

CSC 425: Computer Networks

Logistics

- Lecture: Mon & Wed, 2:00 – 3:15 PM, GS 906
- Instructor: Beichuan Zhang
 - bzhang@cs.arizona.edu
 - Office Hour: Friday 3-4 PM, Gould-Simpson 723
- Teaching Assistant:
 - Md Ashiqur Rahman
 - marahman@email.arizona.edu
 - Office Hour: Thu 4-5pm, Gould-Simpson 934
 - Ju Pan
 - pjokk722@email.arizona.edu
 - Office Hour: Tue 4-5pm, Gould-Simpson 934
- D2L for course material, and Piazza for discussion.

What this course is about

- Understand how today's network works.
 - within the dept, across campus, across the globe
 - from the wire to the applications.
 - Basic concepts and essential protocols
- Basic network programming
 - C socket programming
 - Network packet processing

What this course is NOT about

- Network hardware
- Queuing theory
- Cisco Certification
- ...

Prerequisite and Textbook

- CSc 352 or equivalent
 - Comfortable with C programming
- Required Textbook
 - Computer Networks, a system approach, 5th ed, by Peterson and Davie.
 - Free online version available through D2L/Library

Course Workload

- Reading Assignment
 - cover all chapters except chapter 7
 - different depth for different chapters
- Homework
 - 4 to 6 assignments
- Midterm and Final Exams
 - based on the textbook and lectures.
- Two projects
 - socket programming
 - packet processing

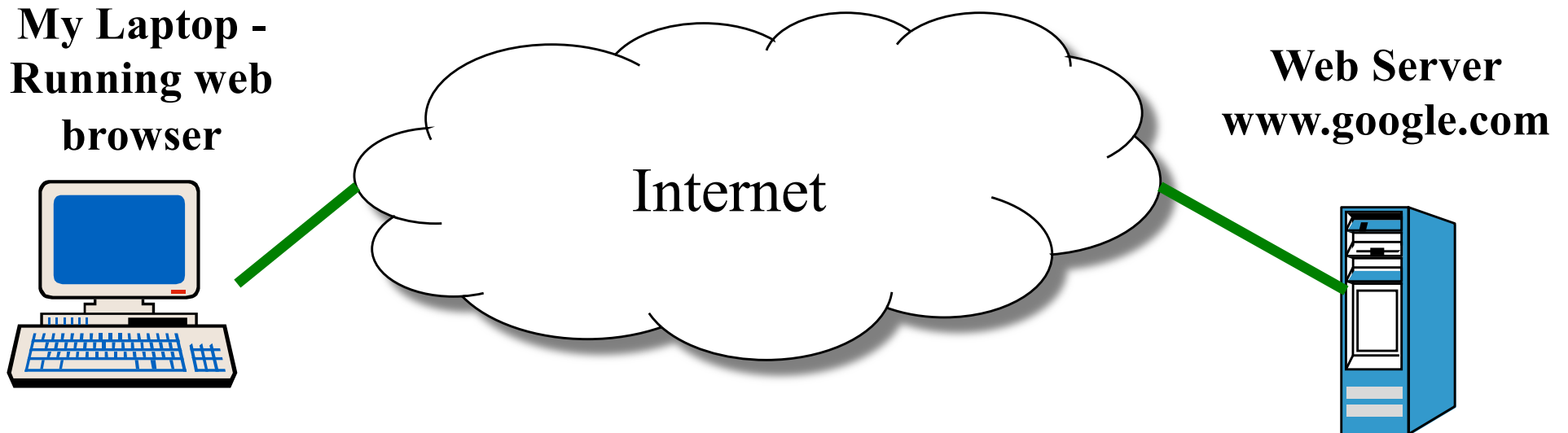
Grading and Policy

- Grading
 - Projects (20%, 20%)
 - Final 25%
 - Midterm 25%
 - Homework 10%
- Grading Policy
 - No credit for late homework, late projects, or missed exams.
 - Talk to the instructor in advance.
 - Partial credits for work incomplete but submitted on time.

SOME BASIC CONCEPTS

Host

Simple Example: Use web browser to visit www.google.com



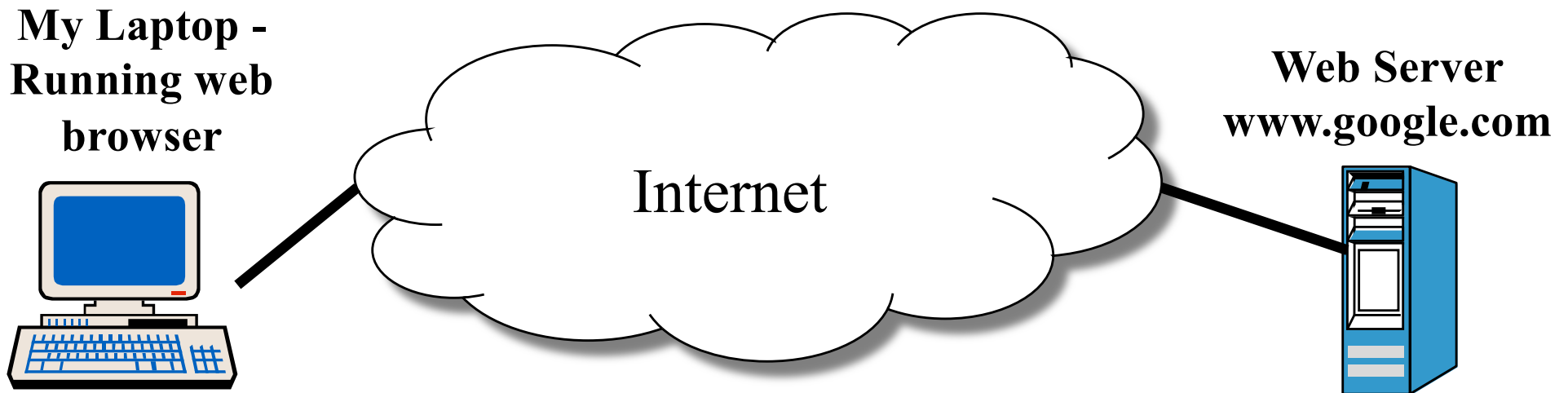
My laptop and the web server are both **End Systems = Hosts**

End systems include generally any user device using the network

End systems are located at the **network edge and connected to the network using **communication links****

Client and Server

Simple Example: Use web browser to lookup www.google.com



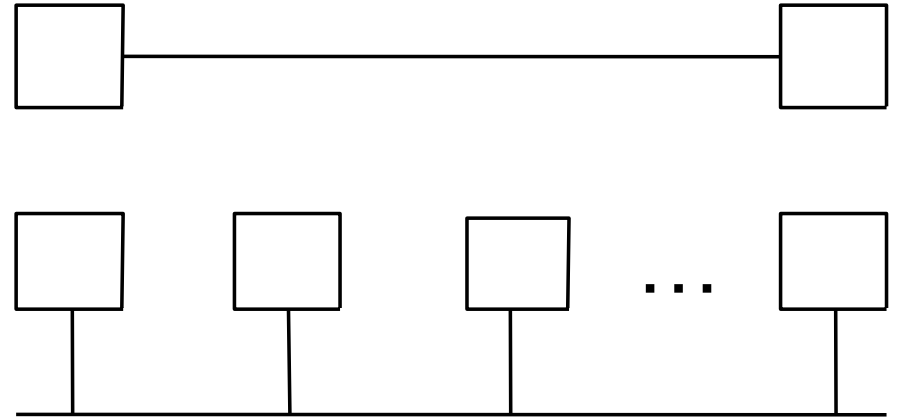
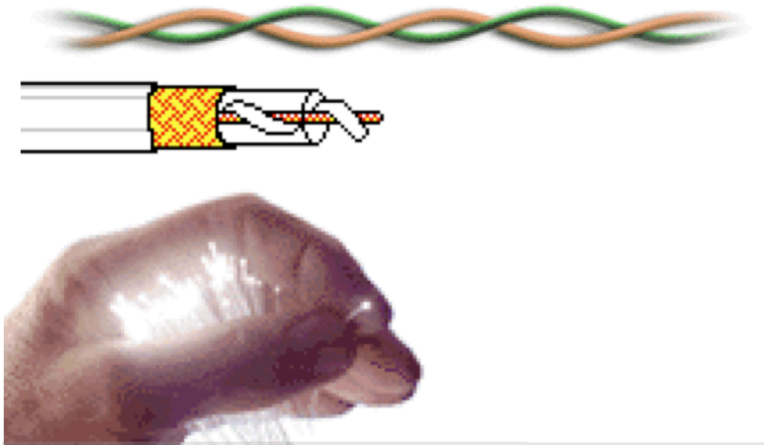
End systems can be classified as a **client**, a **server**, or **both** for a given app.

Client - *a program* that requests services: web browser

Server - *a program* that serves the contents: web server

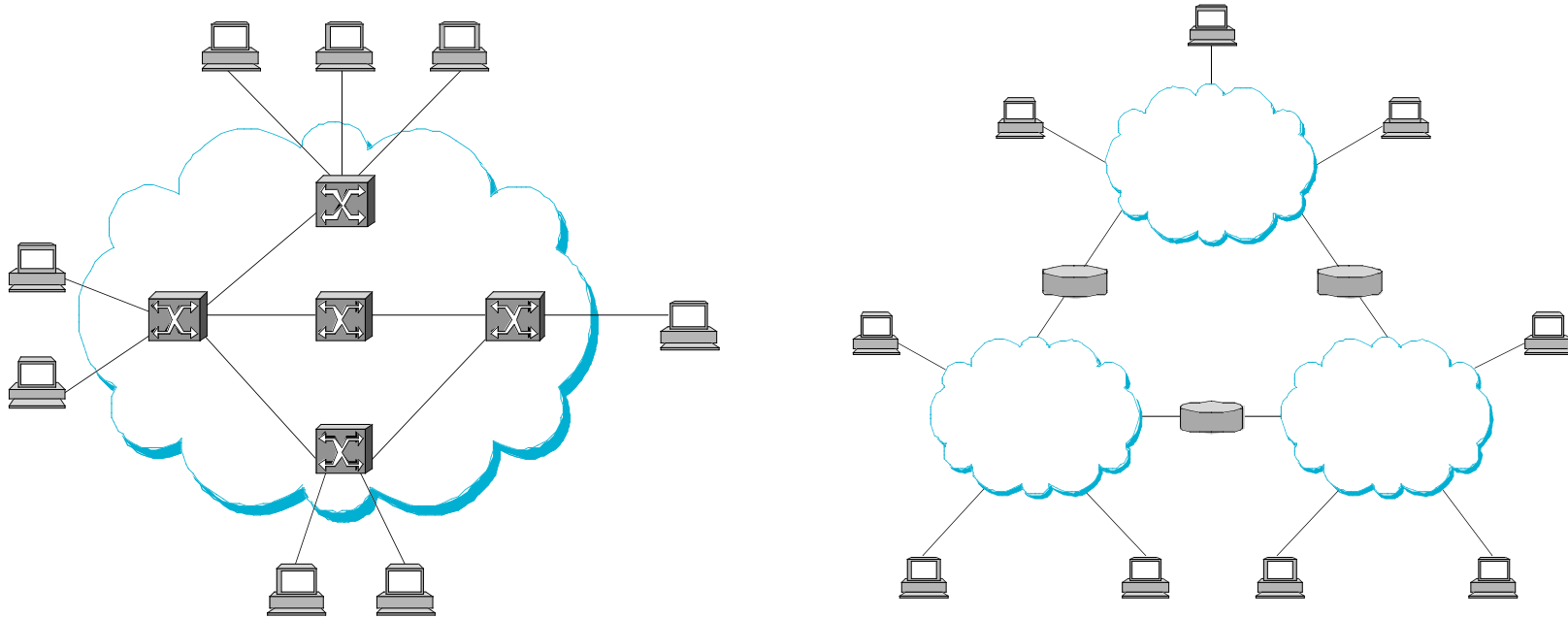
Client and server are application specific. A host can run an email client and a web server at the same time.

Link

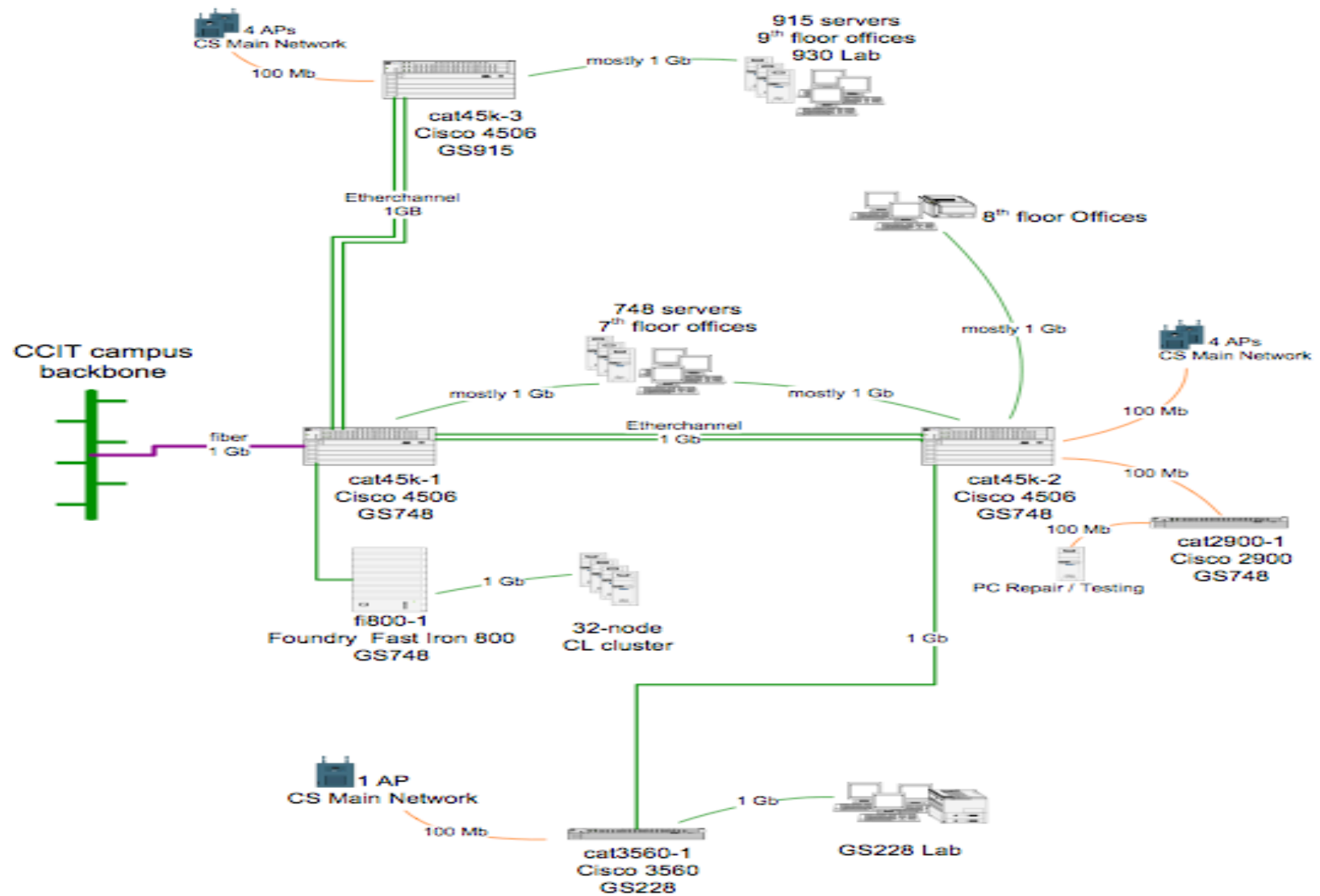


- Links connect one device directly to another.
 - Wired: twisted pair, coaxial cable, fiber ...
 - Wireless: wifi, infrared, microwave, satellite, ...
- Links can be either ***dedicated*** (used by only two nodes) or ***shared*** (used by more than 2 nodes).
- Links have properties such as bandwidth, delay, maximum transmission unit (MTU), power consumption, etc.

Network Devices



- Connect hosts in a scalable and economic way
- Network devices at different levels with different functionality
 - Hubs, switches, routers, ...

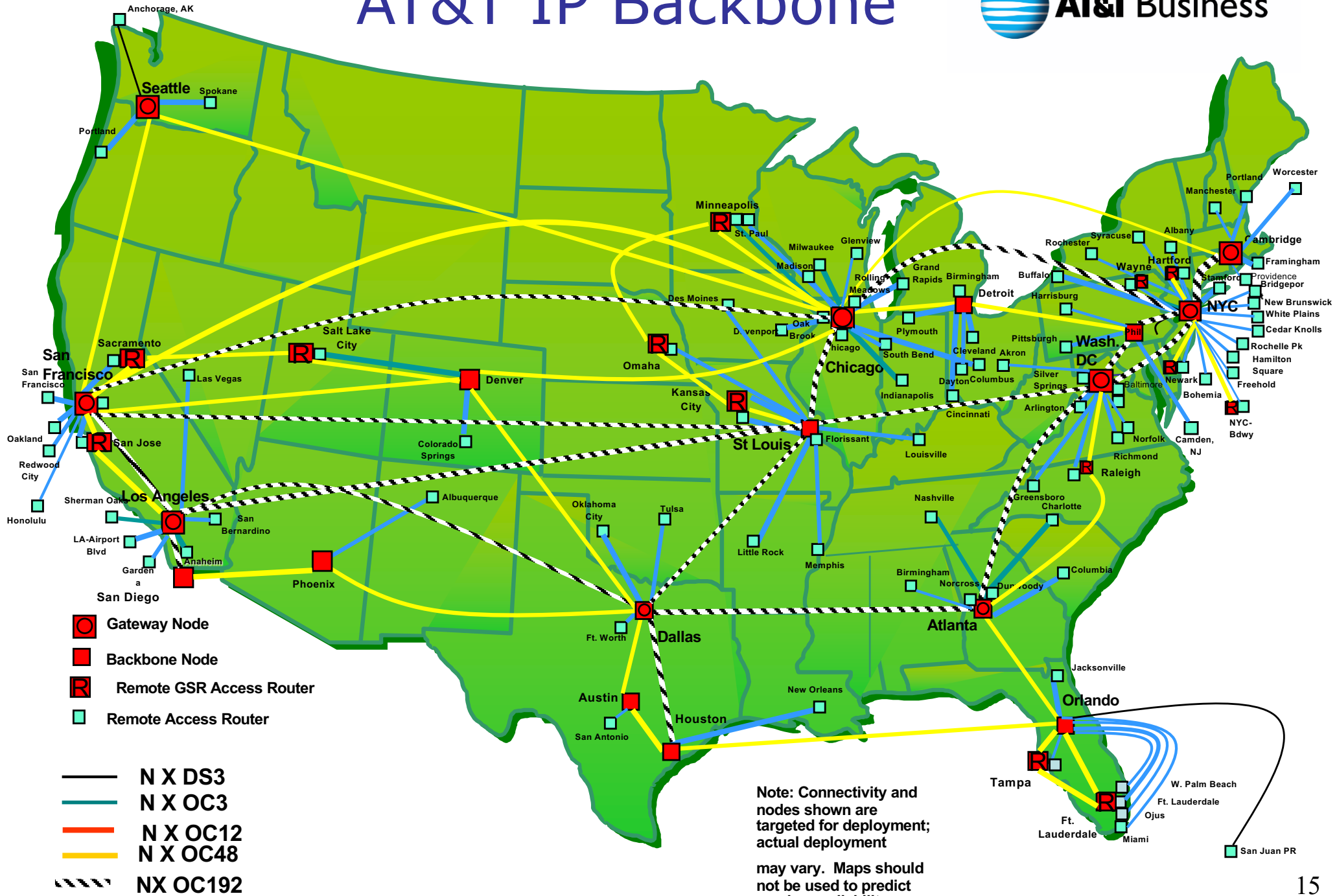


Computer Science Department
Network Diagram
September 2008

10/100/1000 Mb
10/100 Mb



AT&T IP Backbone



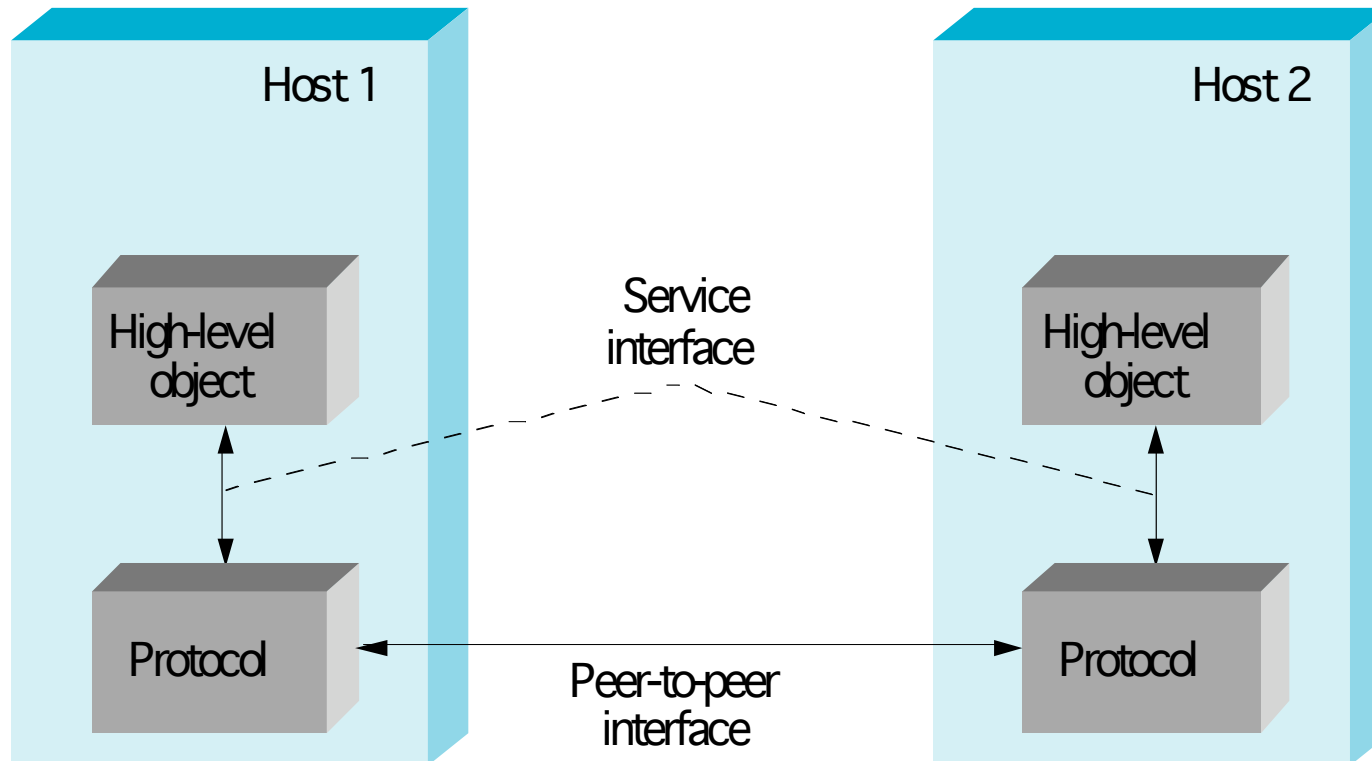
Protocol

- Network communication consists of both hardware and software.
 - Hardware: hosts, routers, switches, links
 - Software: protocols.
- Protocols are the codes and languages between communication parties.
 - Web browser and web server talk in HTTP
 - Routers talk routing protocols between each other, e.g., OSPF.

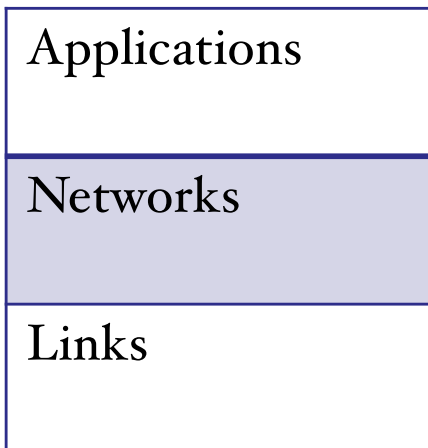
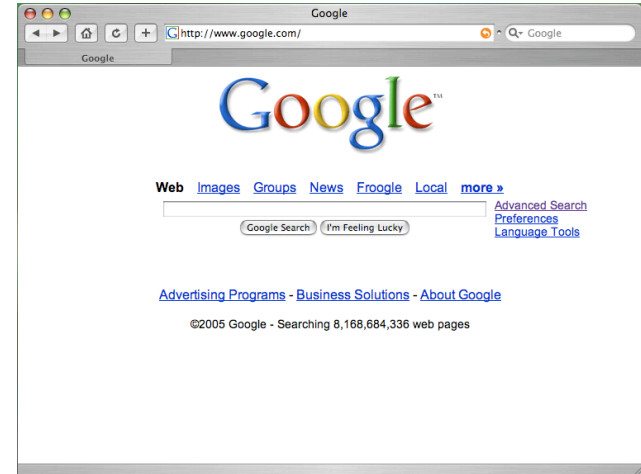
Network Protocols

- Each protocol has two different interfaces
 - *service interface*: how upper-layer entity uses this protocol.
 - *communication interface*: message exchange with other peers running the same protocol.

Protocol Interfaces



What does the network do?

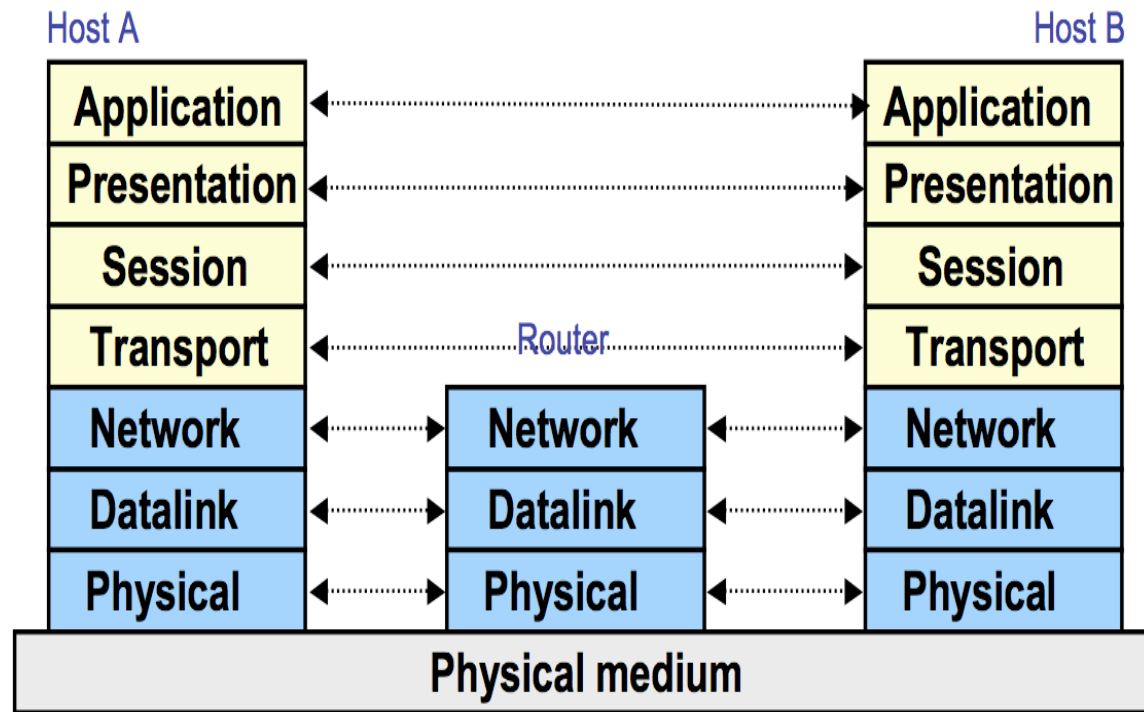


- End-point identification
- Content delivery
 - Fault resiliency
 - Quality of service
- Interface with apps and links

The Approach

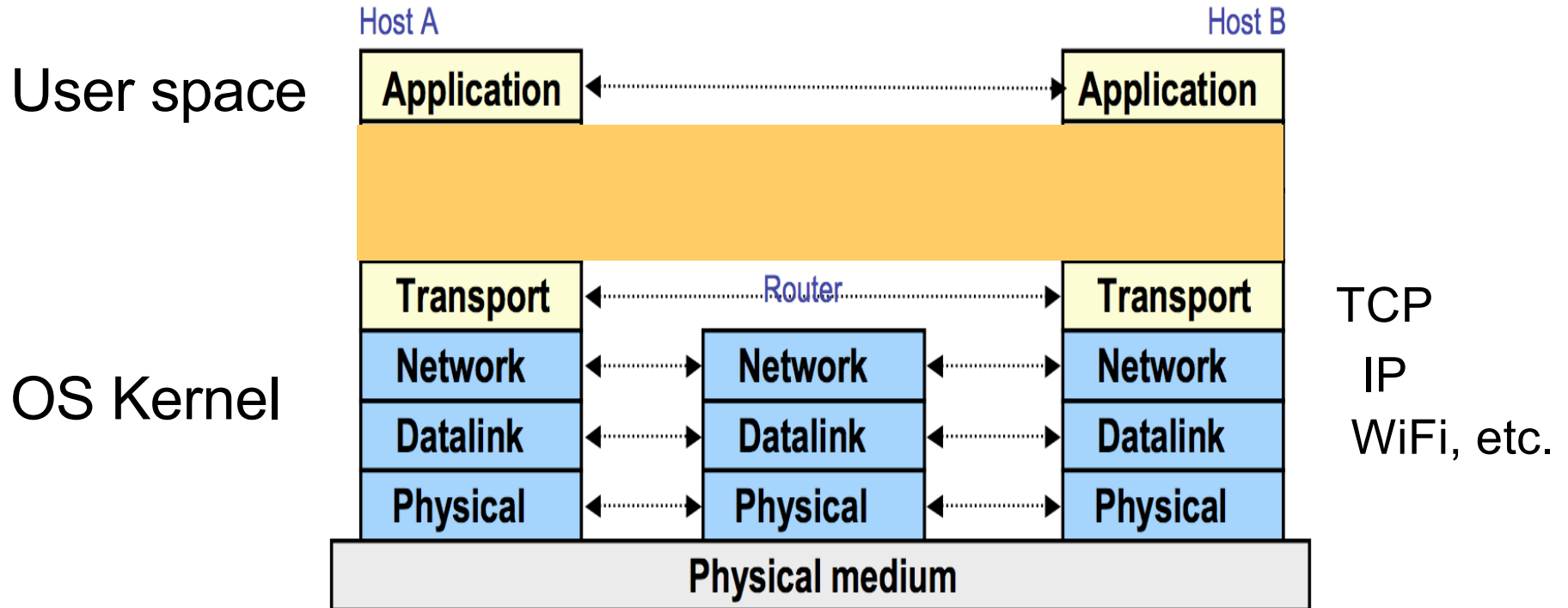
- Divide and Conquer
 - Break the whole problem into smaller parts, solve each part separately, then put them back together.
- Network Layering
 - Divide needed functionality into multiple layers (modules)
 - Lower layers implement certain functions and provide interfaces.
 - Upper layers use the functions through the interfaces, don't care about implementation details.

The OSI Reference Model



- Exists mostly on paper

The Internet Model



- Exists in real systems

Application Layer

- Carry out application-level conversations
- Uses the transport layer to send/receive messages.

HTTP Request:

*protocol
header*

```
POST /index.html HTTP/1.1
Host: map.google.com
User-Agent: Mozilla/4.0
Content-Length: 22
Accept-language: en
```

*Application
data*

```
City=Tucson&State=AZ
```

HTTP Response:

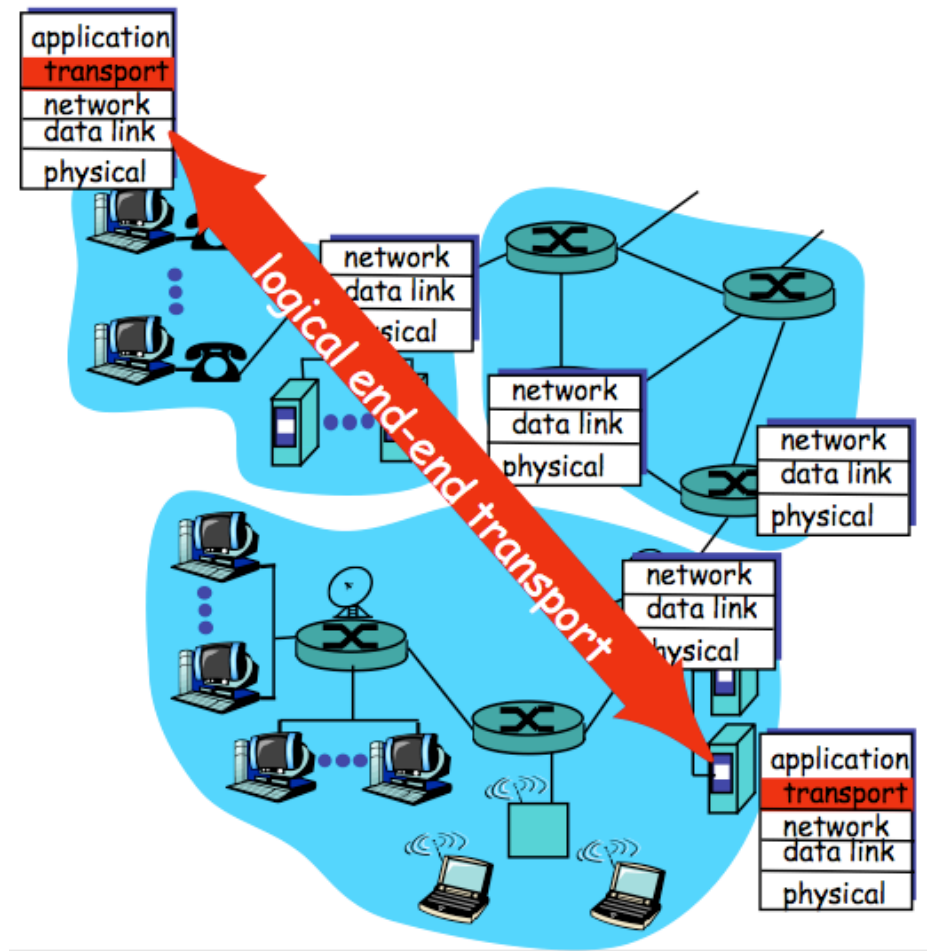
```
HTTP/1.1 200 OK
Date: Tue, 04 Mar 2003 08:01:01 GMT
Server: GWS/2.0
Content-Length: 2824
Content-Type: text/html
```

```
<HTML> ... </HTML>
```

- E.g.. HTTP protocol specifies the headers and actions.

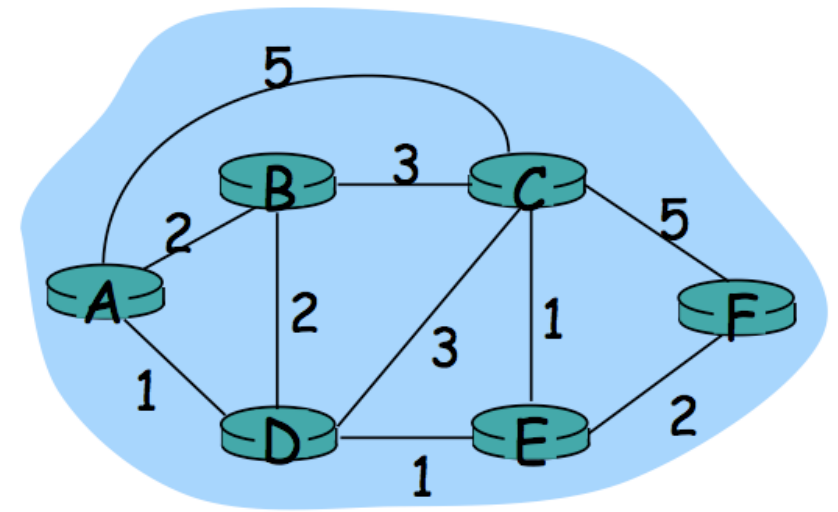
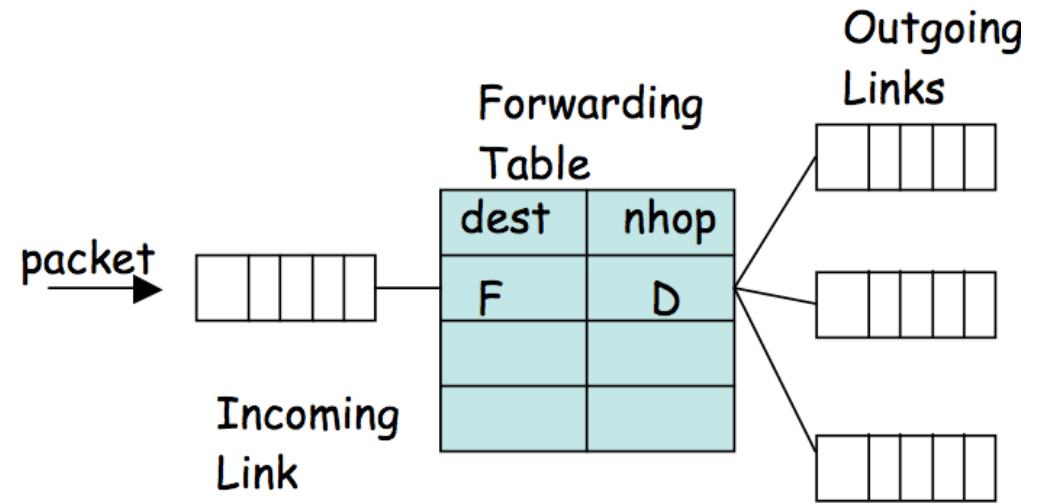
Transport Layer

- Anything can happen to messages in transit.
 - Transmission error
 - Equipment failure
 - Congestion
- Transport Layer provides an end-to-end connection and data delivery service through that connection.
 - E.g., TCP, UDP.



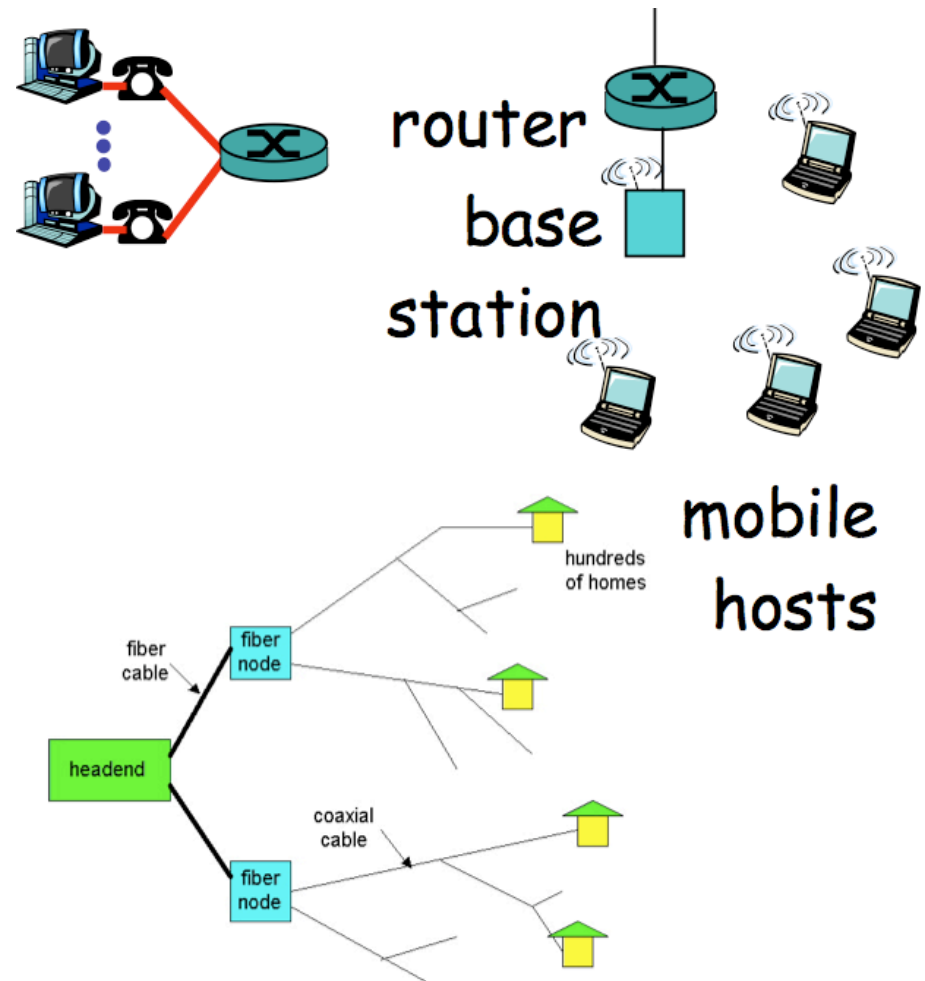
Network Layer

- Best-effort packet delivery from source host to destination host.
- How to find a good path?
- How to forward packets as fast as possible?
- Involve many ISPs and local networks.



Data Link Layer

- Transmit/Receive messages over a single link.
 - Framing
 - Error detection & correction
 - Access sharing
- Example
 - Dialup, DSL
 - Cable Modem
 - WiFi, CDMA, GSM



Physical Layer

- Transmit bits on physical media
 - Twisted pair
 - Coaxial cable
 - Fiber optics
 - Radio link
- Bandwidth
- Dedicated vs. shared



Internet's Hourglass Architecture

- The key is a single, minimal network layer, IP.
- Support diverse transport and applications
- Use diverse physical link technologies.

