CSE/EE 461 Introduction to Computer-Communication Networks

David Wetherall
djw@cs.washington.edu

Winter 2000.

This Lecture

1. Administrivia
2. Course Intro
3. A Brief Look at the Internet
4. Protocols and Layering

1. Administrivia – People

- David Wetherall (Instructor)
  - djw@cs.washington.edu
  - Sieg 210, Th 11 → 12
- Eric Hoffman (TA, Labs)
  - hoffman@cs.washington.edu
  - Tu 1 → 2
- Ratul Mahajan (TA, Homeworks)
  - ratul@cs.washington.edu
  - M 10 → 11

Materials

- Course Web page
  - www.cs.washington.edu/education/course/461/00wi
- Mailing list
  - Important – make sure you join!
- Textbook
  - S. Keshav, “An Engineering Approach to Computer Networking”, Addison-Wesley, 1997.

Work and Grading

- 4 homeworks 20%
- 4 projects 40%
- Midterm 15%
- Final 25%

- Late Policy
- Collaboration/Cheating Policy

Programming Projects

- You will add forwarding, routing, reliability and congestion control to an emulated network.
  - 4 projects, progressively more difficult
- Developed by Eric Hoffman
- Computer lab or do-it-yourself
  - Grab a partner
  - Get an account!

Timeline

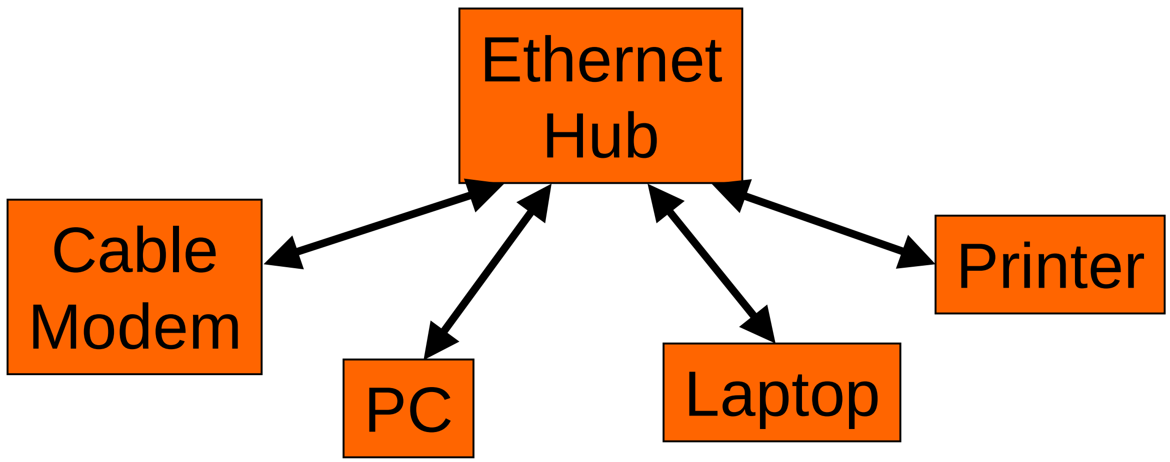
- Homeworks
  - Due at end of weeks 3, 5, 7, and 9 (tentative)
- Projects
  - Due at end of weeks 4, 6, 8, and 10 (tentative)
- Midterm
  - Monday 2/7, in class
- Final
  - Monday 3/13, at 4:30

2. Intro – What is 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, …

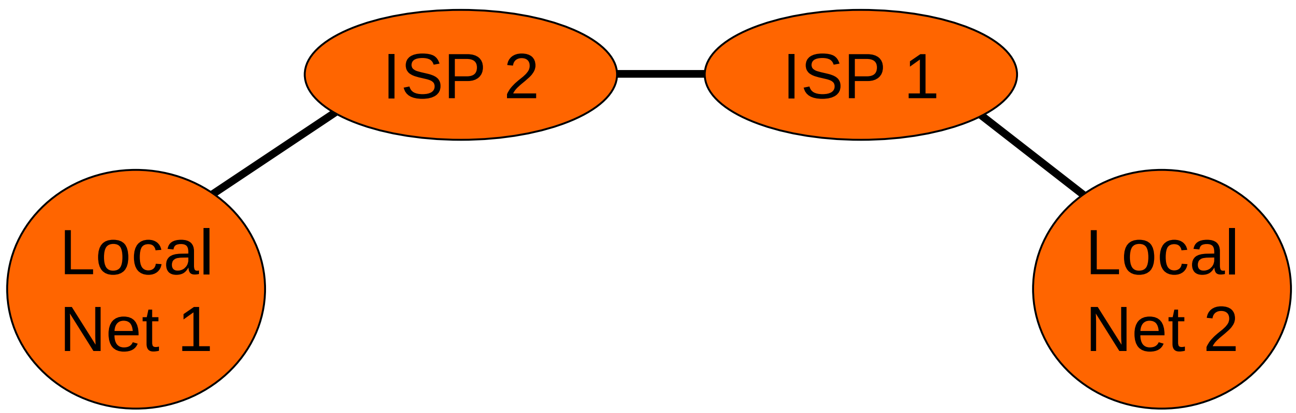
- Also called channels, nodes, intermediate systems, end systems, and much more.

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

Other Networks

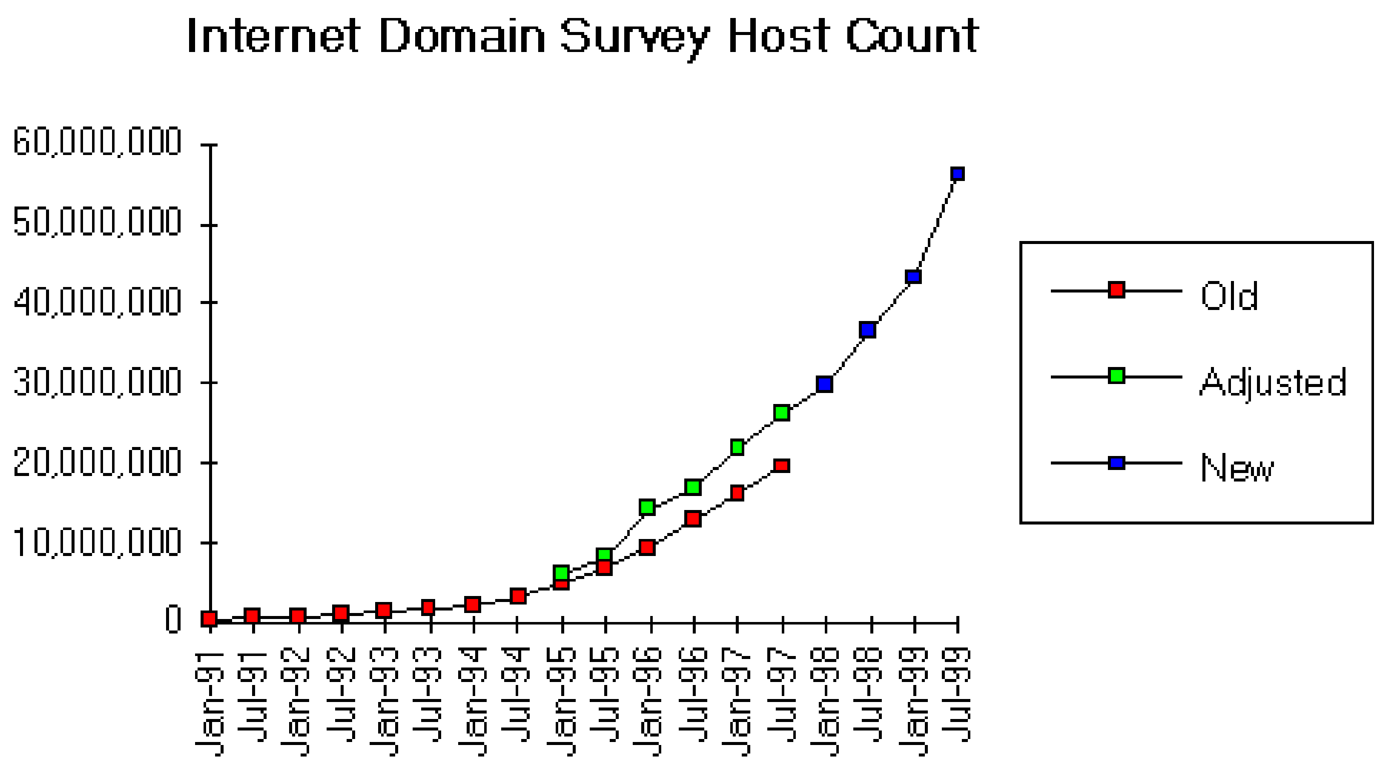
- You’ve all used networks:
  - Telephone, Cable TV,
  - ATMs, Processor Interconnects

- We are interested in networks that are:
  - Distributed
  - Large scale
  - Multi-purpose

The meaning of “Distributed”

- There are distributed and parallel networks:
  - Cash machines versus a parallel computer
- What is the essential difference?
  - Tolerance of failed components
  - Decentralized operation
  - Heterogeneity
- Hard to get it right
  - “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

The meaning of “Large-scale”



Source: Internet Software Consortium (http://www.isc.org/)

The meaning of "Multi-purpose"

- Telephone network
  - Designed for telephone calls
- Internet
  - Web, email, Quake, e-commerce, audio/video, …
  - But evolution was at work: Web/email a "surprise"
- Computer networks
  - Carry digital information and support a rich variety of distributed applications

Why Build Networks?

- Communication at a distance
  - Want performance sufficient to given task
    - Video conference, etc.

- Resource sharing
  - Networks are shared among users
  - Fundamental issues concern the effective sharing of distributed resources
    - Effective = cost, control, secure, reliable, …

Goal of this Course

- For you to understand the design of *large-scale computer* networks.

- Fundamental problems in building networks
  - That are fast, efficient, secure and robust
- Design principles of proven value
  - Networking is young and there are few!
- Common implementation technologies
  - These will change of course …

Lecture Topics

- Multi-access (Ethernet)
- Routing (IP)
- Transport (TCP)
- Congestion control (TCP)
- Multicast (Mbone)
- Real time (DiffServ)
- Naming (DNS)
- Security (IPSEC)

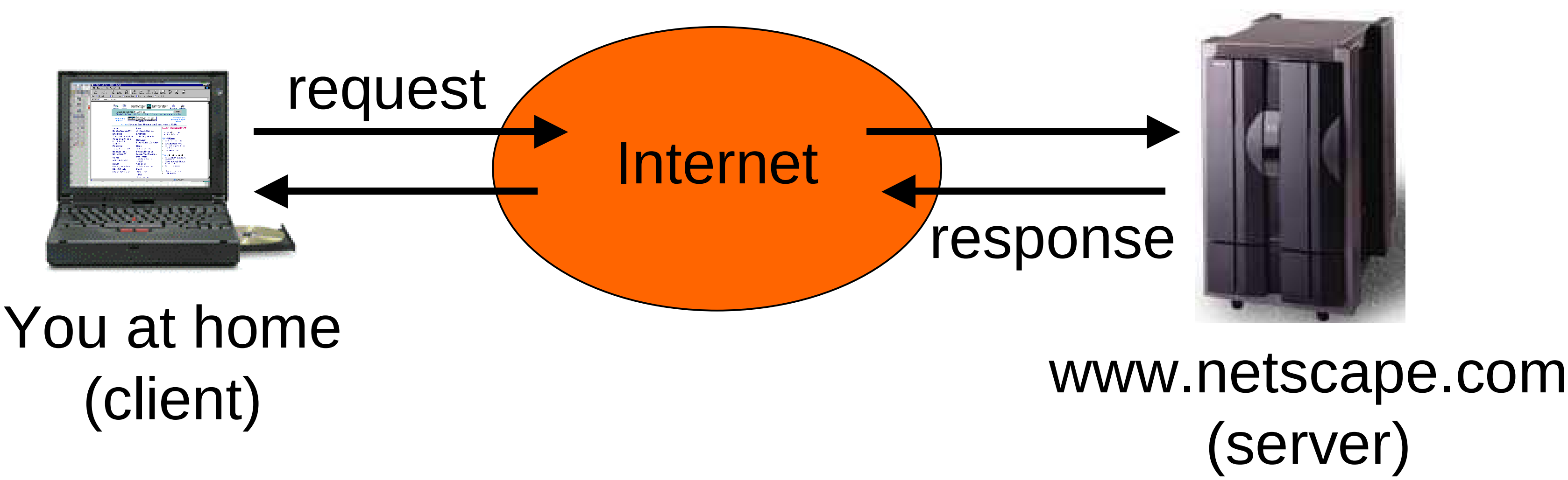
- Coordination
- Robust operation
- Reliable delivery
- Resource allocation
- Efficient delivery
- Multimedia
- Distributed state
- Authentication/Privacy

Lecture Emphasis

- What we do cover:
  - Communications
  - Internetworking ← We focus here
  - Distributed systems
- What we don't cover:
  - Design of communications hardware
  - Queuing theory
  - Protocol standards

3. A Brief Tour of the Internet

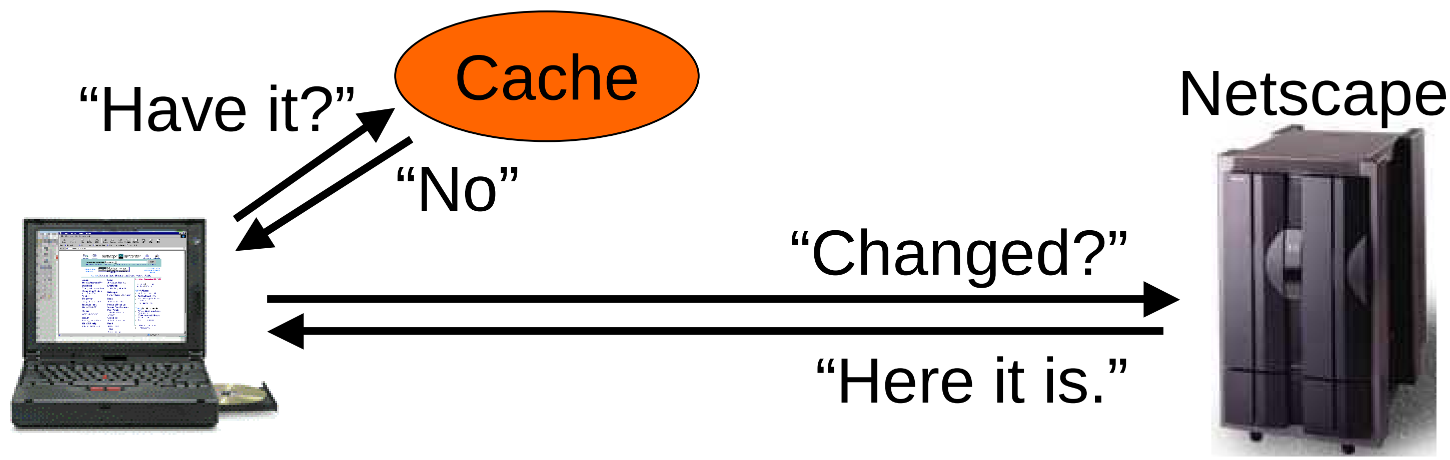
- What happens when you “click” on a web link?



- This is the view from 10,000 ft …

9,000 ft: Scalability

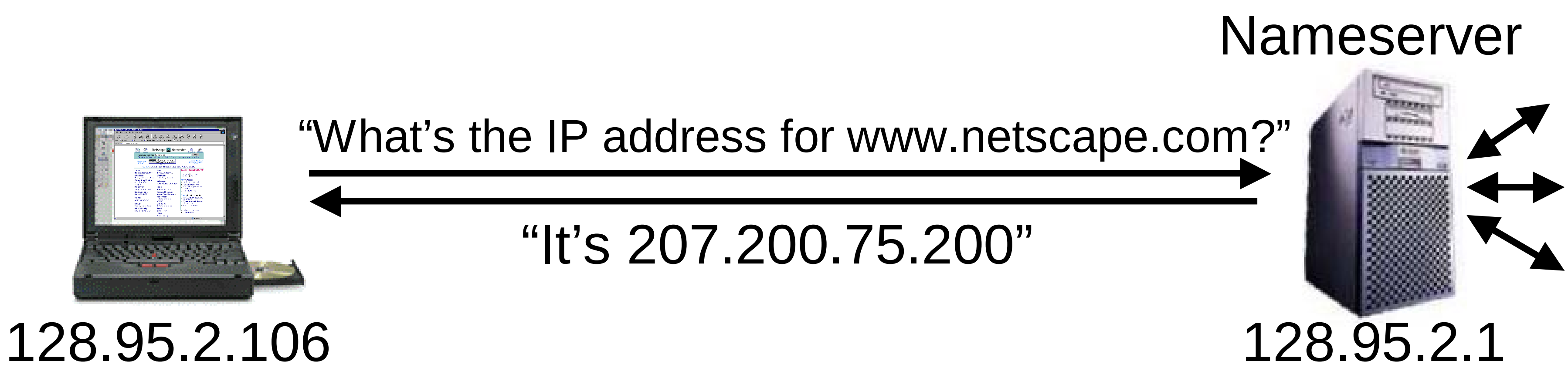
- Caching improves scalability



- 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)

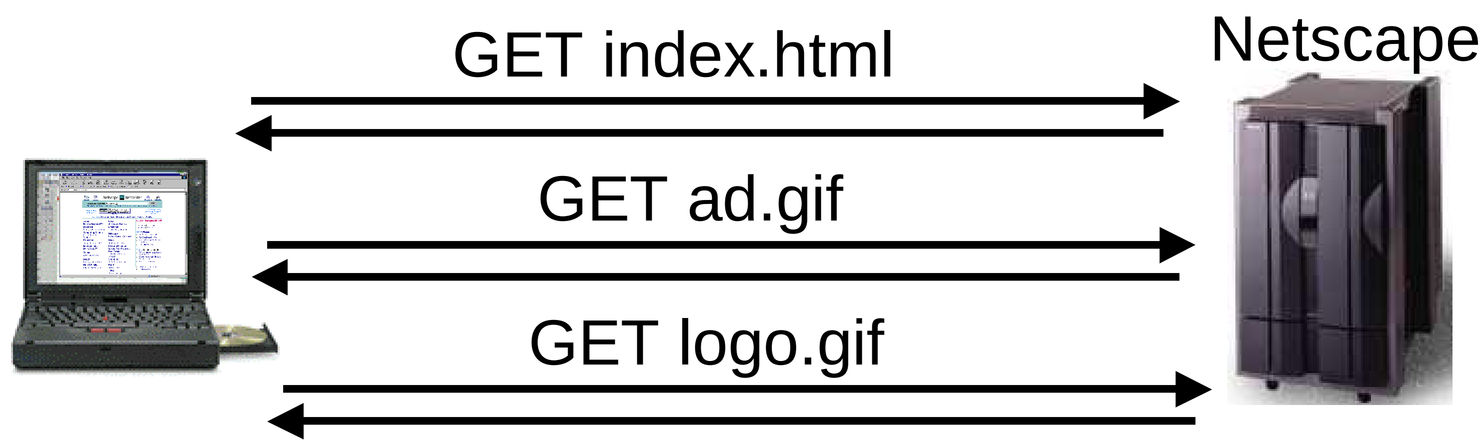
- Map domain names to IP network addresses



- All messages are sent using IP addresses
  - So we have to translate names to addresses first
  - But we cache translations to avoid next time

7,000 ft: Sessions (HTTP)

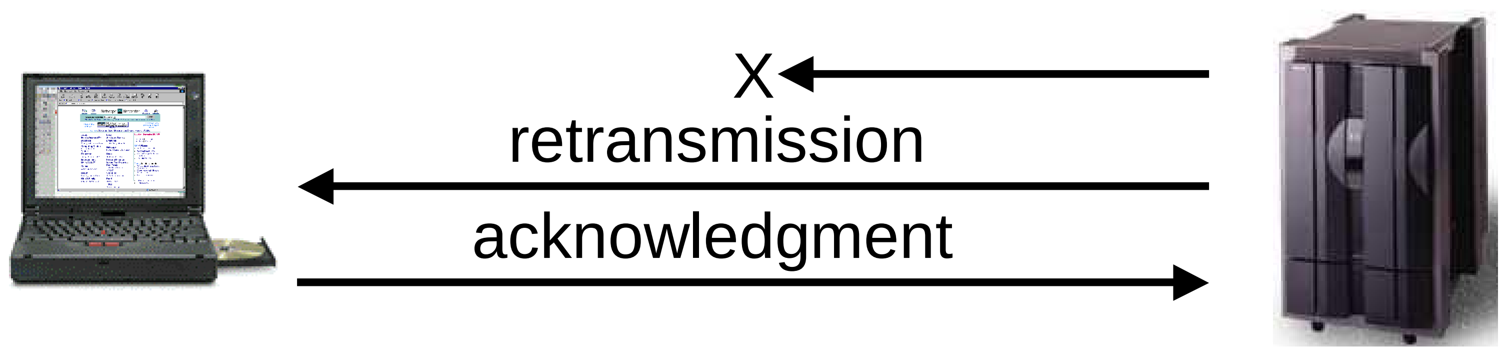
- A single web page can be multiple "objects"



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

6,000 ft: Reliability (TCP)

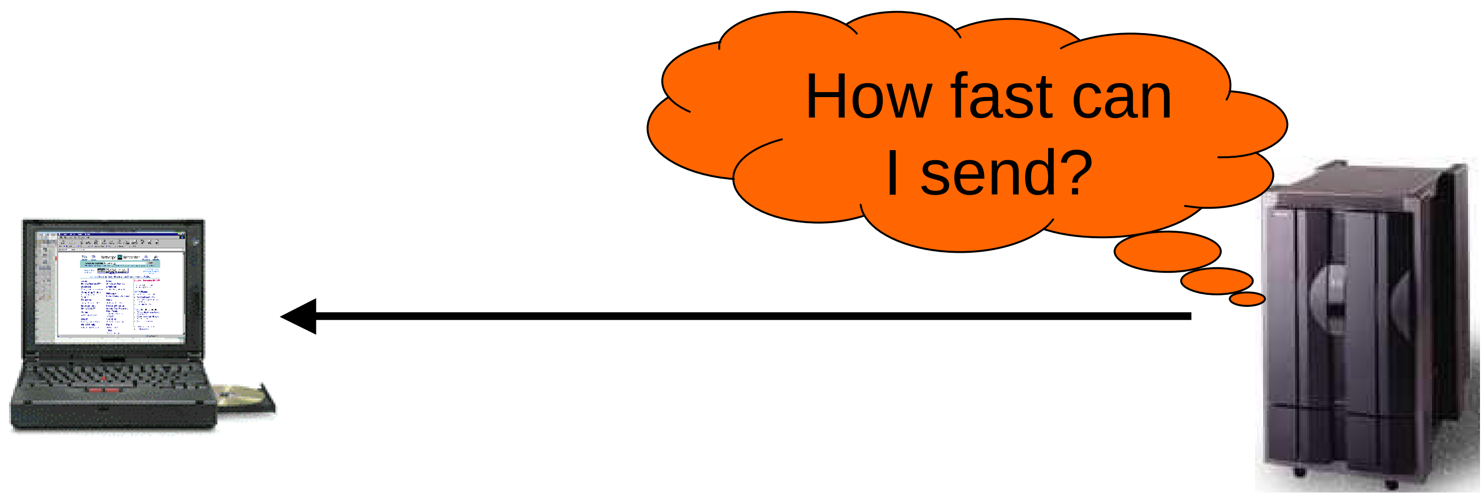
- Messages can get lost



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

5,000 ft: Congestion (TCP)

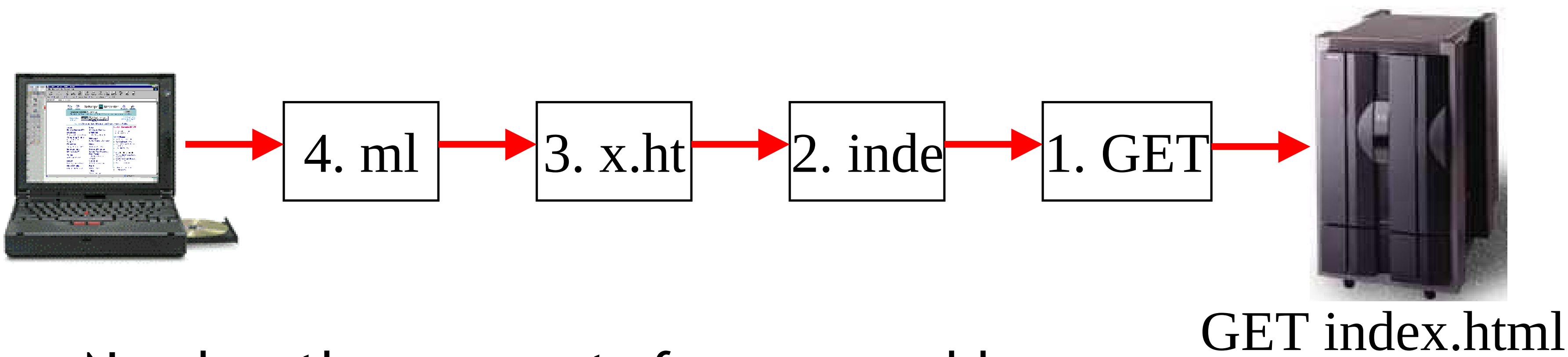
- Need to allocate bandwidth between users



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

4,000 ft: Packets (TCP/IP)

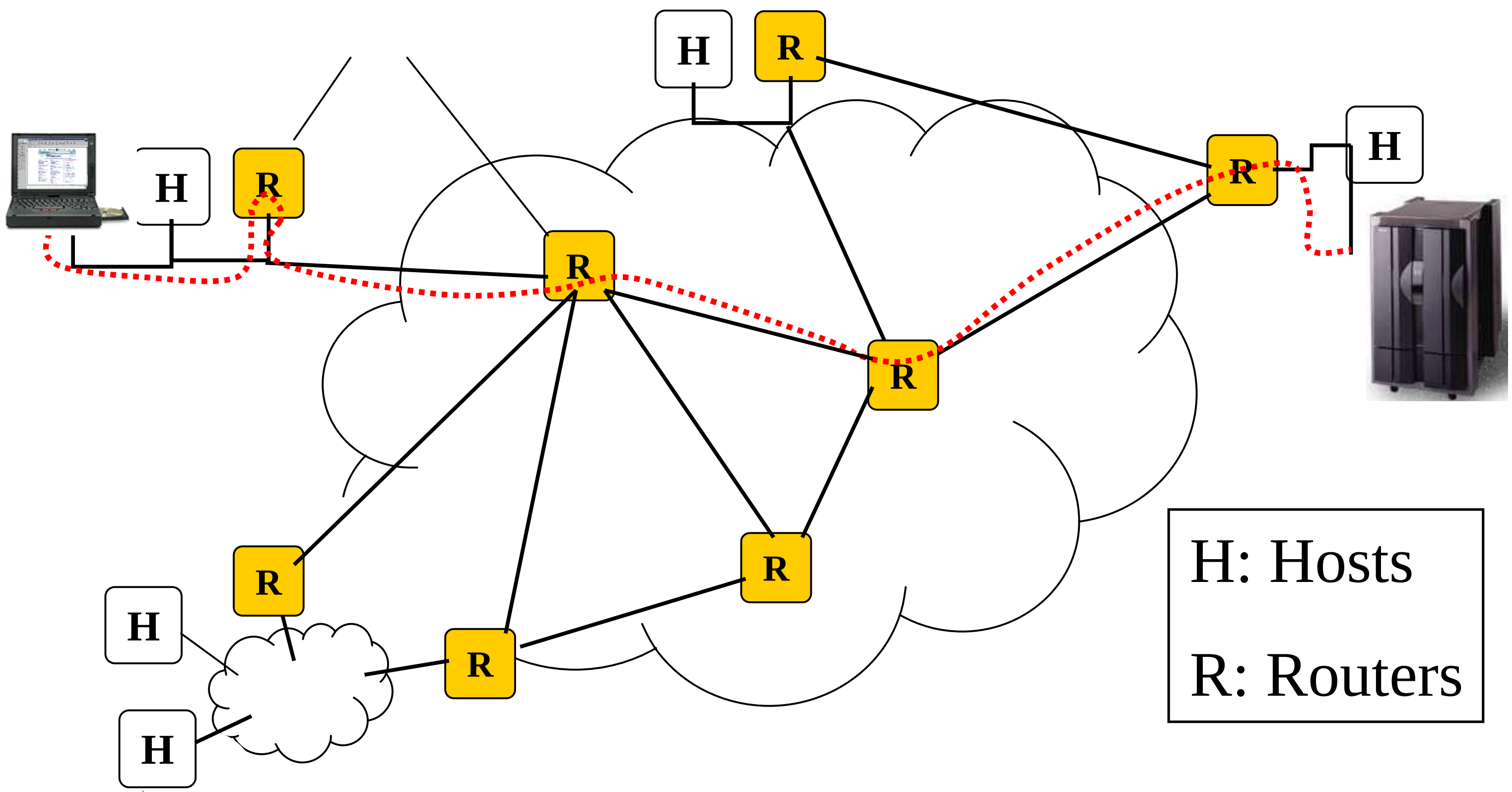
- Long messages are broken into packets
  - Maximum Ethernet packet is 1.5 Kbytes
  - Typical web page is 10 Kbytes



- Number the segments for reassembly

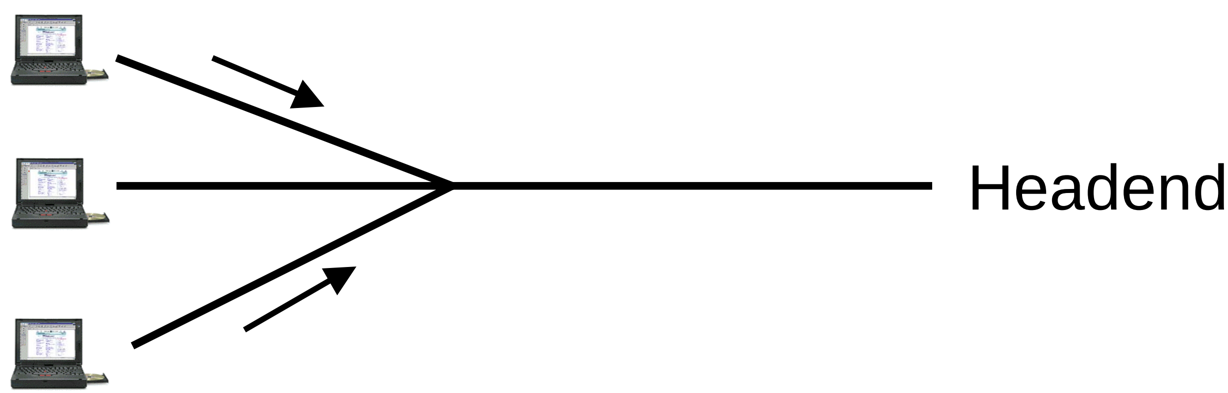
3,000 ft: Routing (IP)

- Packets are directed through many routers



2,000 ft: Multi-access (Cable)

- May need to coordinate packet transmissions



- Poll headend to receive a timeslot to send upstream
  - Headend controls all downstream transmissions
  - A lower level of addressing is used …

1,000 ft: Framing/Modulation

- Protect, delimit and modulate payload

Sync / Unique	Header	Payload w/ error correcting code

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

4. Protocols and Layering

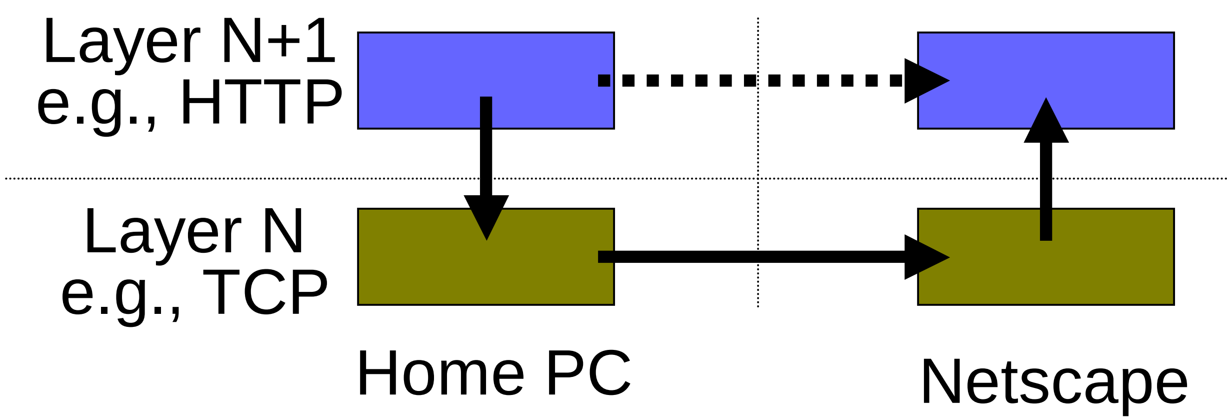
- Need abstractions to handle complexity
  - Protocols and layering
- Protocol
  - Agreement dictating the form and function of data exchanged between parties to effect communication
  - Two parts:
    - Syntax: where the bits go
    - Semantics: what they mean, what to do with them
  - Examples:
    - IP, the Internet protocol
    - TCP and HTTP, for the Web

Protocol Standards

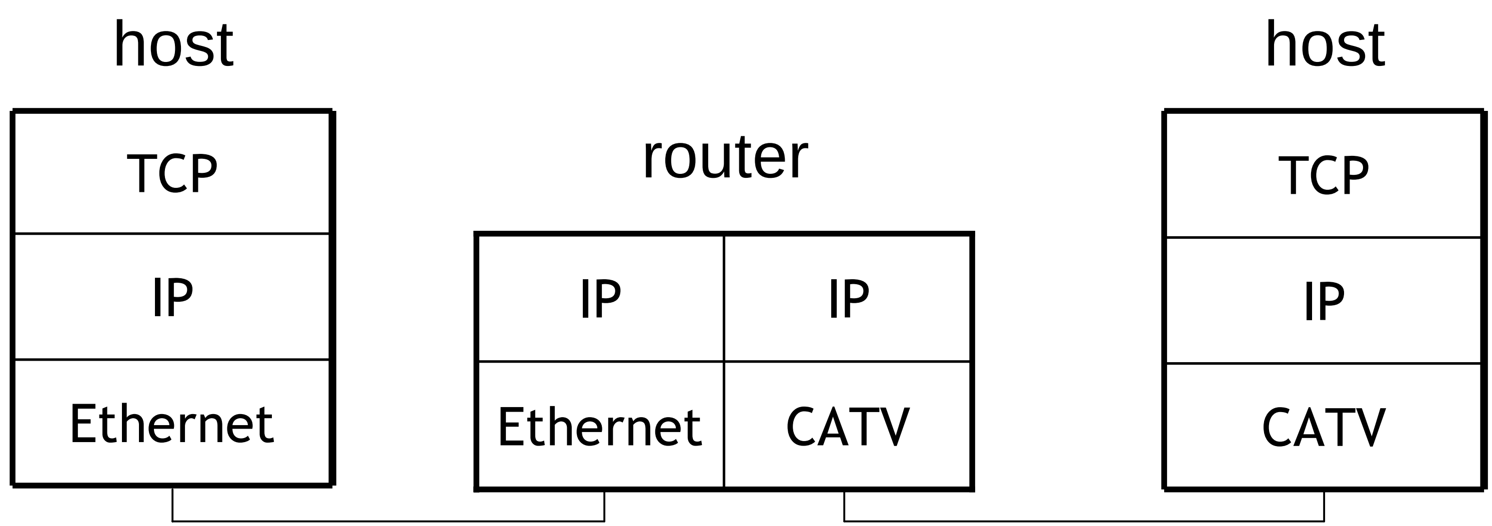
- Different functions require different protocols
- Thus there are many protocol standards
  - E.g., IP, TCP, UDP, HTTP, DNS, FTP, SMTP, NNTP, ARP, Ethernet/802.3, 802.11, RIP, OPSF, 802.1D, NFS, ICMP, IGMP, DVMRP, IPSEC, PIM-SM, BGP, …
- Organizations: IETF, IEEE, ITU
- IETF specifies Internet-related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

Protocol Layering

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



Example – Layering at work

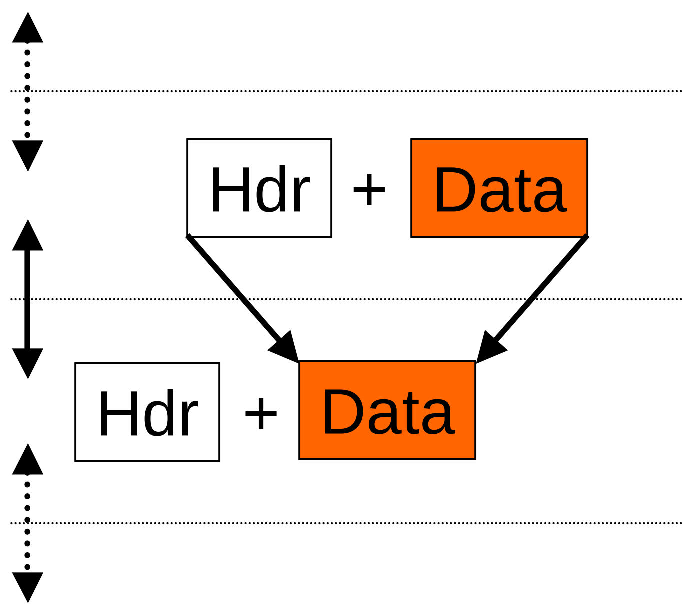


- We can connect different systems

Layering Mechanics

- Encapsulation and decapsulation

Messages passed between layers



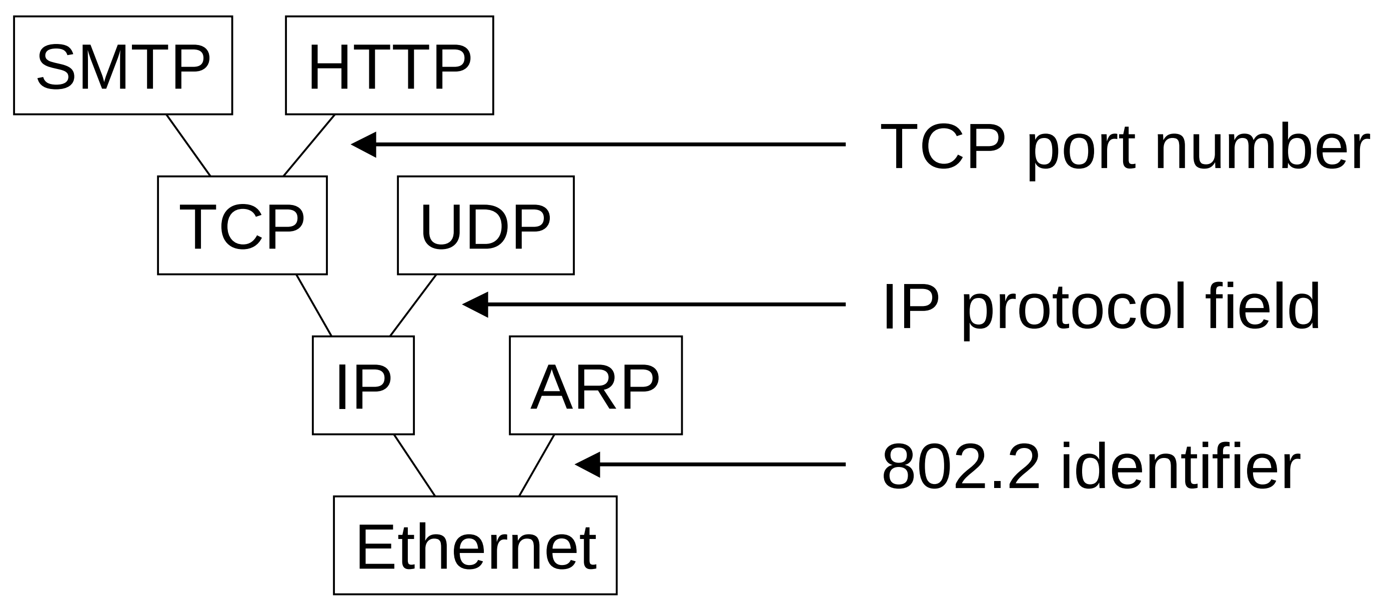
Layer N+1 PDU

becomes

Layer N ADU

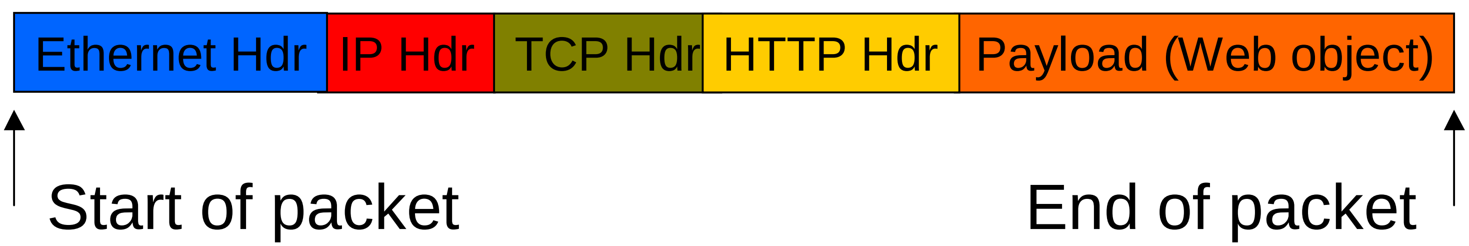
More Layering Mechanics

- Multiplexing and demultiplexing in a protocol graph



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

Internet Protocol Stacks

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

OSI Reference Model

- Seven Layers

Application
Presentation
Session
Transport
Network
Link
Physical

Their functions:

- Your call
- Encode/decode messages
- Manage connections
- Reliability, congestion control
- Routing
- Framing, multiple access
- Symbol coding, modulation

Key Concepts

- Networks are used to share distributed resources
- Protocol layers are used to handle complexity
- The Internet/OSI models give us a roadmap

- Next time: We start working our way up the layers from the bottom, beginning with how to send bits.