6.033 2010 Lecture 9: Networking Introduction 3/3/2010

Based on notes from Robert Morris

First of 6 lectures on data networks.

Overview today: identify the problems and design space
Dig into details over next five lectures.
* network are a useful systems building block
* internal workings are a good case study of system design.
  Internet in particular an example of a very successful system.
  complex enough to be a subject of science, like the weather

Seen networking in 6.02.
You've seen:
- Link protocols
- Channel access protocols
- Link-state and DV Routing
- Sliding windows & reliability (RTT, etc)

About networks in the small -- this course is about large networks like the Internet.

Will review enough so that those of you who haven't taken 6.02 still understand how networks work.

Why build networks?
A key use of computers is for communication between people, orgs.
People are geographically distributed, along with their computers.
So we're forced to deal with inter-computer comms.

Allow us to build systems out of multiple computers.
Your problem may be too big for one computer.

You may want multiple computers for structural reasons: client/server.

Set of deep problems that have to be solved in order to build a network at the scale of the Internet. Let's try to understand these.

Universality (challenge -- common protocols)

Early days of networking -- each business set up its own little network.

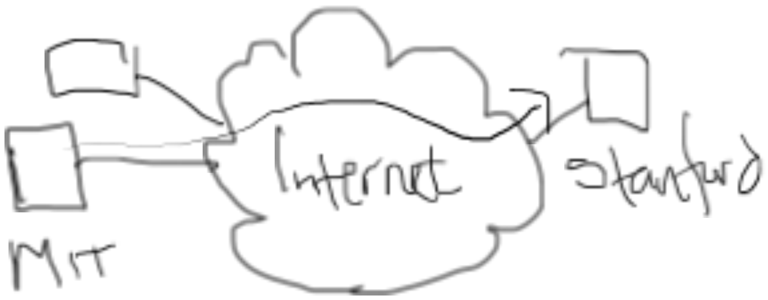
This is useful and has some nice features -- e.g., privacy & security, predictability, reliability.

But more value to having more people connect.
The more people that use a network, the more useful it is for everybody.
Value to me is proportional to N people using the same net.
Value to society is proportional to N^2.

Technical challenges to universality:
- everyone has to use the same protocols
    - tension with allowing evolution
    - hard to standardize whats required, not standardize stuff that is changing

Add'l advantages to universality:
- don't have to design it each time, or understand how it works -- it's just a black box.

Cloud picture:



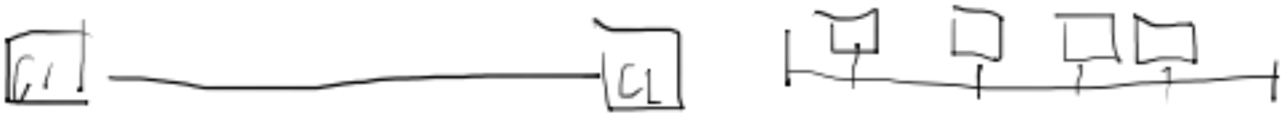
TCP/IP -- intergalactic standard

We will learn how these standard protocols work.

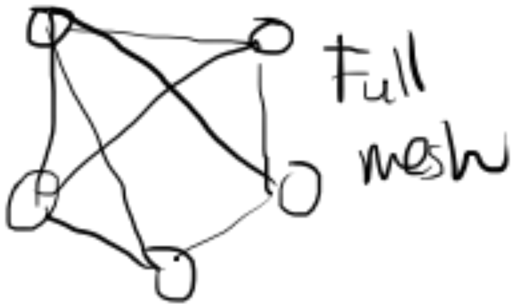
<u>Large scale routing</u>

Simplest network -- two computers on a wire; everybody hears everything
Will learn about how Ethernet -- one common protocol for getting two computers to talk -- works
next time in recitation

More complex networks -- N computers on a wire

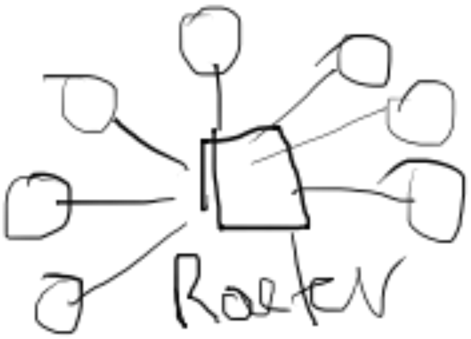


Eventually going to need more than 1 wire -- show full mesh

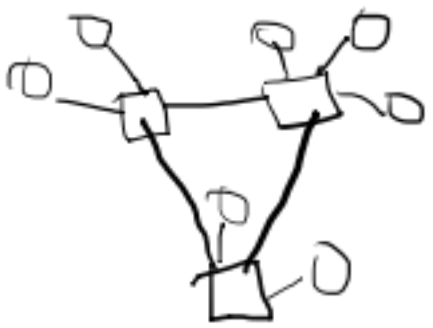


N^2 wires, but everyone can talk to everyone; even better -- lots of redundancy

Star



Lots of wires, central thing is very expensive, centralized point of failure



Partial mesh

Pros -- less expensive, simpler routers, some redundancy, autonomy (didn't suddenly appear one day)

Cons -- complex routing, longer paths, shared links

(Really the only way to build something at the scale of the Internet)

Autonomy --

Internet wasn't designed from the top down -- instead it consists of a bunch of smaller networks that have all grown up autonomously. (Show slides.)

Each network knows how to route between its own machines. No central organizational structure.

Each of this little networks is known as an **Autonomous System**.

Because each network is autonomous, need to be able to add/remove machines within that network without telling everyone on the Internet about it.

===> topological addressing (hierarchy)

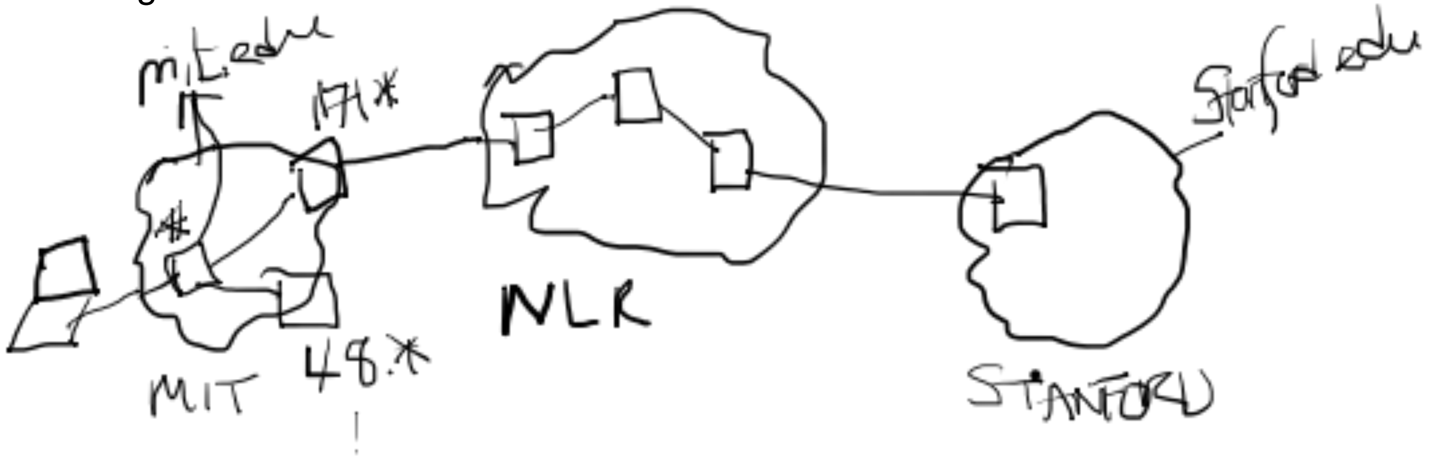
Each machine is identified by an address. (Called an IP address on the internet). Form is A.B.C.D.

Each network is assigned one or more subsets of this address space. For example, all 18.X.X.X addresses belong to MIT.

Suppose we want to send a message to the machine Stanford.edu, which has address 171.67.216.14. (Will talk later about how we convert Stanford.edu into that address.)

Laptop doesn't have a direct connection to that machine, and doesn't even know where it is located, so it just sends message up to MIT, because MIT says it knows how to get to *.*.*.* MIT doesn't know about that address either, so it sends it up to one of it's external networks that says it knows how to get to 171.*. Process repeats, until eventually reaches stanford.

Diagram:



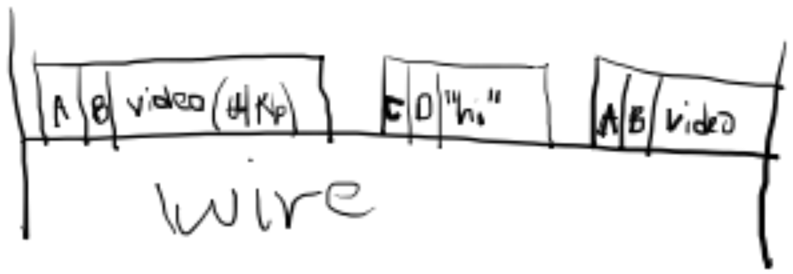
Show traceroute.

Shared links & congestion

Lots of different applications share links. Your youtube videos are streamed over the same wires as my email and bank transfer requests.

All messages on the Internet sent using the "Internet Protocol". Will study in more detail, but essentially carves all communication up to in a sequence of variable length "packets" up to some maximum size. Asynchronous -- no reservations -- any host/app can send whenever it wants.

Show wire with different sized packets from different apps on top of it.



Links have a maximum throughput, determined by the speed of the router and the speed of light (number of bits you can pack onto the wire.) Possible that a link can become "congested", meaning that it queues form at routers and eventually overflow, requiring data to be dropped.

"Congestion control" is an essential part of the Internet that makes it work in the presence of these saturated links.

Will discuss more.

Reliability

The Internet is "best effort" -- messages can be lost or corrupted or re-ordered many places along the way.

Why? Collisions on the wire (two machines try to send at the same time)
Overload (router has too much traffic)
Power failures
Links cut
Noise on the wire

But what if we want to build an RPC system, or Voice chat, or a banking application -- important that data actually gets through!

We will spent quite a bit of time talking about how we can make something reliable out of these best effort links.

Heterogeneity

In applications, data rates, computational power, etc.

Apps have different needs. Clients handle data at different rates. Pretty amazing that it all works. We'll see a bit about how this is all handled.

Internet in design space

Asynchronous (no reservations)
Packets (no connections)
No b/w or delay guarantees
May drop, duplicate, re-order, or corrupt packets -- and often does
Not of much direct use! End hosts must fix
  but gives host flexibility, room for innovation
"best effort"

Networking themes (from 6.02)

- Sharing
- Reliability
- Scalability

Next lecture: start to talk about solutions.