CSE/EE 461

Computer Networks

Brian Bershad

Bershad@cs.washington.edu

Administration

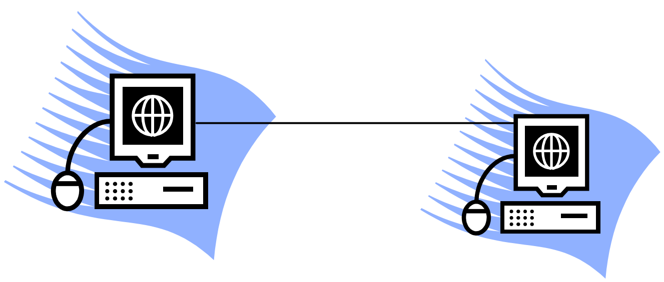
- Generally, check the web site
- Grading
  - Final: 40%
  - Midterm: 25%
  - Homework: 30%
  - Quizzes: 5%
- Policies
  - Late policy: Don't be late.
  - Partner policy: Work alone.
  - Cheating policy: Don't cheat.

This Week

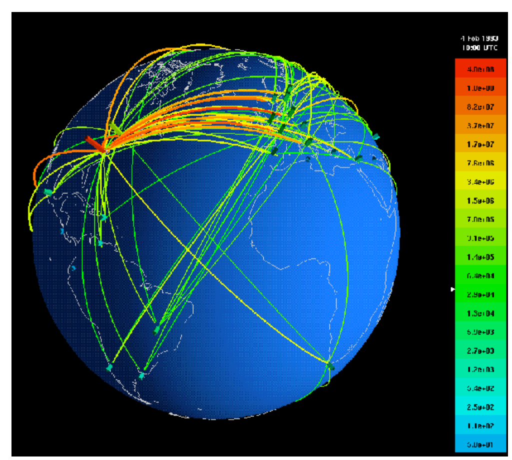
- Readings
  - CN:Chapter 1, Chapter 2 through end of 2.5
- HW 1 due Thurs
- HW 2 out Wed
  - Mostly based on readings
  - Due Fri Jan 20
  - After that, we go to a Fri out/Fri in schedule
- Topics
  - Network principles (today)
  - Elements of network protocols (wed)
  - Reliable transmission (fri)

A Computer Communications Network

- “Network” is clearly an overloaded word:
  - Economic networks, regulatory networks, social networks…
  - Telephone, Cable TV, Bank tellers, computer clusters
- For 461, a network is what you get anytime you connect two or more computers together by some kind of a link.



OR



Essential terms

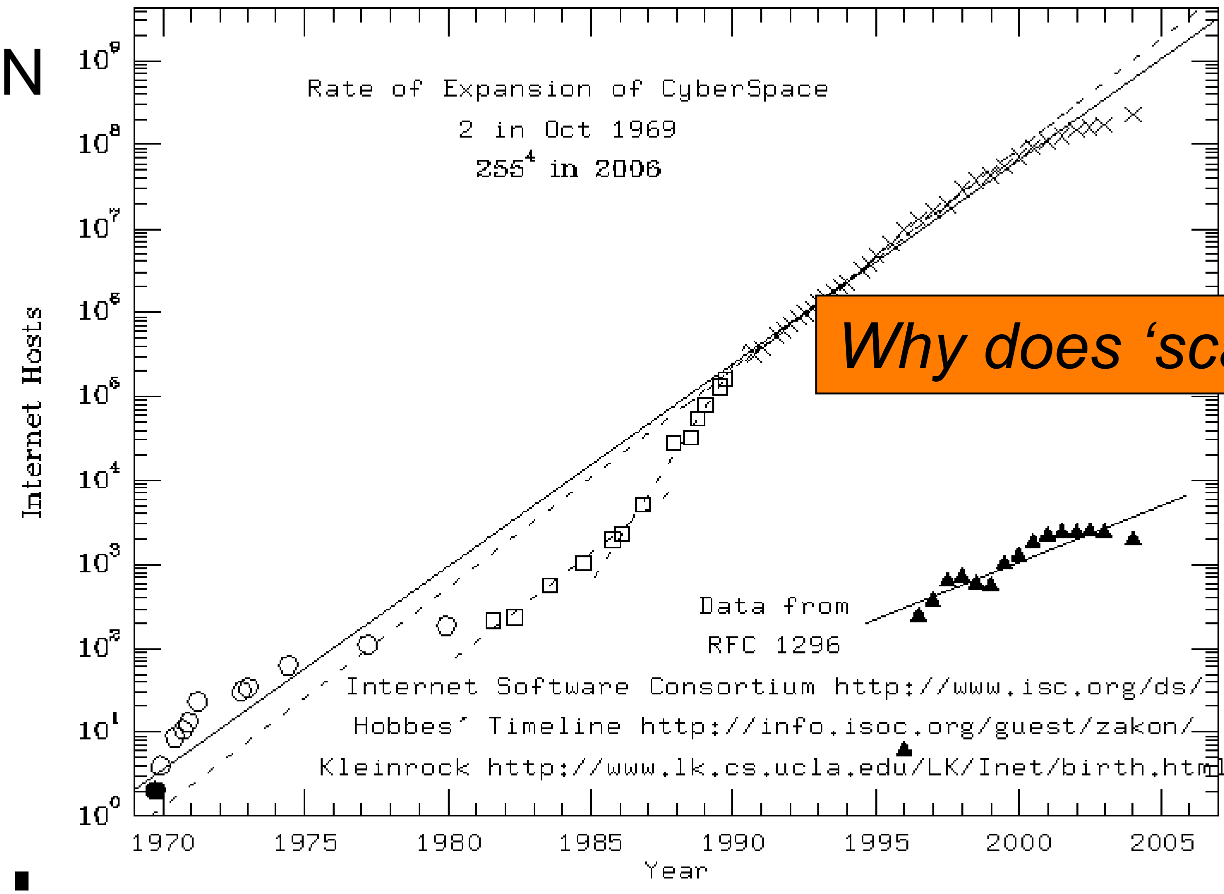
- Node
  - Bandwidth
- Link
  - Aggregate Bandwidth
  - Latency
  - Round-trip time
- Switch/Router
  - Bandwidth
  - Aggregate (bisection) Bandwidth
- Protocol
  - Message syntax and semantics
- Throughput
  - BW achieved by a protocol

The networks we study

- We are interested in networks that are:
  - Large scale
  - Intrinisically Unreliable
  - Distributed
  - Heterogeneous

The meaning of “Large-scale”

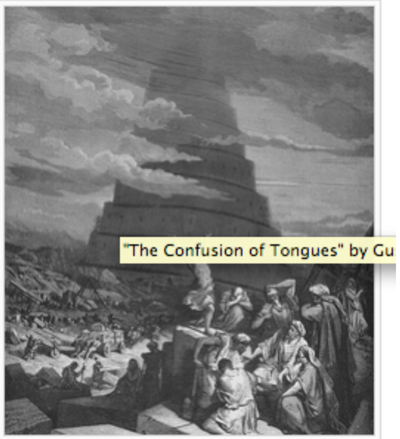
1 BILLION



1

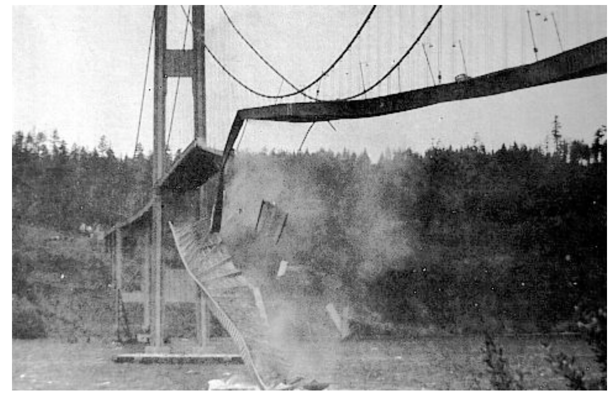
Why does ‘scale’ matter?

Some Famous Non Scalable Artifacts



"The Confusion of Tongues" by Gustave Doré

Icarus and Daedalus



A basic design concept goes “out of whack” relative to some other one

Intrinsic Unreliability

- Information sent from a first place to a second
  - May not arrive
  - May arrive more than once.
  - May arrive in garbled fashion
  - May arrive before other subsequently sent information from that same first place
  - May arrive after other previously sent information sent from that same first place
  - May be read by others
  - May be written by others
- Why build intrinsically unreliable networks?

Distributed

"A distributed system is a system in which I can't do my work because some computer has failed that I've never even heard of." – Lamport

- (Hopefully) independent failure modes
- Exposed and hidden dependencies
- Independent administrative controls

Heterogeneity

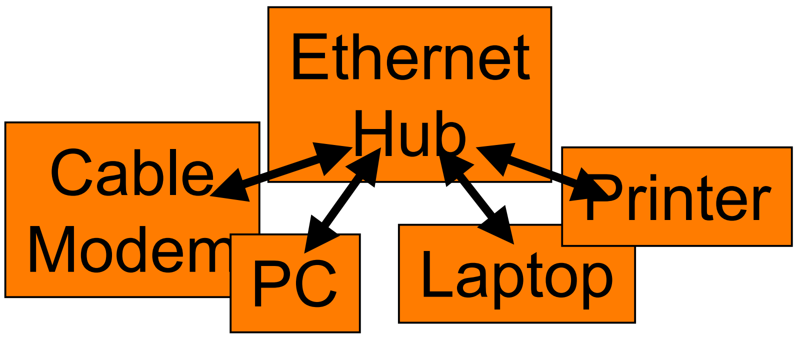
- Heterogeneous: Made up of different kinds of stuff
  - Homogeneous: Made up of the same kind of stuff
- Principle 1: Homogeneous networks are easier to deal with, but more difficult to scale
  - Consider telephone network vs Internet
- Principle 2: Heterogeneous networks lead to greater innovation
  - Consider telephone network vs Internet
  - Reasons?

Model of a Network

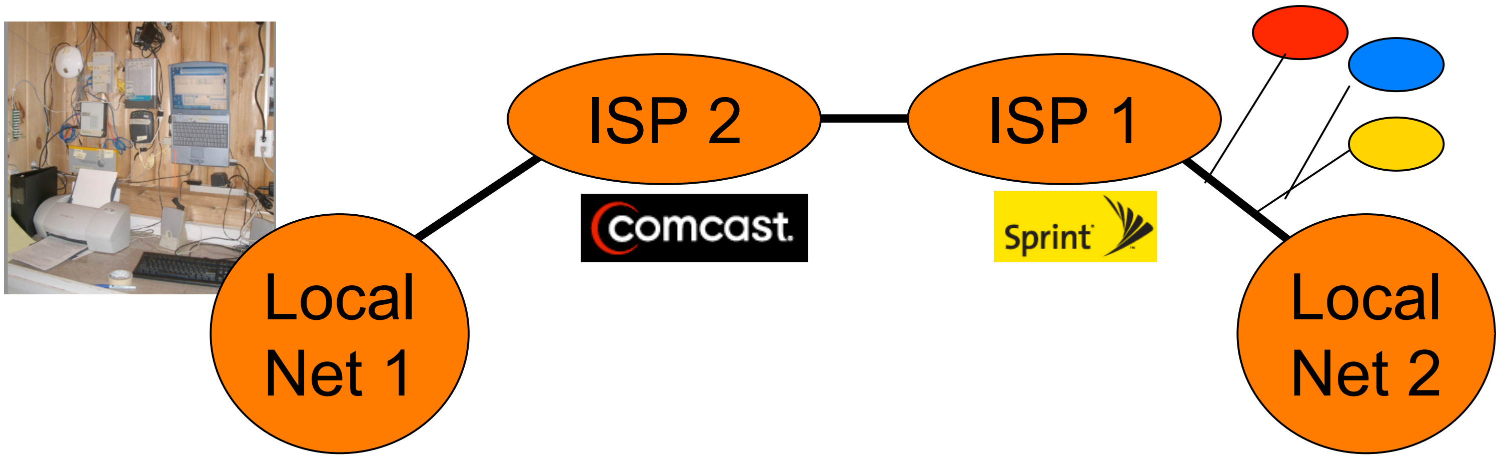
- Links carry information (bits)
  - Wire, wireless, fiber optic, smoke signals …
  - May be point-to-point or broadcast
- Switches move bits between links
  - Routers, gateways,bridges, CATV headend, PABXs, …
- Hosts are the communication endpoints
  - PC, PDA, cell phone, tank, toaster, …
  - Hosts have names

- Much other terminology: channels, nodes, intermediate systems, end systems, and much more.

Example – Local Area Network



Example – An Internetwork



- Internetwork is a network of networks
- The Internet is a global internetwork in which all participants speak a common language
  - IP, the Internet Protocol
- Sharing is everywhere

Sharing

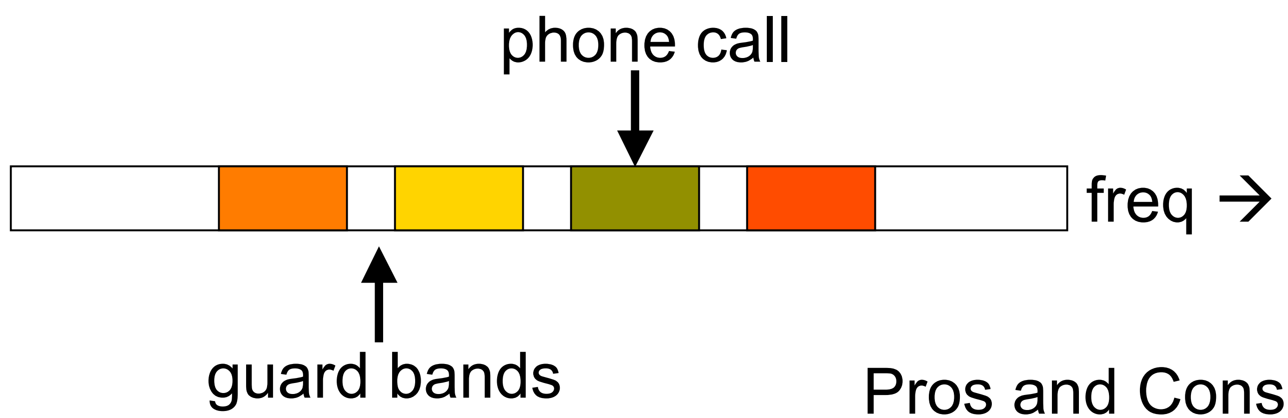
- The Vacation Home Problem
  - N families
  - One Vacation Home.
- How to coordinate access to the home?
  - Option 0: Do nothing
    - Conflicts
  - Option 1: Time share
    - Each family gets the whole vacation every N weeks
  - Option 2: Space share
    - Each family gets one bedroom in the vacation home anytime they want it
- *Q: Why not just buy more homes?*

A Network is like a vacation home

- Problem: How to multiplex (share) a resource amongst multiple users, especially sharing a network?
- First Solution: Static Partitioning
  - Like the condo solution
    - Families are nodes
    - Home is the link
- Really, *two* kinds of static partitioning
  - (Synchronous) Time Division Multiplexing (TDM, STDM)
    - (*"the smiths get it even weeks, the jones get it the others"*)
  - Frequency Division Multiplexing (FDM)
    - (*"the smiths stay on the first floor, the jones get the second"*)

(STATIC) Frequency Division Multiplexing

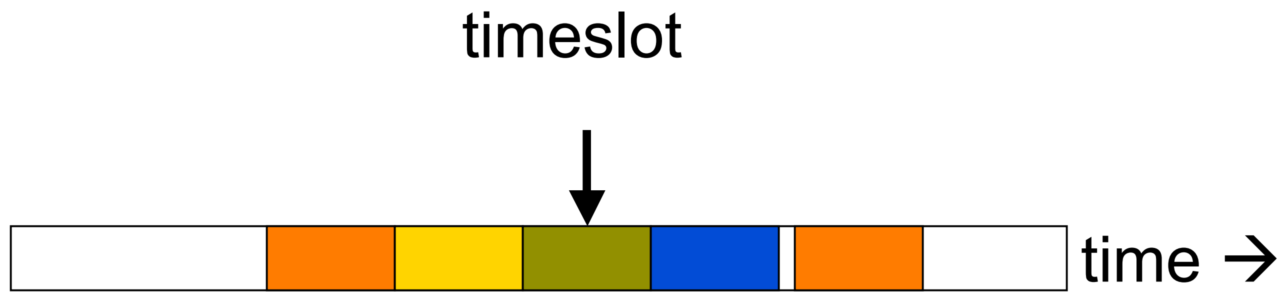
- Simultaneous transmission in different frequency bands
- "Speaking at different pitches"
  - (or, staying on different floors)
  - Eg, Take One 3MHz signal and break it into 1000 3KHz signals
    - Analog: RadiøTV, AMPS cell phones (800MHz)
  - Also called Wavelength DMA (WDMA) for fiber



Pros and Cons?

(STATIC) Time Division Multiplexing

- "Slice up" the given frequency band between users
- Speaking at different times
  - Staying at different times
  - Digital: used extensively inside the telephone network
  - T1 (1.5Mbps) is 24 x 8 bits/125us; also E1 (2Mbps, 32 slots)



Pros and Cons?

Statistical Multiplexing

- Static partitioning schemes (which assume peak rate) are not well-suited to data communications because peak rate >> average rate.
  - In other words, it's rare for the whole family to show up in a given week. Much more likely is nobody shows up, or just a few do.
  - Consequently… rooms, or whole vacation homes, go unused.
- Consider one condo, multiple floors, but only one bathroom.
- If we share on demand we can support more users
  - Based on the statistics of their transmissions
    - If you need more, you get more. If you need less, you get less.
    - It's all supposed to "balance out" in the end
    - Occasionally we might be oversubscribed
  - This is called <u>statistical multiplexing</u>
- Statistical multiplexing is heavily used in data networks

Statistical Multiplexing

- Consider:
  - 10 Mbps *peak bandwidth (p)*
  - Each node sends at 1 Mbps (*b)*
  - Each node is idle 90% of the time.
    - Prob[Node is sending] = 10% = .1
- How many nodes (n) can this system support?
  - Static partitioning: n = p/b
    - First mbps to first node, second to second node, …
    - Can support 10 nodes with probability 1
      - Q: what is probability of supporting 11 nodes?
  - Statistical multiplexing: n is defined with some probability
    - *pn = # peak nodes* (nodes required to saturate)
    - Pr[Node_i can send] = 1-Pr[saturated]
    - Pr[saturated] = Pr[p/b nodes are sending]
      - For example, consider a network with 10 nodes
        - » Pr[saturated] = 10^{-10} = 0.0000000100%
        - » Not likely! So keep adding users …

How many nodes can we add?

- Equivalent to asking *"what is the probability that* at least *p/b of n nodes are sending at the same time?"*
  - how many different ways can k = *p/b* nodes be sending at once?

To select k objects from a set of n objects is denoted as $P(n,k)$. To calculate:

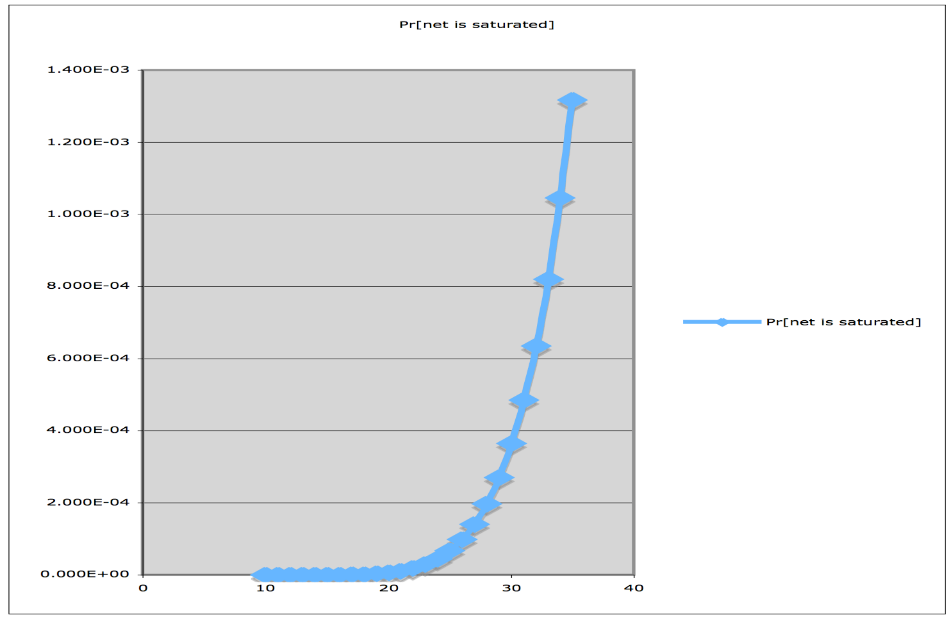
$$P(n,k) = n(n-1)(n-2)\ldots(n-k+1) = \frac{n!}{(n-k)!}$$

But order doesn't matter -->
C(n,k) = P(n,k) / k!

- Or, how many different ways can k = *p/b* nodes be sending at once?
- Pr[saturated] = Pr[p/b nodes are sending] * C(n,k)
  - For n = 20, there are C(n,k) = 184756 ways in which 10 nodes can be sending, so Pr[saturated] = .000018.

Turning the crank

NODES	# ways to get p/b sending and p/b not sending (C(n,p/b))	Pr[net is saturated]
1		
2		
3		
4		
5		
6		
7		
8		
9		
10	1	1.0000000E-10
11	11	9.9000000E-10
12	66	5.3460000E-09
13	286	2.0849400E-08
14	1001	6.5675610E-08
15	3003	1.7732415E-07
16	8008	4.2557795E-07
17	19448	9.3019181E-07
18	43758	1.8836384E-06
19	92378	3.5789130E-06
20	184756	6.4420434E-06
21	352716	1.1068602E-05
22	646646	1.8263193E-05
23	1144066	2.9080623E-05
24	1961256	4.4867246E-05
25	3268760	6.7300870E-05
26	5311735	9.8427522E-05
27	8436285	1.4069346E-04
28	13123110	1.9697084E-04
29	20030010	2.7057573E-04
30	30045015	3.6527724E-04
31	44352165	4.8529691E-04
32	64512240	6.3529777E-04
33	92561040	8.2036277E-04
34	131128140	1.0459625E-03
35	183579396	1.3179128E-03



Many users

NOTE: this assumes =p/b, not >= p/b!

Key Concepts

- Scalability
- Networks are comprised of links, switches and hosts
- Networks are used to share distributed resources
- Multiplexing lets multiple users share a resource
  - Static multiplexing is simple, but not efficient unless the workloads are static
  - Statistical multiplexing is more complicated and not guaranteed to work
  - but well-suited to data communications (bursty traffic)