CSE 461: Introduction to Computer Communications Networks
Winter 2010

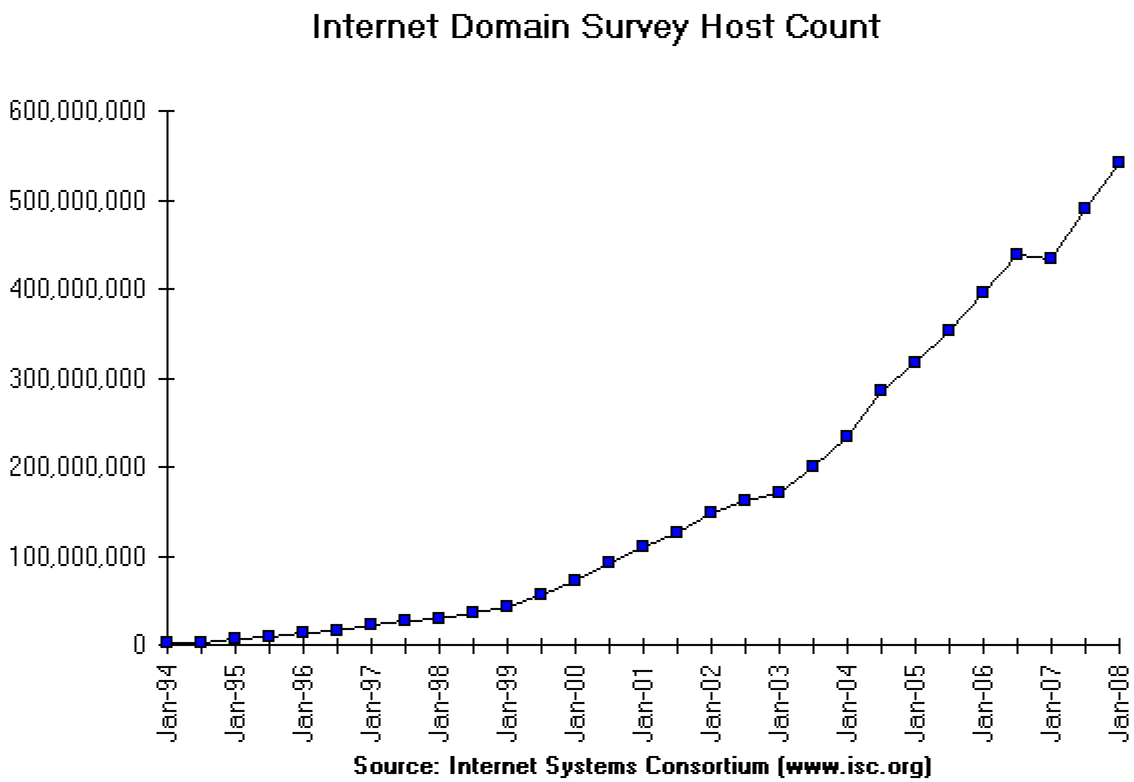
Module 1b
Course Introduction

John Zahorjan
zahorjan@cs.washington.edu
534 Allen Center

Network *Scalability*

- For this course, a "network" is what connects two or more computers. (What's a "computer"?)
- We are interested in network architectures that are "scalable" – continue to work efficiently even as the size of the system grows by orders of magnitude

Why is scalability important?



- The basic network design happened in the 1970's.
  - There were maybe 10,000's of computers in the world at the time
- Not only *could* the design scale, it provided a combination of cost and benefit that drove demand

Implications of Scaling

1. Sharing
   - E.g., shared network links, not N^2 dedicated links

2. Intrinsic Unreliability
   - 5 year MTBF x 1 hour MTR x 600,000,000 nodes ⇒
     ~14,000 nodes down at any moment

3. Distribution
   - *"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

4. Heterogeneous
   - HW, SW, uses

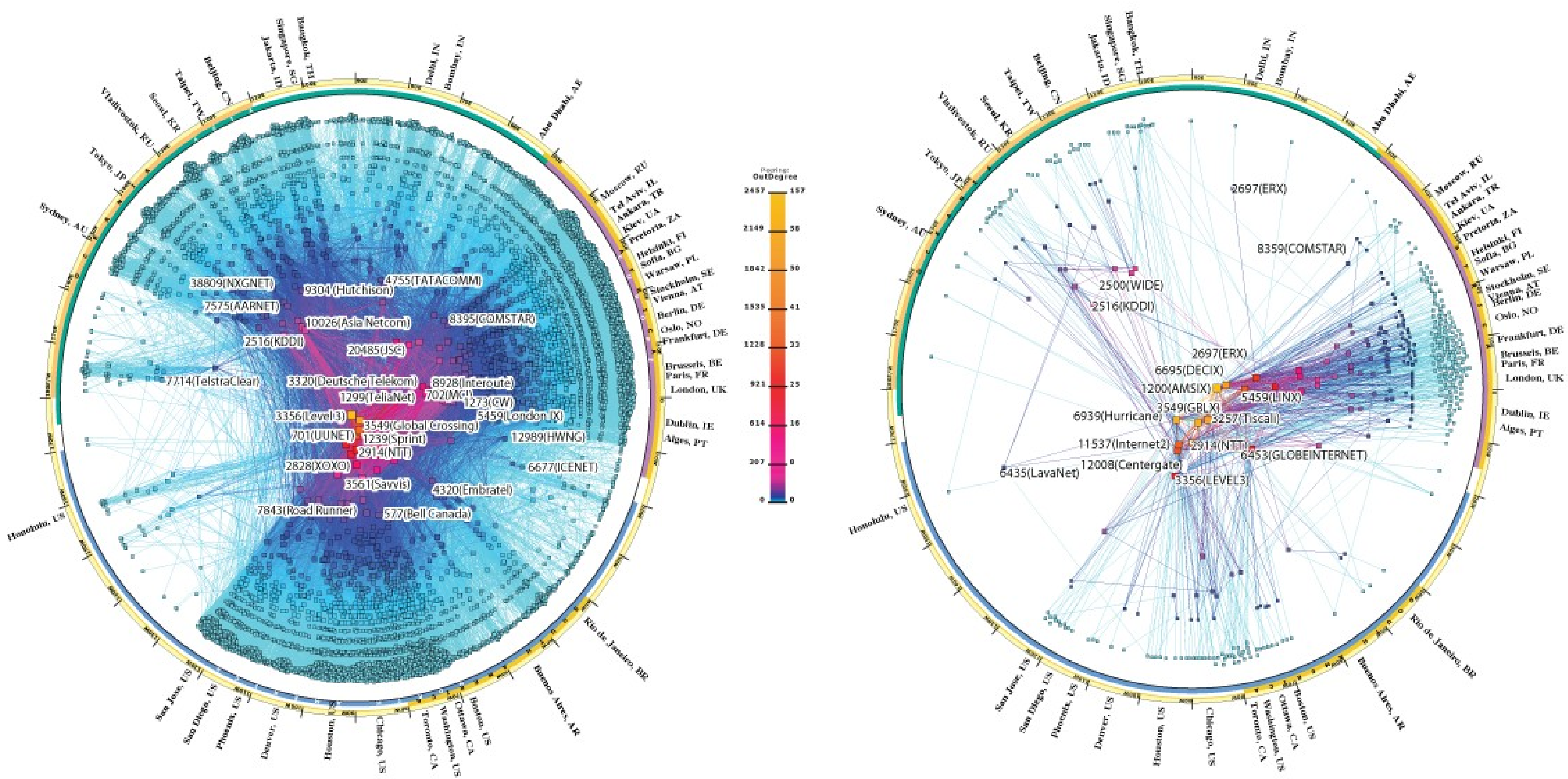
5. Autonomous Authorities
   - Many different owners ⇒ goals ⇒ policies

IPv4 & IPv6 INTERNET TOPOLOGY MAP JANUARY 2009

About 23,000 IPV4 AS's

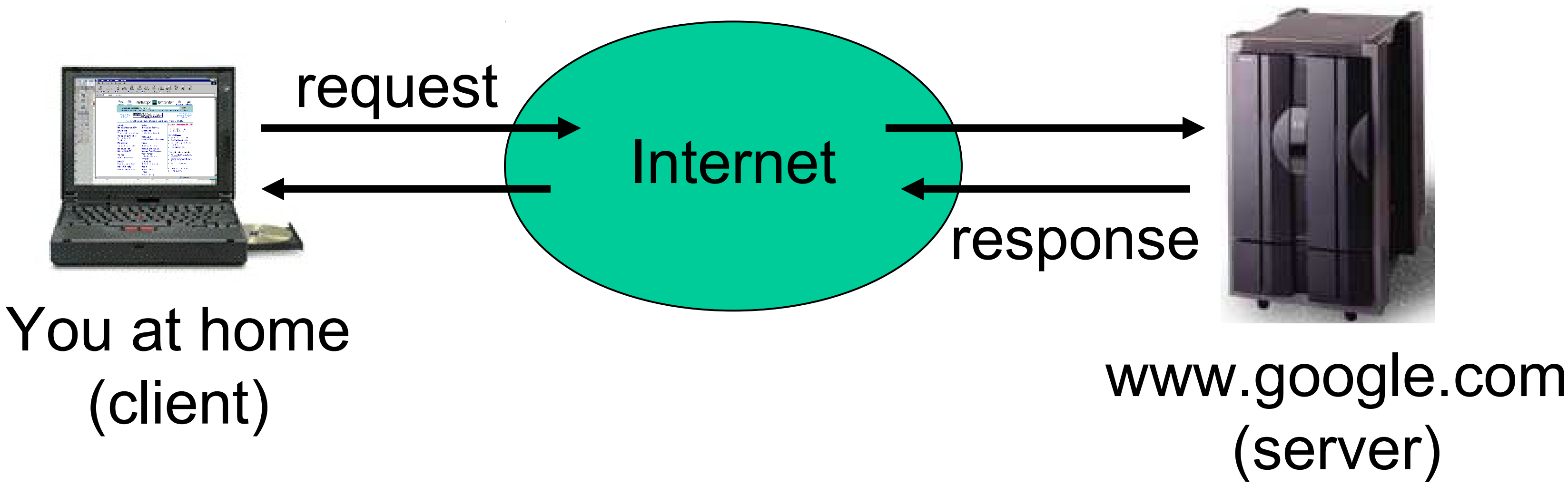
AS-level INTERNET GRAPH

IPv4 **IPv6**



A Brief Tour of the Internet

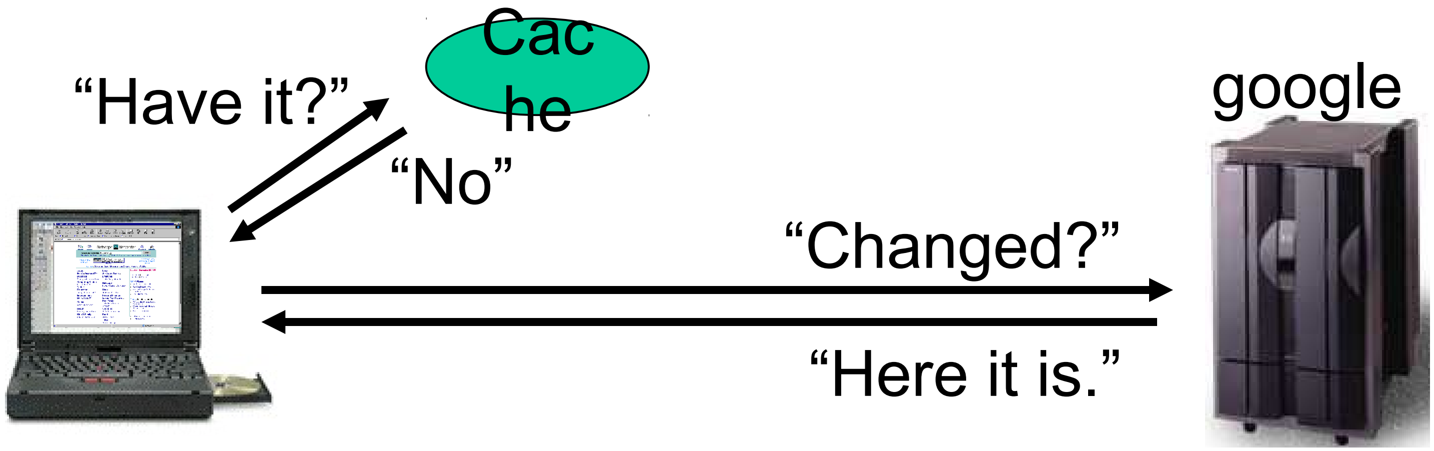
- What happens when you “click” on a web link?



- This is the view from 10,000 ft …

9,000 ft: Scalability

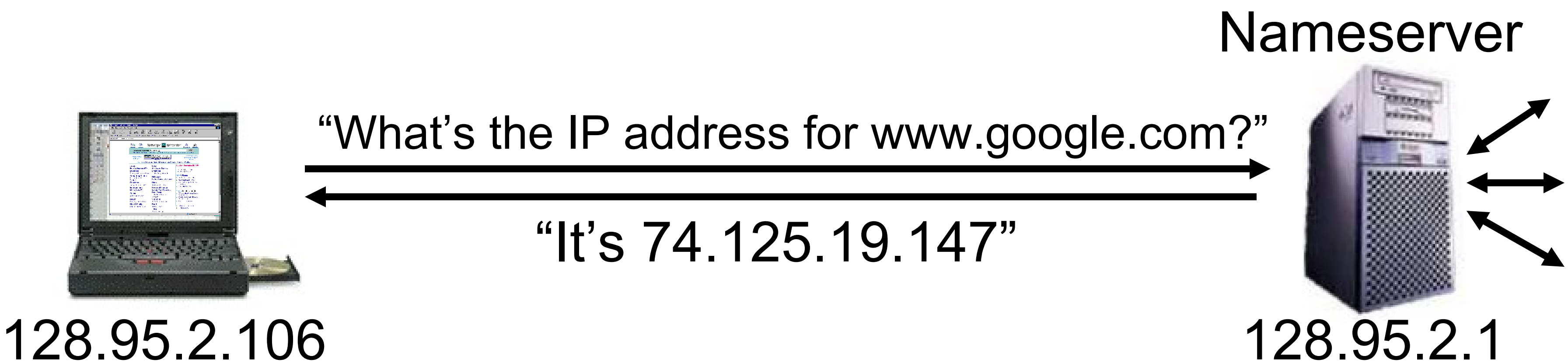
- Caching improves scalability



- 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)

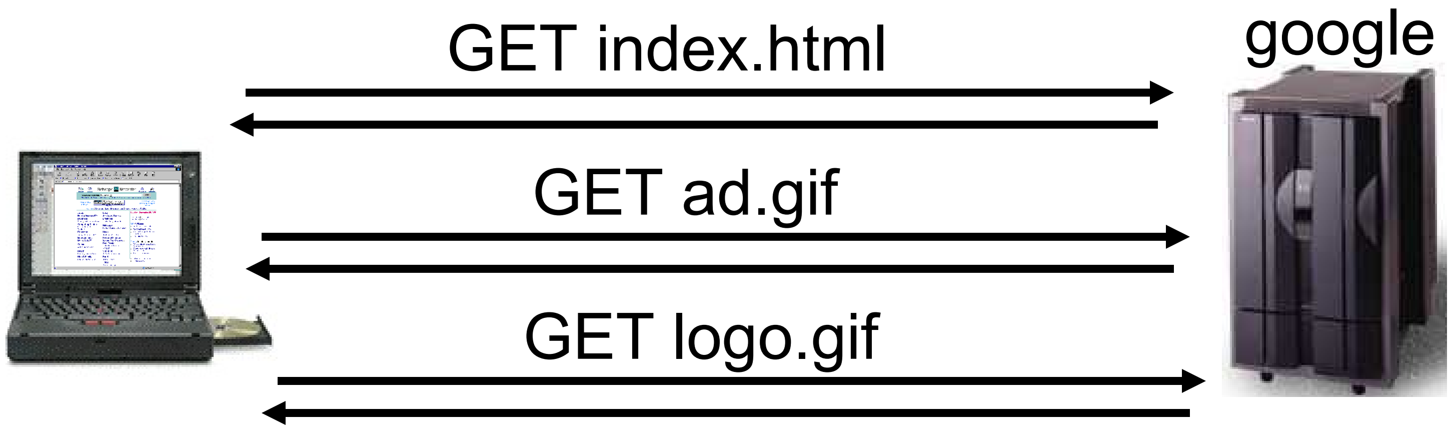
- Map domain names to IP network addresses



- 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

7,000 ft: Sessions (HTTP)

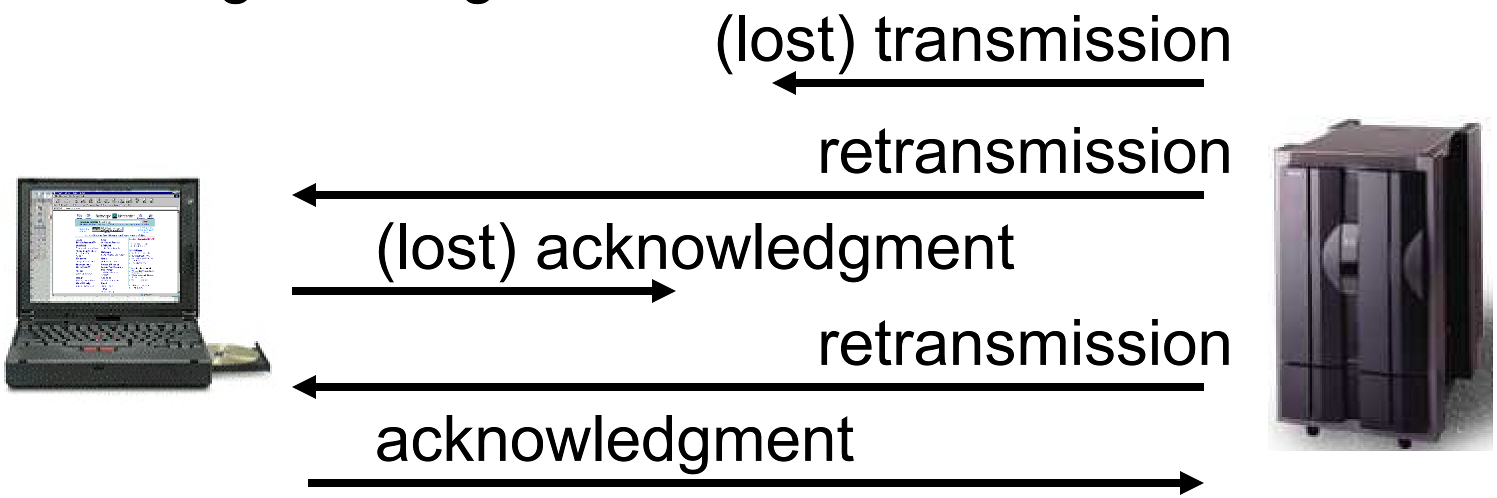
- A single web page can be multiple “objects”



- Fetch each “object”
  - either sequentially or in parallel

6,000 ft: Reliability (TCP)

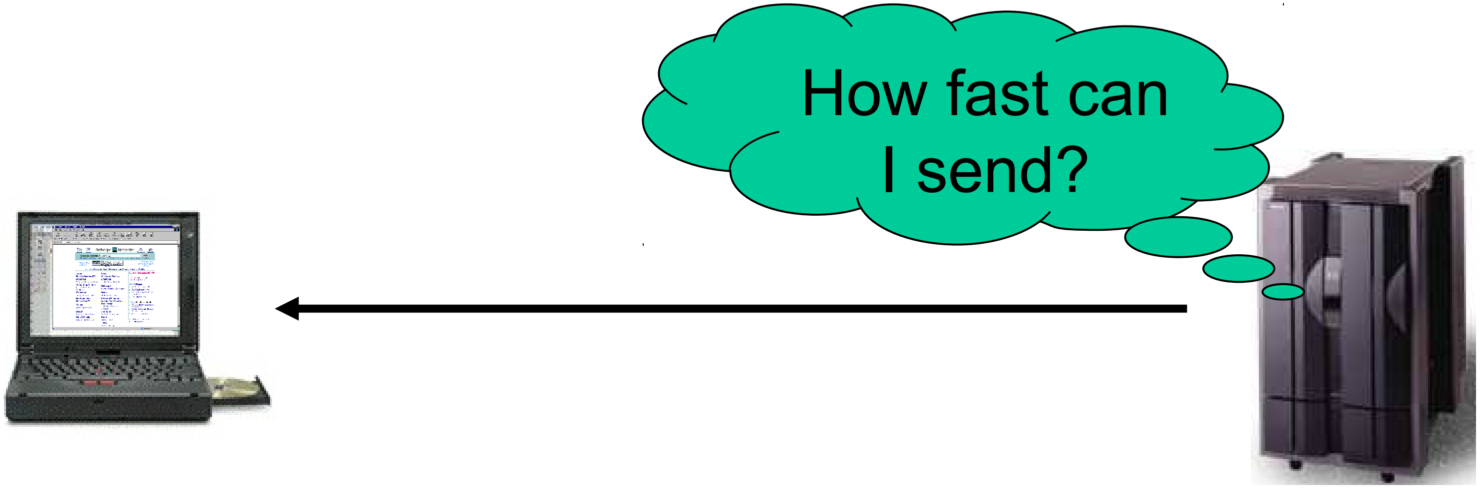
- Messages can get lost



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

5,000 ft: Congestion (TCP)

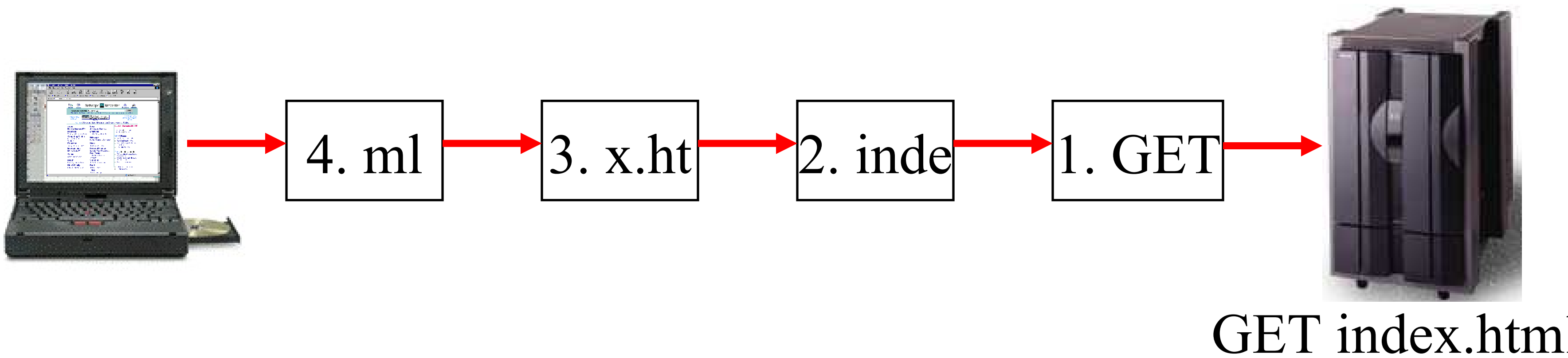
- Need to allocate bandwidth among users



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

4,000 ft: Packets (TCP/IP)

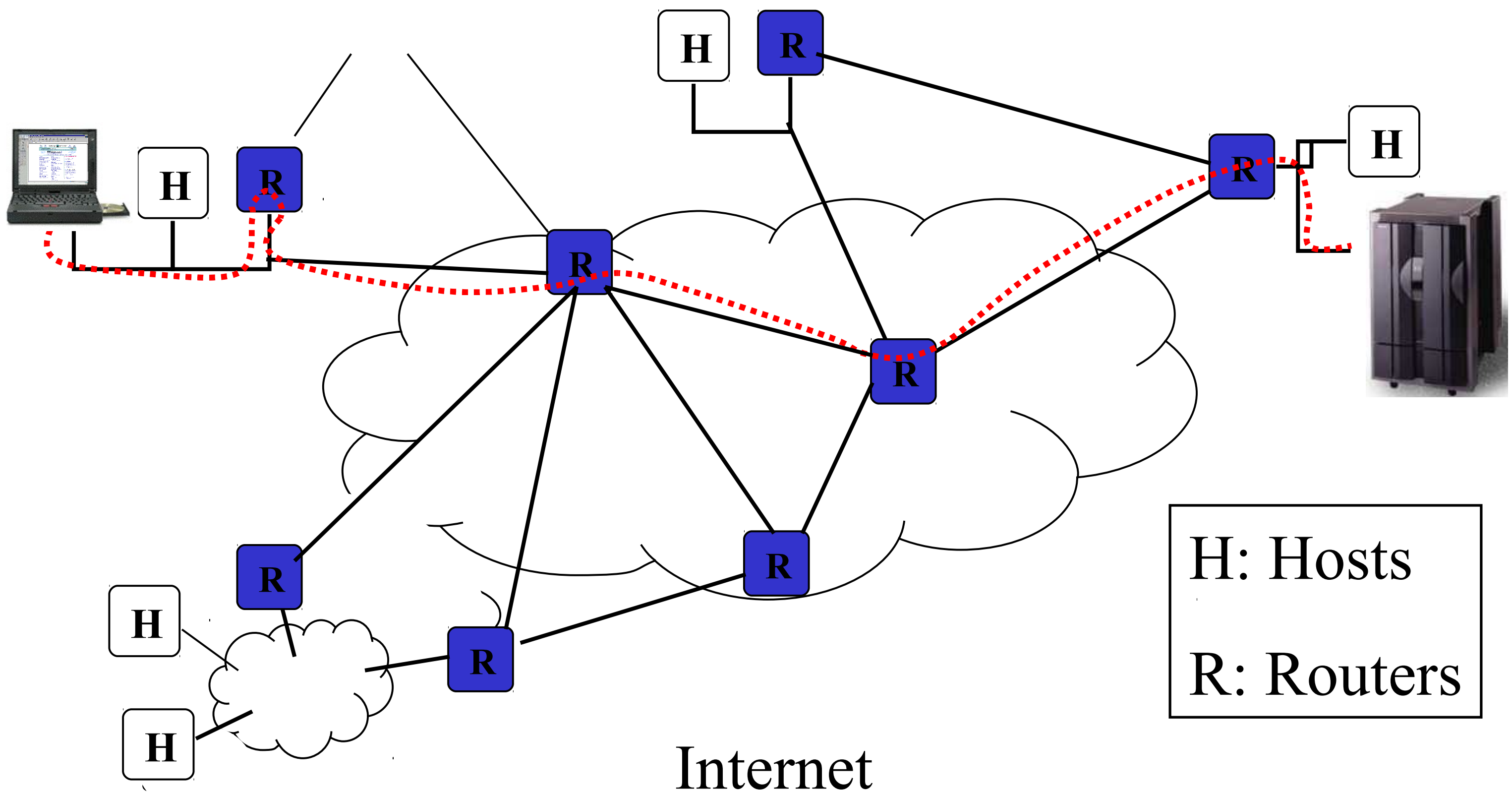
- Long messages are broken into packets
  - Maximum Ethernet packet is 1.5 Kbytes
  - Typical web page is 10 Kbytes



- Number the segments for reassembly

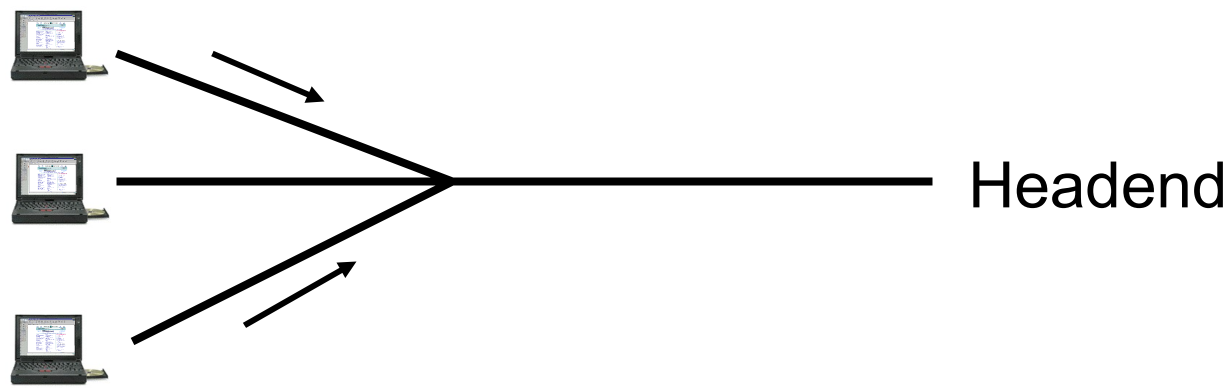
3,000 ft: Routing (IP)

- 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, 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)

Part 3. Protocols and Layering

- We need abstractions to handle all this system complexity

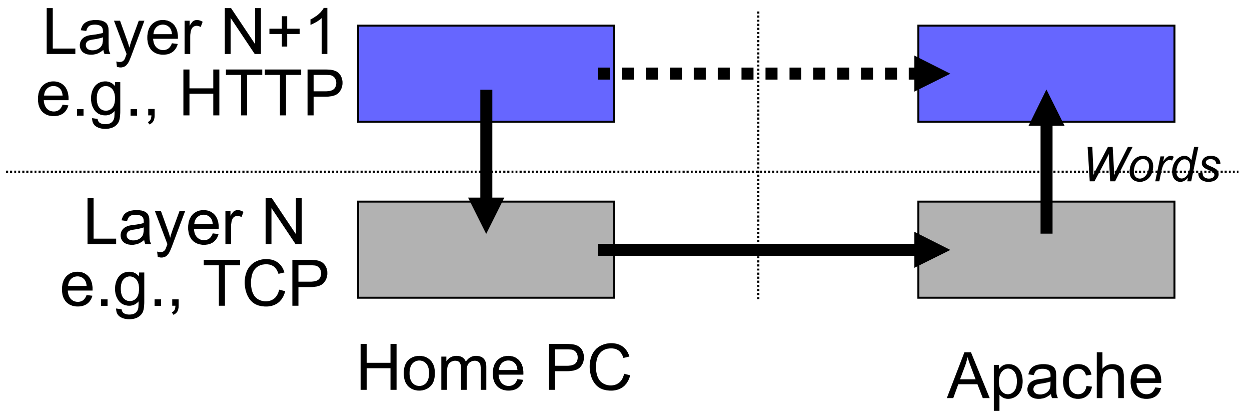
  A <u>protocol</u> is an agreement dictating the form and function of data exchanged between parties to effect communication

- Two parts:
  - Syntax: format -- where the bits go
  - Semantics: meaning -- what the words mean, what to do with them

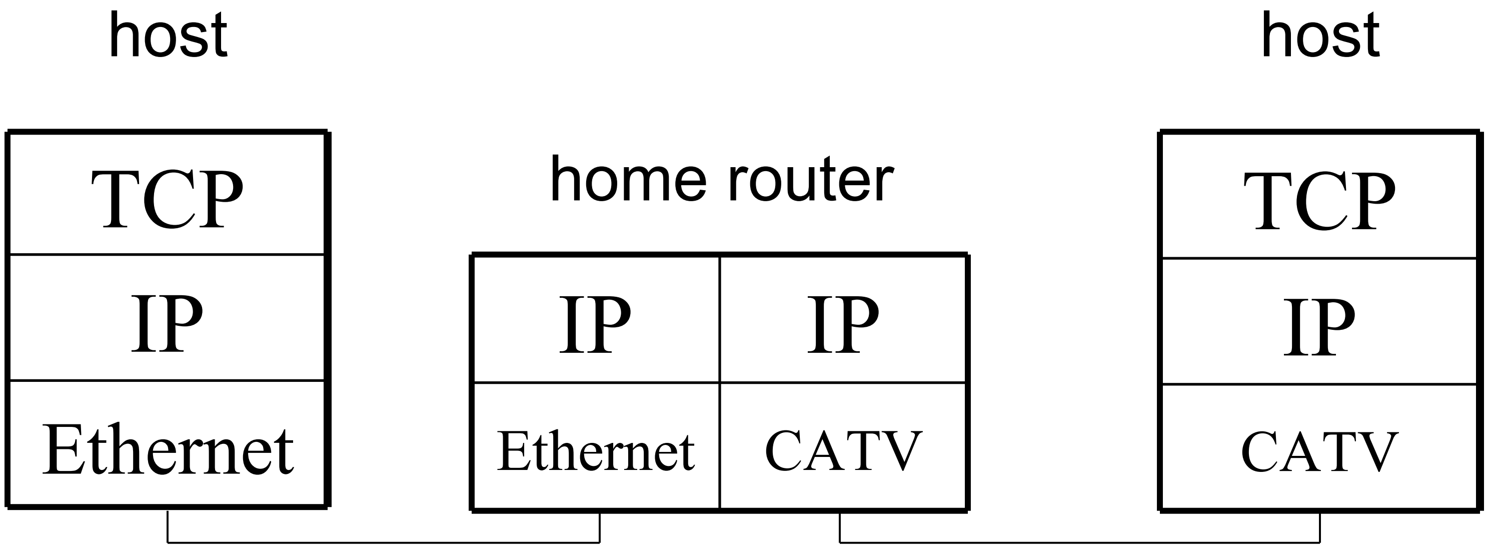
- Examples:
  - Ordering food from a drive-through window
  - TCP/IP, the Internet protocol
  - HTTP, for the Web

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



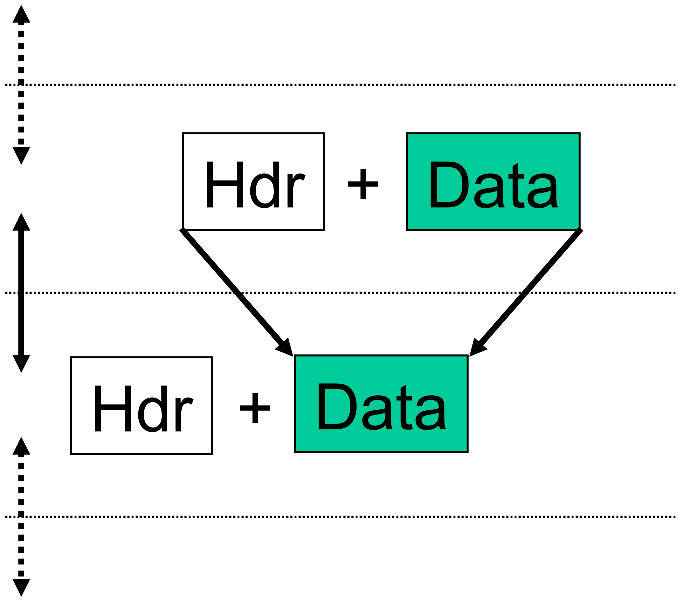
Example – Layering at work



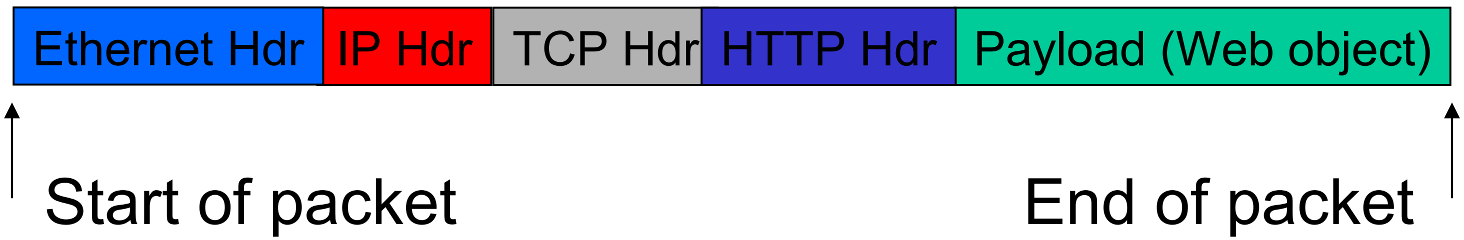
Layering Mechanics

- Encapsulation and de(en)capsulation

Messages passed between layers



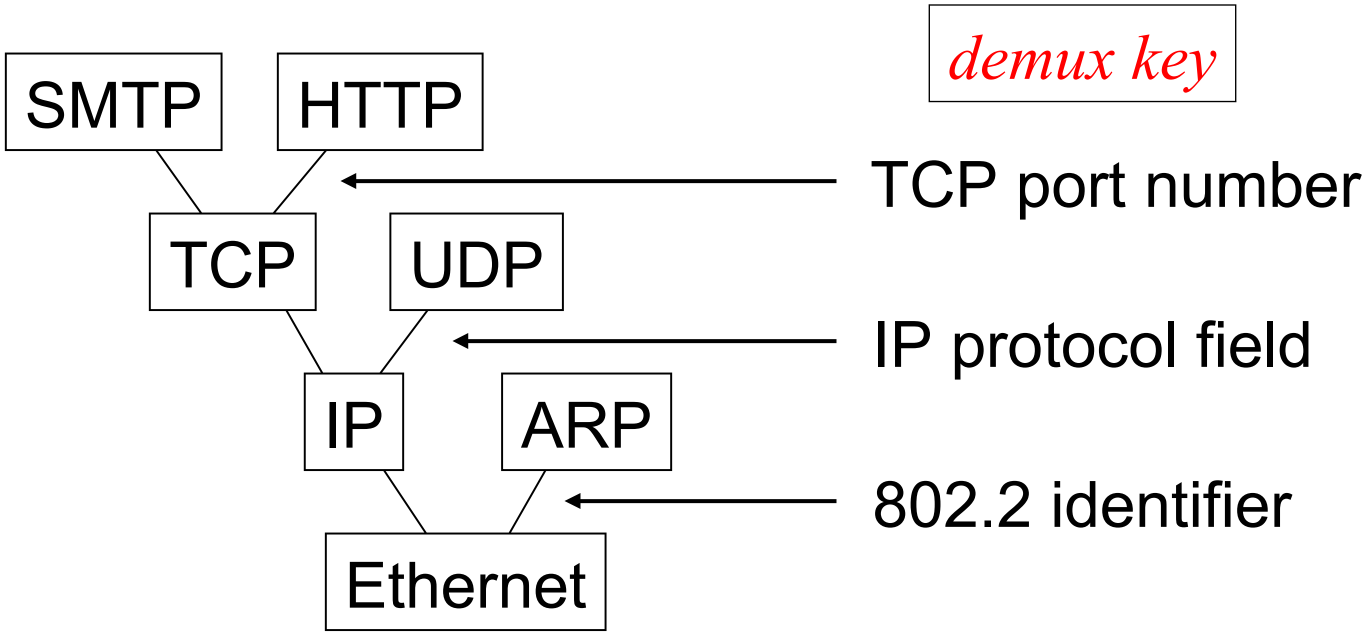
A Packet on the Wire



- This isn't entirely accurate
  - ignores segmentation and reassembly, Ethernet trailers, etc.
- But you can see that:
  - layering adds overhead
  - one protocol's header is another protocol's data

More Layering Mechanics

- Multiplexing and demultiplexing in a protocol graph



OSI/Internet Protocol Stacks

Key Question: What functionality goes in which protocol?

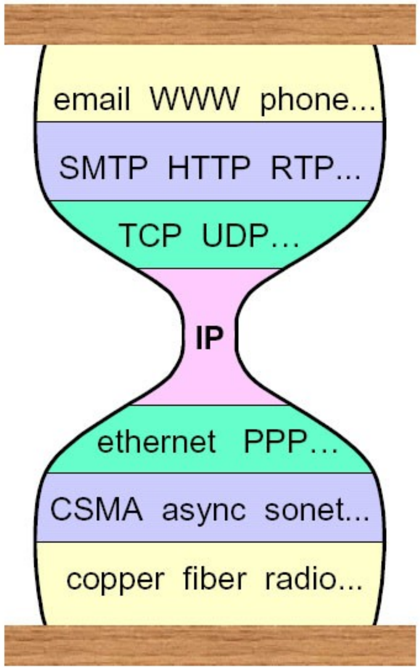
- The "End to End Argument" (Reed, Saltzer, Clark, 1984):

  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

Model	Protocols
Application	Many (HTTP,SMTP)
Transport	TCP / UDP
Network	IP
Link	Many (Ethernet, …)



The "narrow waist"

OSI "Seven Layer" Reference Model

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