

CS155b: E-Commerce

Lecture 2: Jan. 16, 2003

How Does the Internet Work?

Acknowledgements: S. Bradner and R. Wang

Internet Protocols Design Philosophy

- Ordered set of goals:
 1. multiplexed utilization of **existing networks**
 2. survivability in the face of failure
 3. support multiple types of communications service
 4. accommodate a variety of network types
 5. permit distributed management of resources
 6. cost effective
 7. low effort to attach a host
 8. account for resources
- Not all goals have been met

Packets!

- Basic decision: use packets not circuits (Kleinrock)
- Packet (*a.k.a.* datagram)



- self contained
- handled independently of preceding or following packets
- contains destination and source **internetwork** address
- may contain processing hints (*e.g.*, QoS tag)
- **no delivery guarantees**
 - net may drop, duplicate, or deliver out of order
 - reliability (where needed) done at higher levels

Telephone Network

- Connection-based
- Admission control
- Intelligence is "in the network"
- Traffic carried by relatively few, "well-known" communications companies

Internet

- Packet-based
- Best effort
- Intelligence is "at the endpoints"
- Traffic carried by many routers, operated by a changing set of "unknown" parties

Technology Advances

	1981	1999	Factor
MIPS	1	1000	1,000
\$/MIPS	\$100K	\$5	20,000
DRAM Capacity	128KB	256MB	2,000
Disk Capacity	10MB	50GB	5,000
Network B/W	9600b/s	155Mb/s	15,000
Address Bits	16	64	4
Users/Machine	10s	≤ 1	< 0.1

- Expensive machines, cheap humans
- Cheap machines, expensive humans
- (Almost) free machines, really expensive humans, and communities

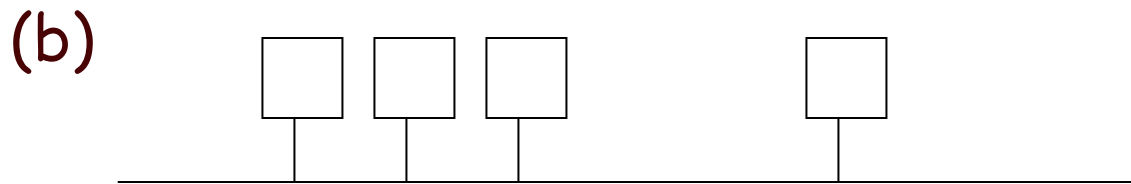
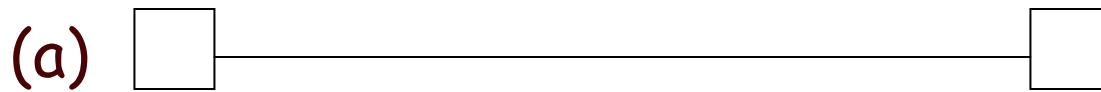
The Network *is* the Computer

- Relentless decentralization
 - "Smaller, cheaper, more numerous"
mainframe → mini → PC → palms → ubiquitous/embedded
 - More computers → more data communication
- (Shifting) reasons computers talk to each other
 - Efficient sharing of machine resources
 - Sharing of data
 - Parallel computing
 - *Human* communication

The Network *is* the computer (continued)

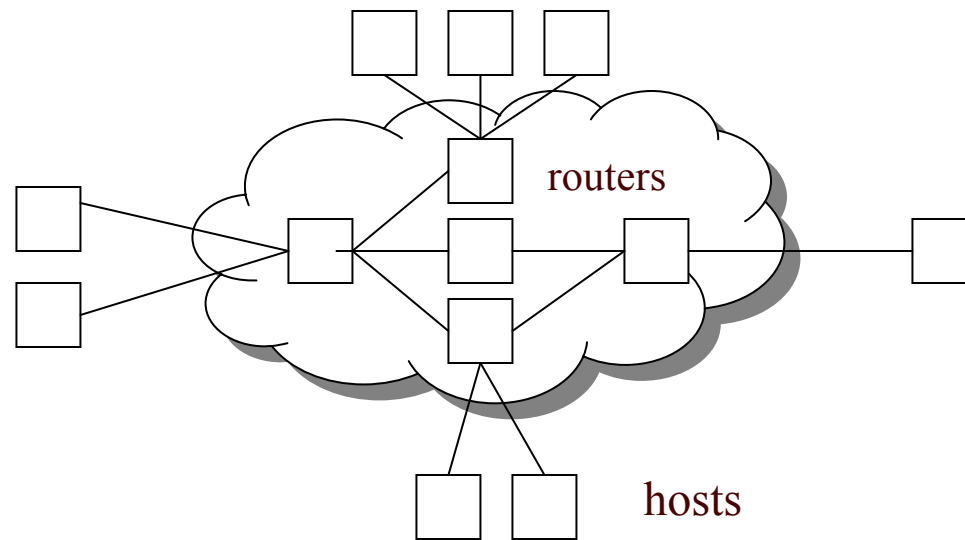
- Networks are everywhere and they are converging
 - SAN, LAN, MAN, WAN
 - All converging towards a similar technology
 - Sensor nets
- New chapter of every aspect of computer science
 - Re-examine virtually all the issues in the context of distributed systems or parallel systems
- This is only the beginning.

Directly Connected



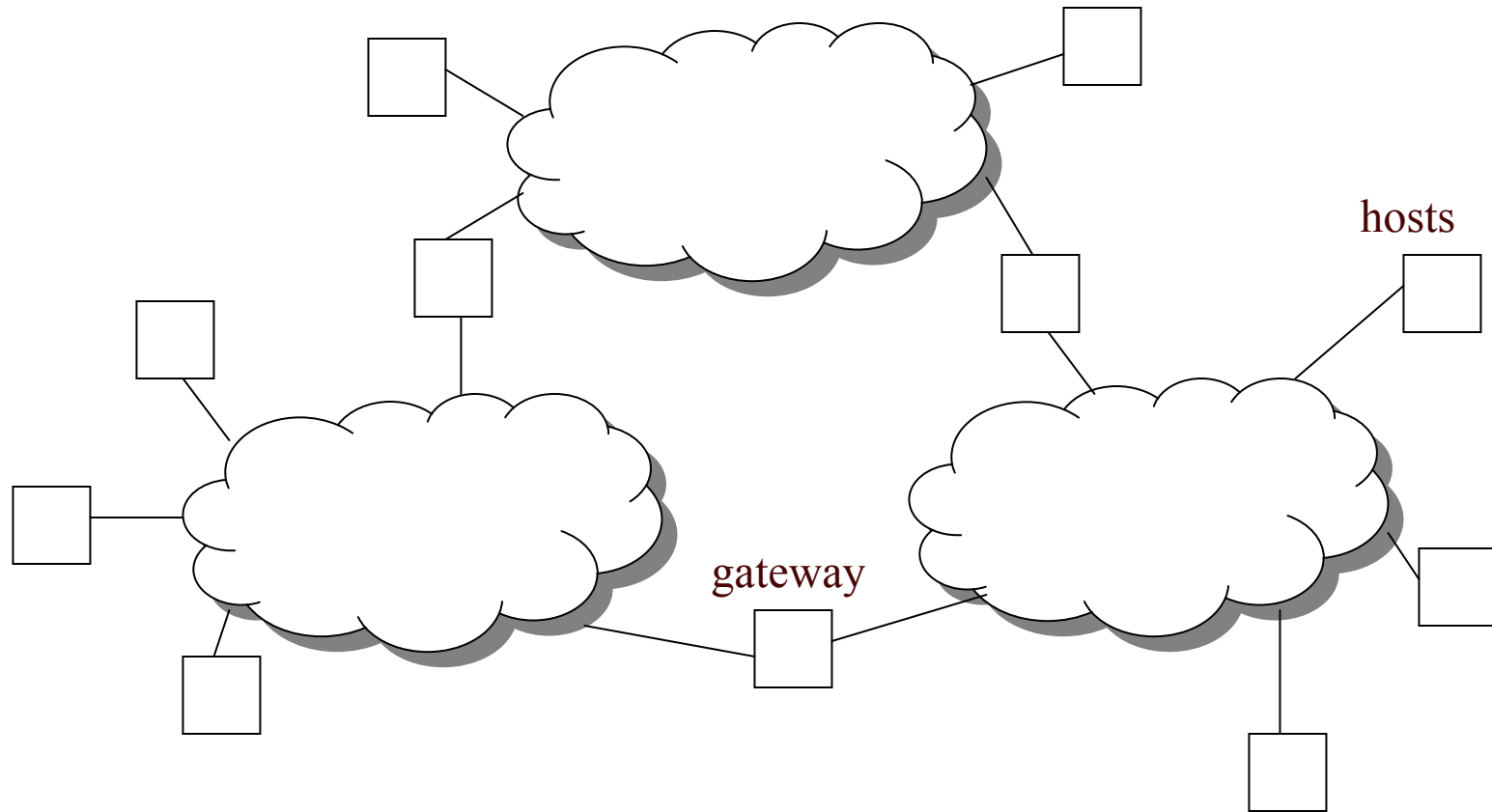
- (a) Point-to-point: e.g., ATM
- (b) Multiple-access: e.g., Ethernet
- Can't build a network by requiring *all* nodes to be directly connected to each other; need scalability with respect to the number of wires or the number of nodes that can attach to a shared medium

Switched Network



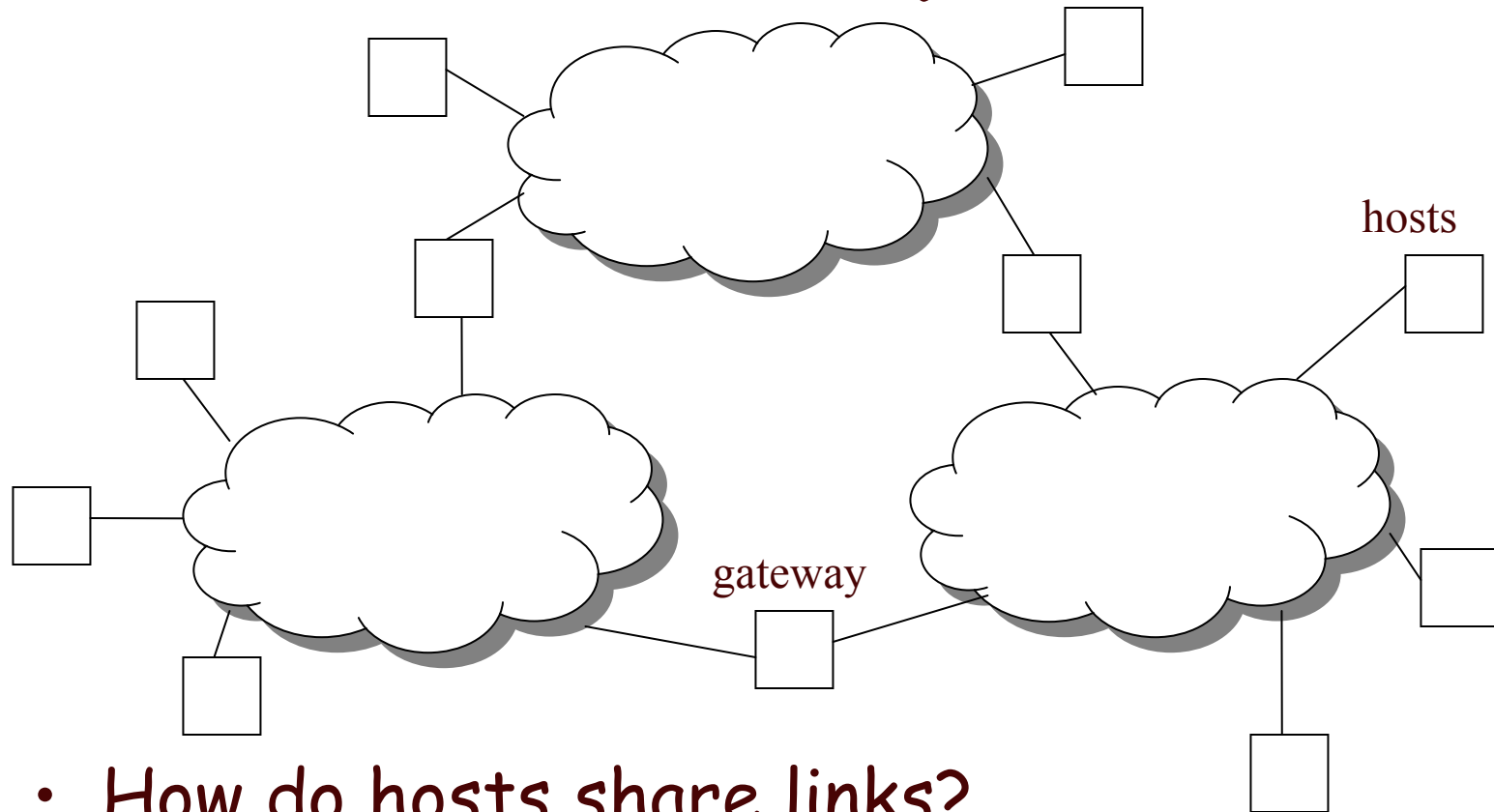
- Circuit switching vs. packet routing
- Hosts vs. "the network," which is made of routers
- Nice property: scalable aggregate throughput

Interconnection of Networks



Recursively build larger networks

Some Hard Questions

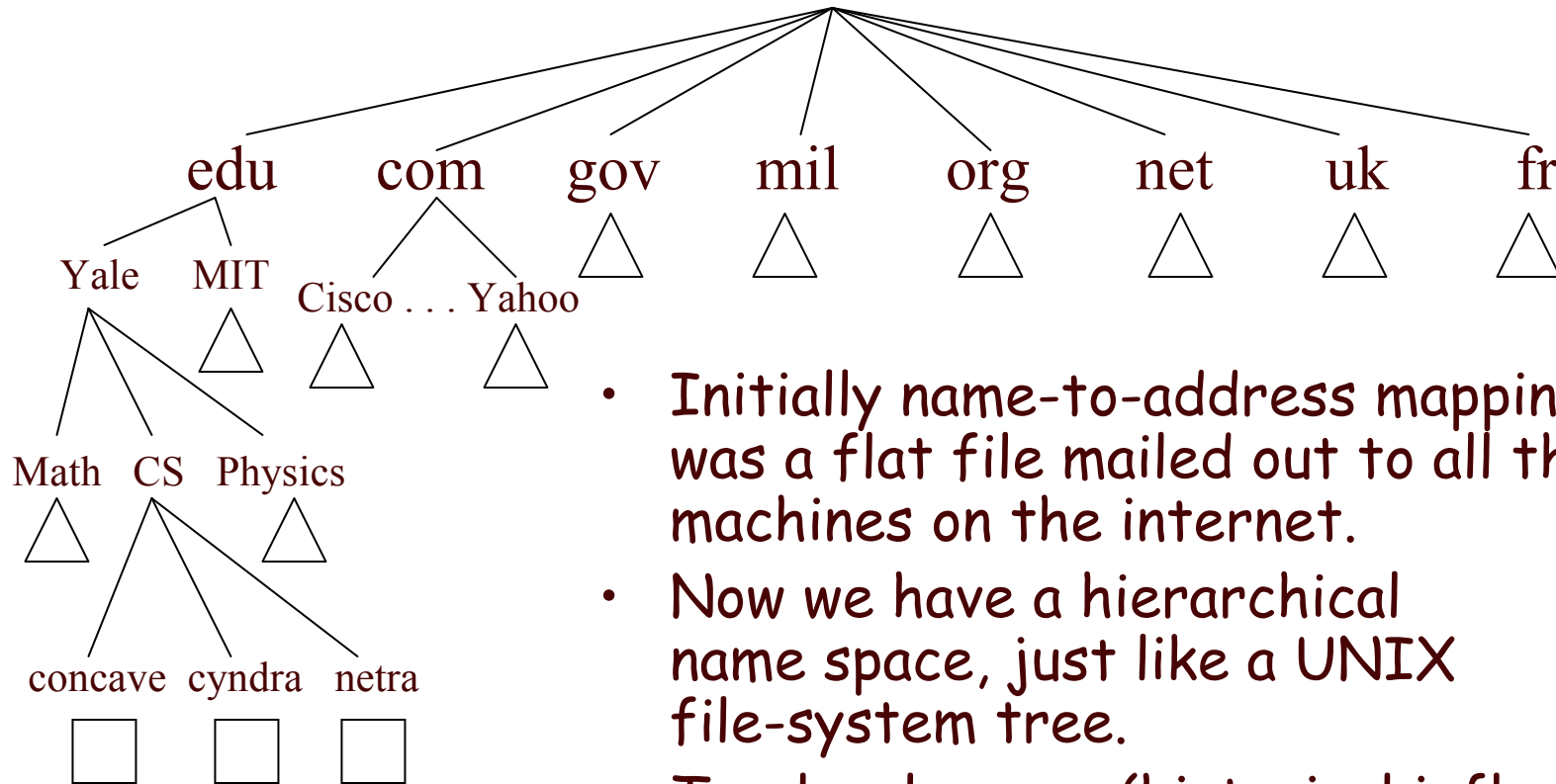


- How do hosts share links?
- How do you name and address hosts?
- Routing: given a destination address, how do you get to it?

IP Addresses and Host Names

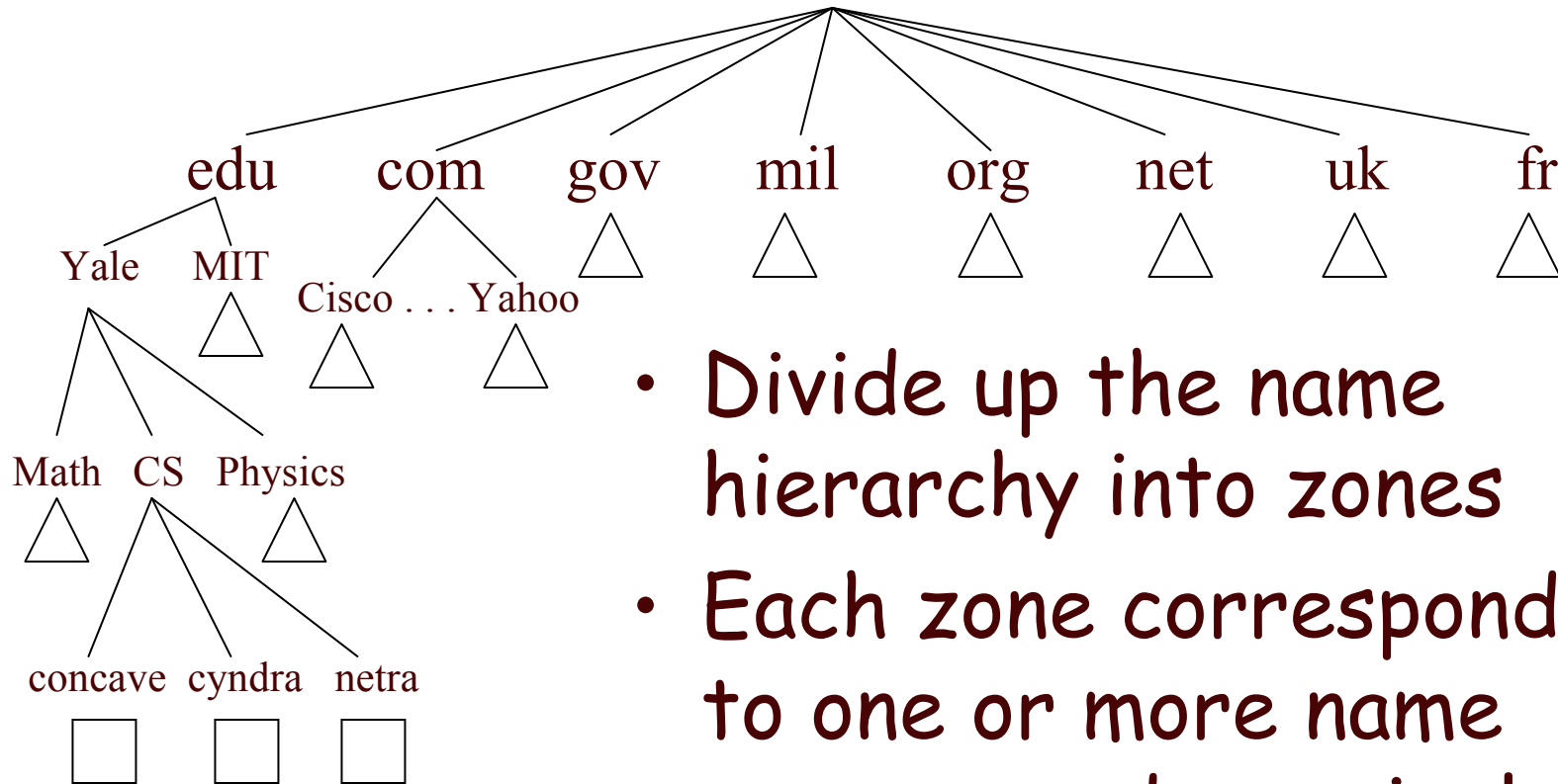
- Each machine is addressed by an integer, its IP address, written down in a "dot notation" for "ease" of reading, such as 128.36.229.231
- IP addresses are the universal IDs that are used to name everything
- For convenience, each host also has a human-friendly host name. For example, 128.36.229.231 is concave.cs.yale.edu.
- Question: how do you translate names into IP addresses?

Domain Hierarchy



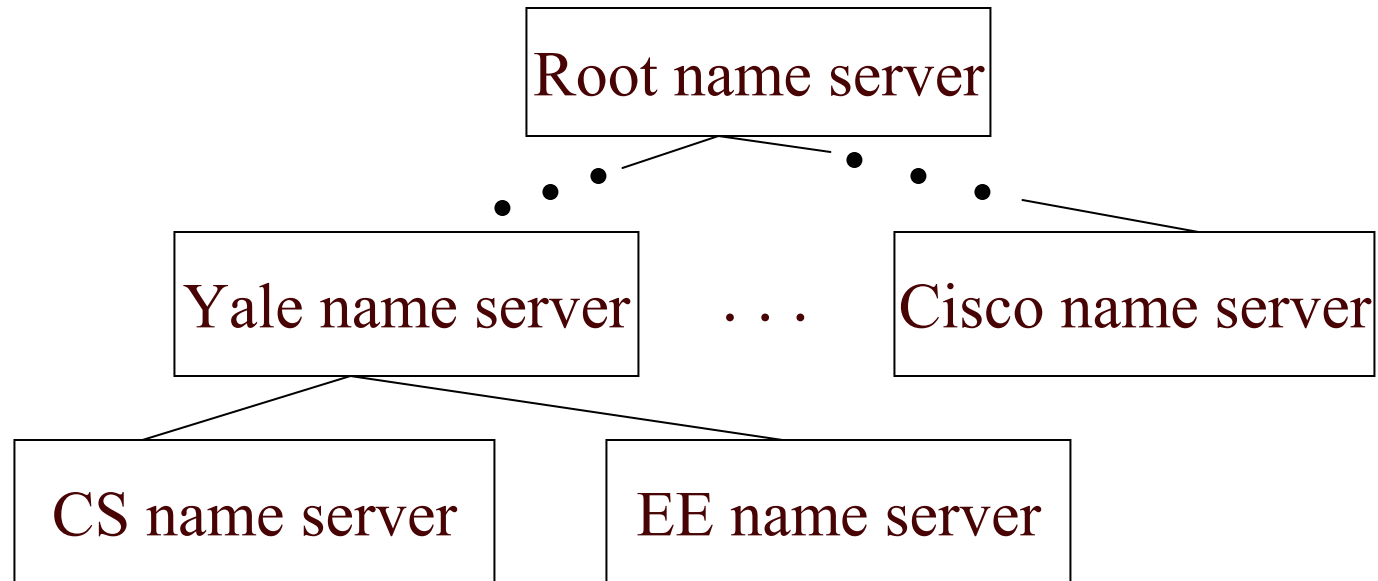
- Initially name-to-address mapping was a flat file mailed out to all the machines on the internet.
- Now we have a hierarchical name space, just like a UNIX file-system tree.
- Top-level names (historical influence): heavily US-centric, government-centric, and military-centric view of the world.

DNS Zones and Name Servers



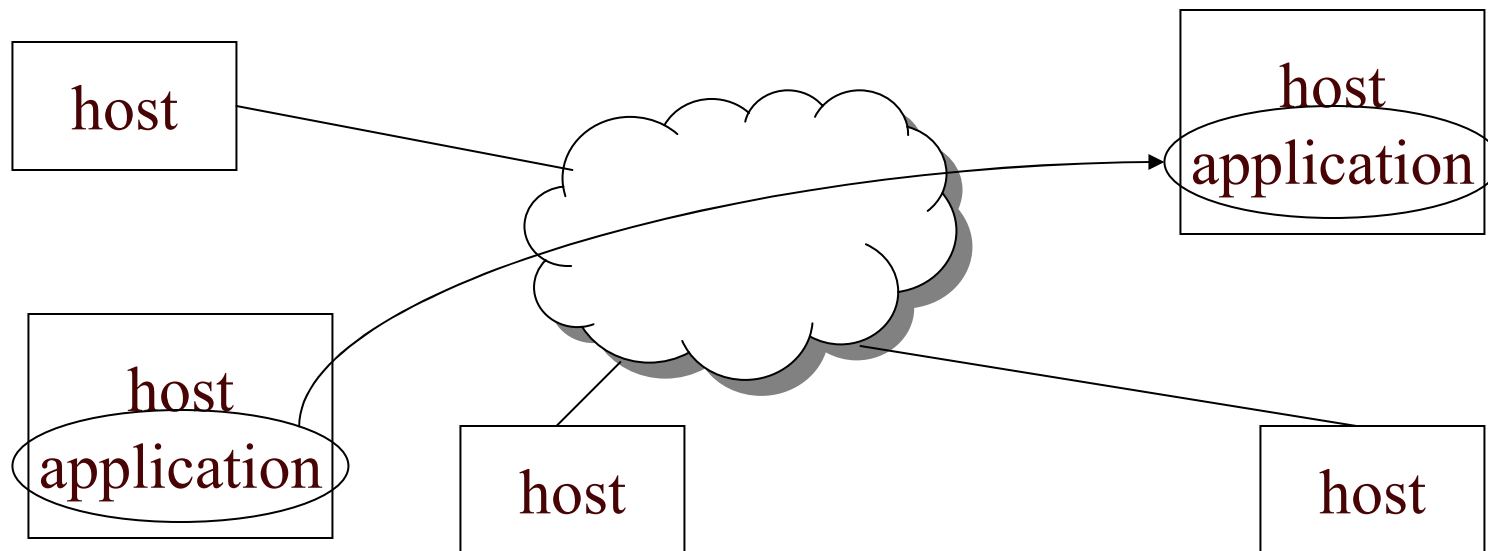
- Divide up the name hierarchy into zones
- Each zone corresponds to one or more name servers under a single administrative control

Hierarchy of Name Servers



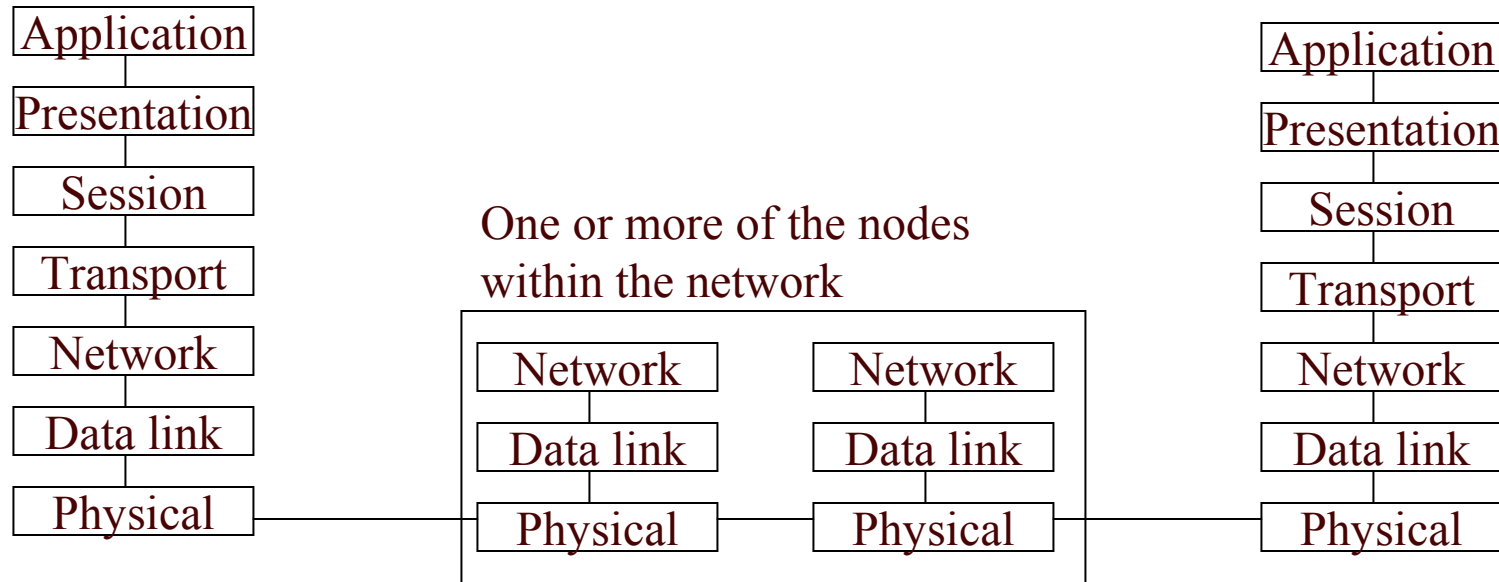
- Clients send queries to name servers
- Name servers reply with answers or forward request to other name servers
- Most name servers also perform lookup caching

Application-Level Abstraction



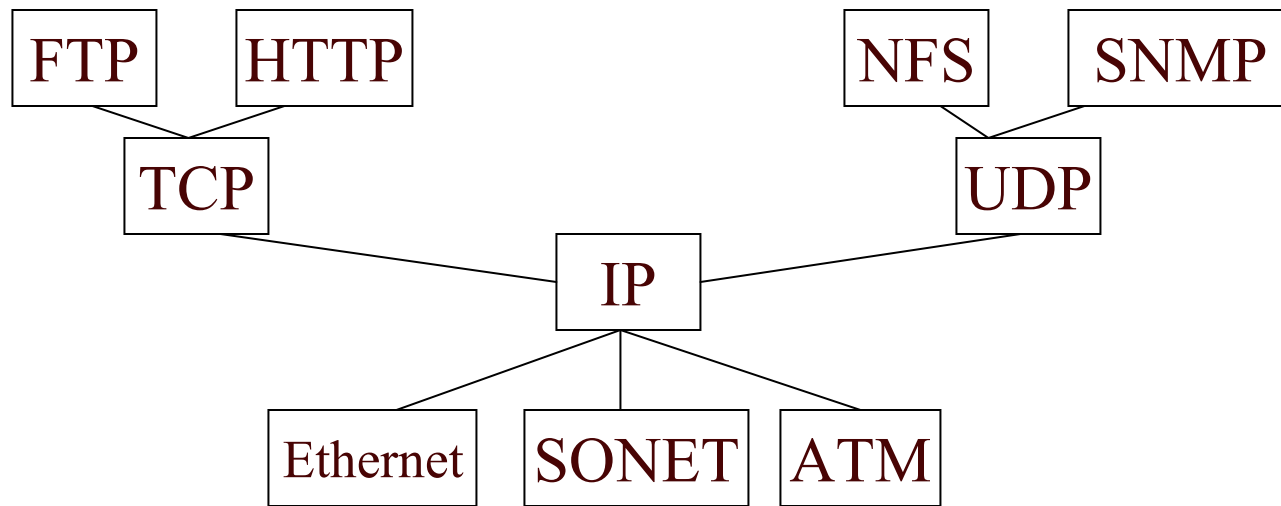
- What you have: hop-to-hop links, multiple routes, packets, can be potentially lost, can be potentially delivered out-of-order
- What you may want: application-to-application (end-to-end) channel, communication stream, reliable, in-order delivery

OSI Architecture



- Physical: handles *bits*
- Data link: provides "*frames*" abstraction
- Network: handles hop-to-hop routing, at the unit of *packets*
- Transport: provides process-to-process semantics such as in-order-delivery and reliability, at the unit of *messages*
- Top three layers are not well-defined, all have to do with application level abstractions such as transformation of different data formats

Reality: the "Internet" Architecture



- Protocols: abstract objects that make up a layer
- Lowest level: hardware specific, implemented by a combination of network adaptors and OS device drivers
- IP (Internet Protocol): focal point of the architecture, provides host-to-host connection, defines common methods of exchanging packets
- TCP (transmission Control Protocol): reliable, in-order stream
- UDP (User Datagram Protocol): unreliable messages (maybe faster)
- On top of those are the application protocols
- Not strictly layered, "hour-glass shape," implementation-centric

Reading Assignment For January 21

- Text, Chapter 1
- "Rethinking the design of the Internet: The end to end arguments vs. the brave new world," Clark and Blumenthal, 2000 (<http://itel.mit.edu/itel/docs/jun00/TPRC-Clark-Blumenthal.pdf>)
- (Optional) "TCP and UDP" (<http://www.networkmagazine.com/article/NMG20010126S0005>)
- (Optional) "How Domain Name Servers Work," HowStuffWorks.com (<http://www.howstuffworks.com/dns.htm/printable>)