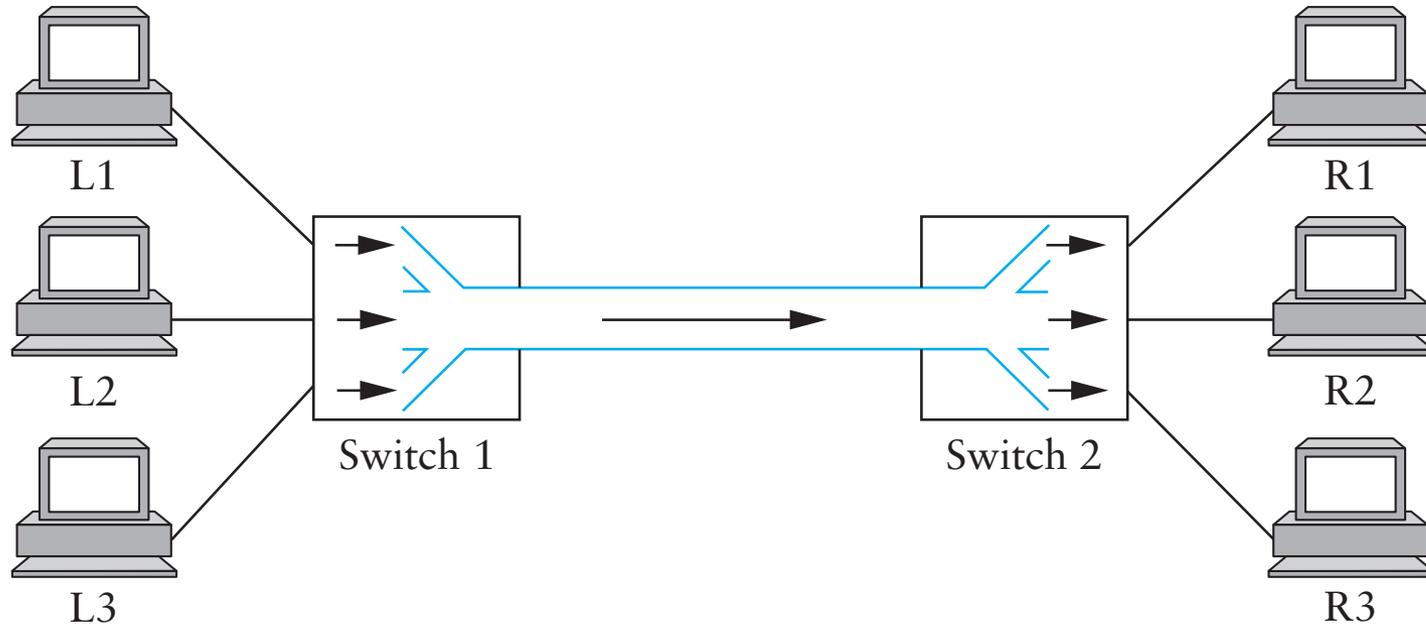
Switching Strategies

- **Circuit Switching – virtual link between two nodes**
 - Set up circuit (e.g. dialing, signaling) – may fail: busy
 - Transfer data at known rate
 - Tear down circuit
- **Packet Switching**
 - Forward bounded-size messages.
 - Each message can have different senders/receivers
 - Focus of this course

Analogy: circuit switching reserves the highway for a cross-country trip. Packet switching interleaves everyone's cars.



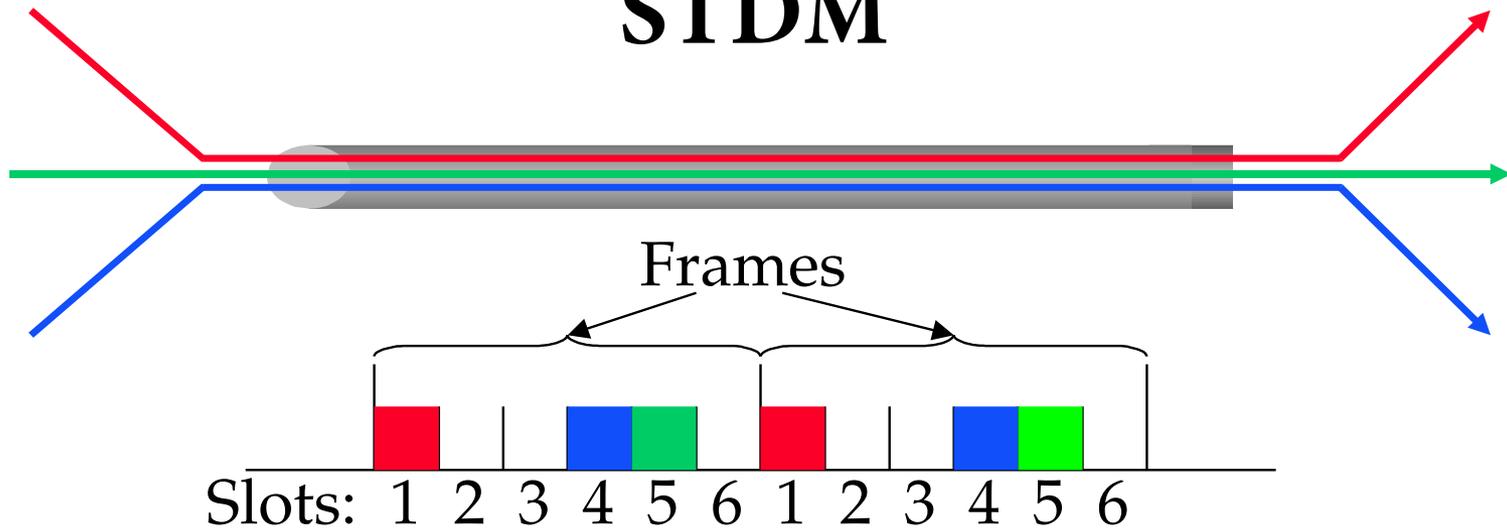
Multiplexing



- **What to do when multiple flows must share a link?**



STDM



- **Synchronous time-division multiplexing**

- Divide time into equal-sized quanta, round robin
- Illusion of direct link for switched circuit net
- But wastes capacity if not enough flows
- Also doesn't degrade gracefully when more flows than slots

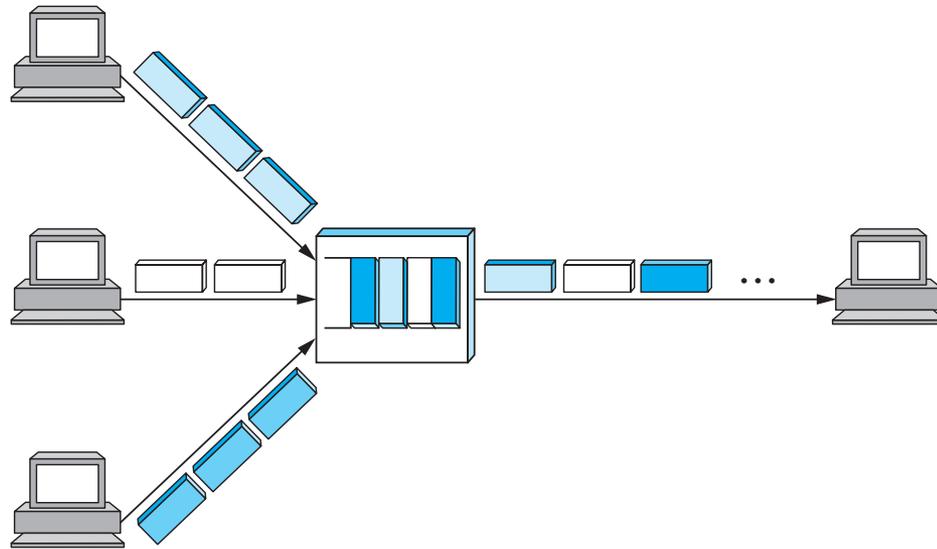


FDM

- **Frequency-division multiplexing: allocates a frequency band for each flow**
 - Same as TV channels and radio stations
- **Similar drawbacks to STDM**
 - Wastes bandwidth if someone not sending
 - Can run out of spectrum



Statistical Multiplexing



- **Idea: like STDM but with no pre-determined time slots (or order!)**
- **Maximizes link utilization**
 - Link is never idle if there are packets to send



Statistical Multiplexing

- **Cons:**
 - Hard to guarantee fairness
 - Unpredictable queuing delays
 - Packets may take different paths
- **Yet...**
 - This is the main model used on the Internet
- **Think of running a restaurant**
 - For a fixed set of people that go there every day
 - Or on a busy corner of Manhattan
 - When would you take reservations?



Roadmap

- **Assignments: learn by implementing**
 - Warm up: Snowcast, a networked music server
 - Get a feel for how applications use the network
- **Build knowledge from the ground up**
 - Link individual nodes
 - Local networks with multiple nodes
 - IP: Connect hosts across several networks
 - Transport: Connect processes on different hosts
 - Applications
- **A few cross-cutting issues**
 - Security, multimedia, overlay networks, P2P...



Coming Up

- **Snowcast: start TODAY!**
- **Next class:**
 - More on layering
 - How to use the network from the application: sockets
- **Then...**
 - We start moving up the network stack, starting from how two computers can talk to each other.

