CSC458 - Winter 2006

Introduction to Computer Networks Protocols and Layering

Stefan Saroiu
http://www.cs.toronto.edu/syslab/courses/csc458

University of Toronto at Mississauga

This Lecture

1. Administrative stuff
2. Introduction to Networks
3. Statistical Multiplexing
4. A top-down look at the Internet
5. Mechanics of protocols and layering
6. The OSI/Internet models

1. Administrative Stuff

- Important high-level questions?
- Is this the same as CSC458 from St. George?
  - Not really: focus here is on systems building
    - You will learn a lot about how the Internet works
    - You will learn less about the theoretical underpinnings of networks
- This guy is a new instructor … who is he?
  - More info see: http://www.cs.toronto.edu/~stefan

Visit the Course Web Page!

- Everything you need is on the course web page
  - http://www.cs.toronto.edu/syslab/courses/csc458
- Your TODO list:
  - Visit and familiarize yourself with the course web page
  - Get Computer Networks by Peterson and Davie (3rd edition)
  - Read chapters 1 and 2
  - Go to the tutorial (after this class)
  - Start on Fishnet assignment 1
  - Start on homework 1
- Is there anything unclear on the handout?

TAs

- Joe Lim
  - He's the grand-master of the projects in this course!
  - This means:
    - He will answer your questions and help you with the projects
    - He won't answer nor help you with the homework

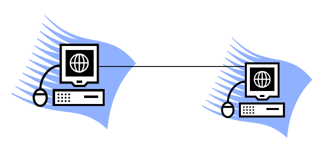
Setting the right expectations

- Read the chapters in the book
  - I will not go over the material in the book during lectures
  - I will assume that you have read the chapters
- Homework
  - Make sure you've read the book chapters first
  - Start early
- Projects
  - You will most likely fail this class if one of the following:
    - You're struggling with Java, a text editor (vi or Emacs), make files, Unix tools …..
    - You start working on the projects 3-4 days before the deadline …
  - Think/design/create first before sitting down to code

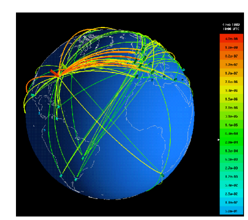
What is a Network?

A Network in CSC458

- "Network" is clearly an overloaded word:
  - Economic networks, regulatory networks, social networks…
  - Telephone, Cable TV, Bank tellers, computer clusters
- For 458, a network is what you get anytime you connect two or more computers together by some kind of a link.



OR



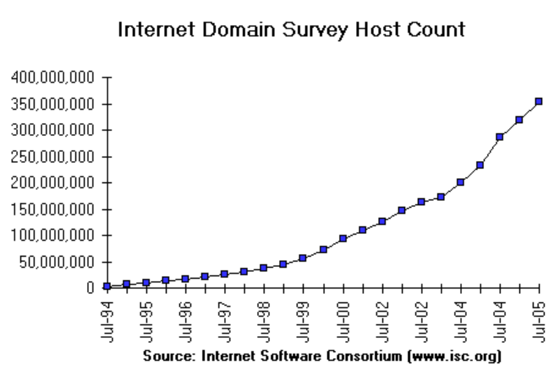
This Lecture

1. Administrative stuff
2. Introduction to Networks
3. Statistical Multiplexing
4. A top-down look at the Internet
5. Mechanics of protocols and layering
6. The OSI/Internet models

2. The networks we study

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

The meaning of "Large-scale"



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 out of order
  - May be read by others
  - May be modified 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
- Leads to…

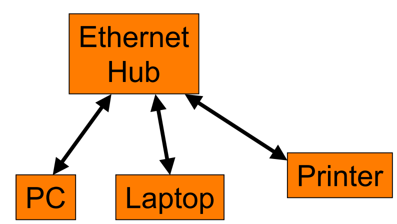
Heterogeneous Networks

- Heterogeneous: Made up of different kinds of stuff
- Homogeneous: Made up of the same kind of stuff
- Principles
  - Homogeneous networks are easier to deal with
  - Heterogeneous networks lead to greater innovation and scale
  - Consider telephone network vs. Internet
  - Reasons?

Model of a Network

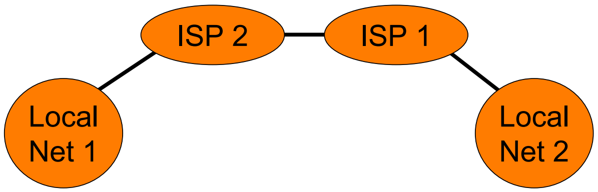
- Links carry information (bits)
  - Wire, wireless, fiber optic, smoke signals …
  - May be point-to-point or broadcast
- Switches move bits between links
  - Routers, gateways,bridges, CATV headend, PABXs, …
- Hosts are the communication endpoints
  - PC, PDA, cell phone, tank, toaster, …
  - Hosts have names
- Much other terminology: channels, nodes, intermediate systems, end systems, and much more.

Example – Local Area Network



- Your home network
  - Ethernet is a broadcast-capable multi-access LAN

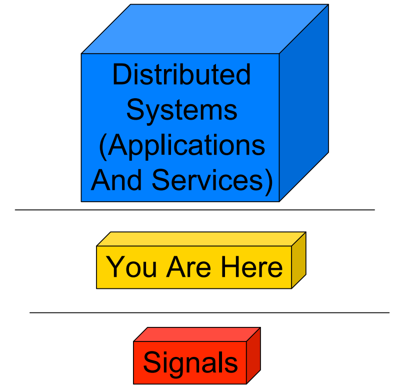
Example – An Internetwork



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

Goal of this Course

- You will understand how to design and build *large, distributed computer* networks.
  - Fundamental problems in building networks
  - Design principles of proven value
  - Common implementation technologies
- This is a systems course, not queuing theory, signals, or hardware design.
- We focus on networks, rather than applications or services that run on top of them (distributed systems).



This Lecture

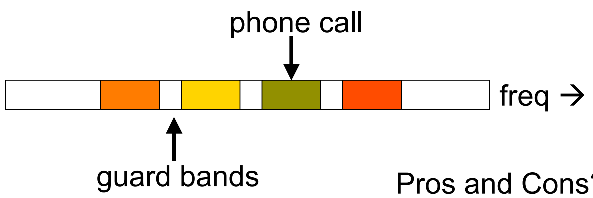
1. Administrative stuff
2. Introduction to Networks
3. Statistical Multiplexing
4. A top-down look at the Internet
5. Mechanics of protocols and layering
6. The OSI/Internet models

3. An example technical problem: multiplexing

- Networks are shared among users
  - This is an important benefit of building them
    - (why we can't just buy everybody their own network!)
- How should you multiplex (share) a resource amongst multiple users?
  - e.g., how do you share a network link?
- First Solution: Static Partitioning
  - (Synchronous) Time Division Multiplexing (TDM, STDM)
  - Frequency Division Multiplexing (FDM)

Frequency Division Multiplexing

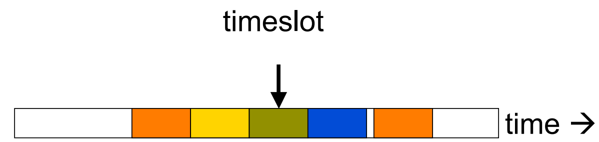
- Simultaneous transmission in different frequency bands
- "Speaking at different pitches"
  - e.g., take one 3MHz signal and break it into 1000 3KHz signals
    - Analog: Radio, TV, AMPS cell phones (800MHz)
  - also called Wavelength DMA (WDMA) for fiber



Pros and Cons?

Time Division Multiplexing

- "Slice up" the given frequency band between users
- Speaking 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?

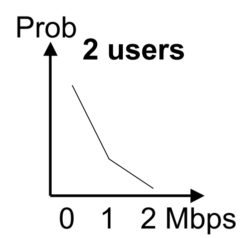
Statistical Multiplexing

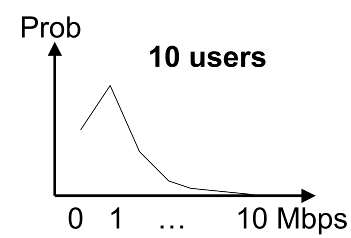
- Static partitioning schemes are not well-suited to data networks
  - why? because peak rate >> average rate.
    - it's rare for many clients to want to transmit at the same time.
    - so, statically assigning fractions of the link wastes capacity, since users tend to underuse their fraction
  - (Q: When would S.P. schemes be well suited to communications?)

- 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> -- used heavily in data networks

Why We Like Statistical Multiplexing

- One user sends at 1 Mbps and is idle 90% of the time.
  - 10 Mbps channel; 10 users if statically allocated
- Two scenarios: 2 users in the population, or 10 users in population
  - what is the probability of a certain bandwidth consumption at any given moment in time?





Example continued

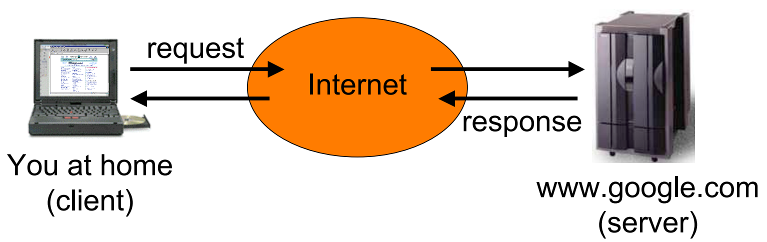
- For 10 users, Prob(need 10 Mbps) = 10^{-10} = 0.0000000100%
- Not likely! So keep adding users …
- For 35 users, Prob(>10 active users) = 0.17%, which is acceptably low
- With statistical multiplexing, we can support three times as many users than static allocation!
- What's the rub?

This Lecture

1. Administrative stuff
2. Introduction to Networks
3. Statistical Multiplexing
4. A top-down look at the Internet
5. Mechanics of protocols and layering
6. The OSI/Internet models

4. A Brief Tour of the Internet

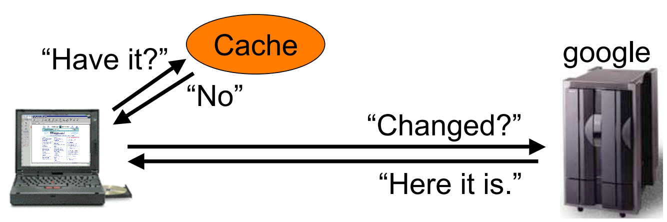
- What happens when you "click" on a web link?



- This is the view from 10,000 ft …

9,000 ft: Scalability

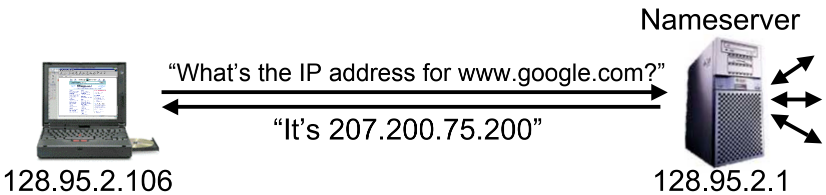
- Caching improves scalability



- We cut down on transfers:
  - Check cache (local or proxy) for a copy
  - Check with server for a new version

8,000 ft: Naming (DNS)

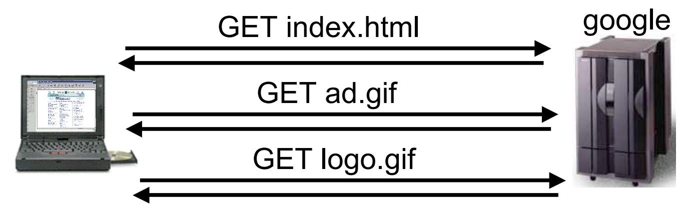
- Map domain names to IP network addresses



- All messages are sent using IP addresses
  - So we have to translate names to addresses first
  - But we cache translations to avoid doing it next time (why?)

7,000 ft: Sessions (HTTP)

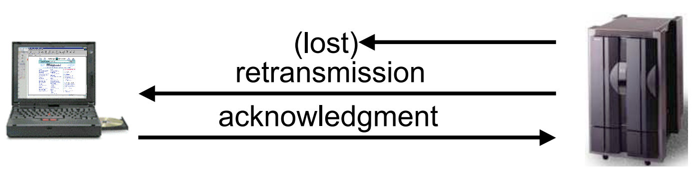
- A single web page can be multiple "objects"



- Fetch each "object"
  - either sequentially or in parallel

6,000 ft: Reliability (TCP)

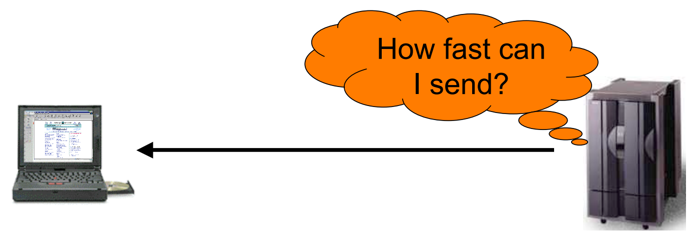
- Messages can get lost



- We acknowledge successful receipt and detect and retransmit lost messages (e.g., timeouts)

5,000 ft: Congestion (TCP)

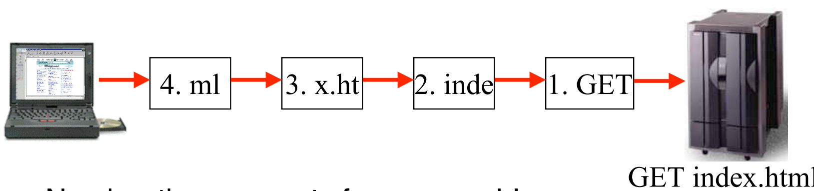
- Need to allocate bandwidth between users



- Senders balance available and required bandwidths by probing network path and observing the response

4,000 ft: Packets (TCP/IP)

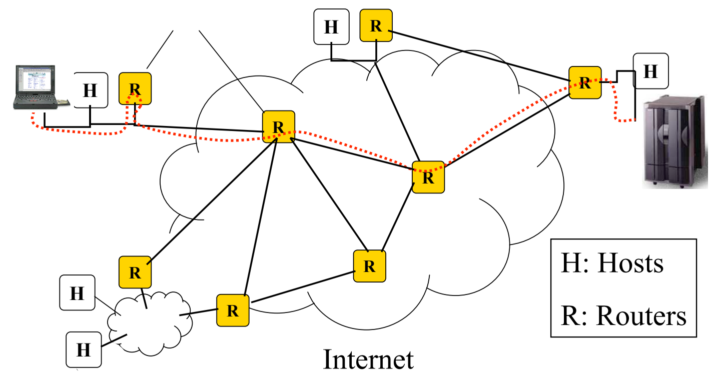
- Long messages are broken into packets
  - Maximum Ethernet packet is 1.5 Kbytes
  - Typical web page is 10 Kbytes



- Number the segments for reassembly

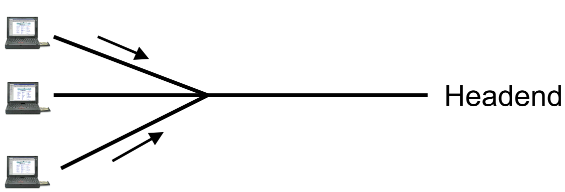
3,000 ft: Routing (IP)

- Packets are directed through many routers



2,000 ft: Multi-access (e.g., Cable)

- May need to share links with other senders



- Poll headend to receive a timeslot to send upstream
  - Headend controls all downstream transmissions
  - A lower level of addressing (than IP addresses) is used … why?

1,000 ft: Framing/Modulation

- Protect, delimit and modulate payload as signal

Sync / Unique	Header	Payload w/ error correcting code

- E.g, for cable, take payload, add error protection (Reed-Solomon), header and framing, then turn into a signal
  - Modulate data to assigned channel and time (upstream)
  - Downstream, 6 MHz (~30 Mbps), Upstream ~2 MHz (~3 Mbps)

This Lecture

1. Administrative stuff
2. Introduction to Networks
3. Statistical Multiplexing
4. A top-down look at the Internet
5. Mechanics of protocols and layering
6. The OSI/Internet models

5. Protocols and Layering

- We need abstractions to handle all this system complexity

  A <u>protocol</u> is an agreement dictating the form and function of data exchanged between parties to effect communication

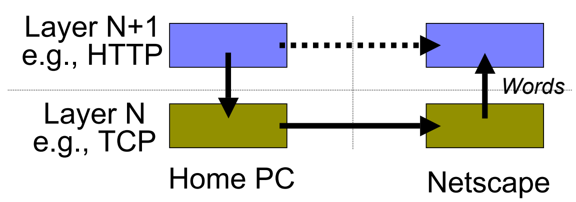
- Two parts:
  - Syntax: format -- where the bits go
  - Semantics: meaning -- what the words mean, what to do with them
- Examples:
  - Ordering food from a drive-through window
  - IP, the Internet protocol
  - TCP and HTTP, for the Web

Protocol Standards

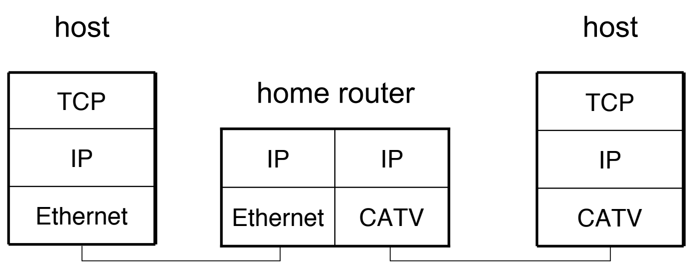
- Different functions require different protocols
- Thus there are many protocol standards
  - E.g., IP, TCP, UDP, HTTP, DNS, FTP, SMTP, NNTP, ARP, Ethernet/802.3, 802.11, RIP, OPSF, 802.1D, NFS, ICMP, IGMP, DVMRP, IPSEC, PIM-SM, BGP, …
- Organizations: IETF, IEEE, ITU
- IETF (www.ietf.org) specifies Internet-related protocols
  - RFCs (Requests for Comments)
  - "We reject kings, presidents and voting. We believe in rough consensus and running code." – Dave Clark.

Layering and Protocol Stacks

- Layering is how we combine protocols
  - Higher level protocols build on services provided by lower levels
  - Peer layers communicate with each other

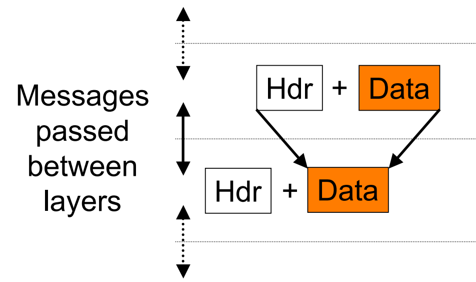


Example – Layering at work



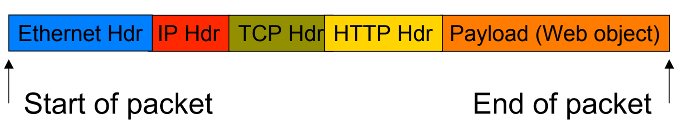
Layering Mechanics

- Encapsulation and de(en)capsulation



A Packet on the Wire

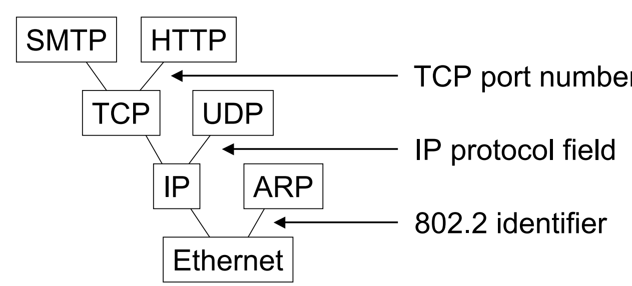
- Starts looking like an onion!



- This isn't entirely accurate
  - ignores segmentation and reassembly, Ethernet trailers, etc.
- But you can see that layering adds overhead

More Layering Mechanics

- Multiplexing and demultiplexing in a protocol graph



This Lecture

1. Administrative stuff
2. Introduction to Networks
3. Statistical Multiplexing
4. A top-down look at the Internet
5. Mechanics of protocols and layering
6. The OSI/Internet models

6. OSI/Internet Protocol Stacks

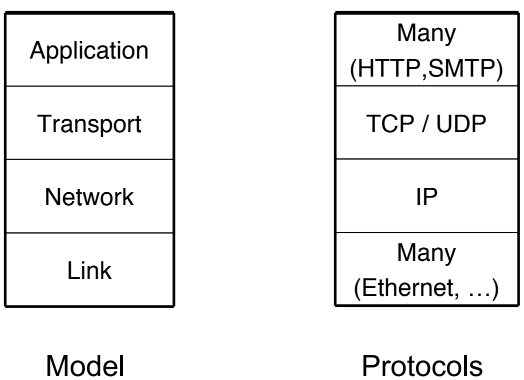
Key Question: What functionality goes in which protocol?

- The “End to End Argument” (Reed, Saltzer, Clark, 1984):

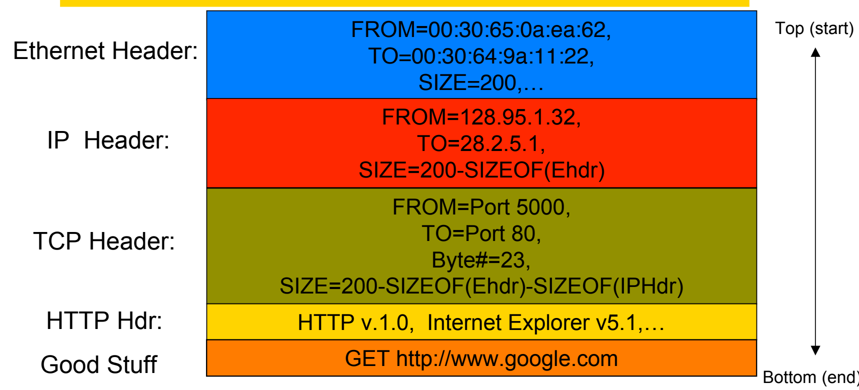
  Functionality should be implemented at a lower layer only if it can be correctly and completely implemented. (Sometimes an incomplete implementation can be useful as a performance optimization.)

- Tends to push functions to the endpoints, which has aided the transparency and extensibility of the Internet.

Internet Protocol Framework



What’s Inside a Packet



OSI "Seven Layer" Reference Model

- Seven Layers:

Application
Presentation
Session
Transport
Network
Link
Physical

Their functions:

- Up to the application
- Encode/decode messages
- Manage connections
- Reliability, congestion control
- Routing
- Framing, multiple access
- Symbol coding, modulation

Key Concepts

- Networks are comprised of links, switches and hosts
- Networks are used to share distributed resources
  - Key problems revolve around effective resource sharing
- Multiplexing lets multiple users share a resource

- Static multiplexing is simple
  - but not efficient unless the workloads are static
- Statistical multiplexing is more complicated
  - not guaranteed to work
  - but well-suited to data communications (bursty traffic)
- Protocol layers are modularity used in networks to handle complexity
- Internet/OSI models are roadmap of what function belongs at what layer