Intro to Distributed Systems and Networks

Hank Levy

Distributed Systems

- **Nearly all systems today are distributed in some way, e.g.:**
  - they use email
  - they access files over a network
  - they access printers over a network
  - they are backed up over a network
  - they share other physical or logical resources
  - they cooperate with other people on other machines
  - they receive video, audio, etc.

Why use distributed systems?

- **Distributed systems are now a requirement:**
  - economics dictate that we buy small computers
  - everyone needs to communicate
  - we need to share physical devices (printers) as well as information (files, etc.)
  - many applications are by their nature distributed (bank teller machines, airline reservations, ticket purchasing)
  - in the future, to solve the largest problems, we will need to get large collections of small machines to cooperate together (parallel programming)

What is a distributed system?

- **There are several levels of distribution.**
- **Earliest systems used simple explicit network programs:**
  - FTP: file transfer program
  - Telnet (rlogin): remote login program
  - mail
  - remote job entry (or rsh): run jobs remotely
- **Each system was a completely autonomous independent system, connected to others on the network**

Loosely-Coupled Systems

- Most distributed systems are “loosely-coupled”:
- Each CPU runs an independent autonomous OS.
- Hosts communicate through message passing.
- Computer don’t really trust each other.
- Some resources are shared, but most are not.
- The system may look differently from different hosts.
- Typically, communication times are long.

Closely-Coupled Systems

- **A distributed system becomes more "closely coupled" as it:**
  - appears more uniform in nature
  - runs a "single" operating system
  - has a single security domain
  - shares all logical resources (e.g., files)
  - shares all physical resources (CPUs, memory, disks, printers, etc.)
- **In the limit, a distributed system looks to the user as if it were a centralized timesharing system, except that it's constructed out of a distributed collection of hardware and software components.**

Tightly-Coupled Systems

- **A “tightly-coupled” system usually refers to a multiprocessor.**
  – Runs a single copy of the OS with a single job queue
  – has a single address space
  – usually has a single bus or backplane to which all processors and memories are connected
  – has very low communication latency
  – processors communicate through shared memory

Some Issues in Distributed Systems

- Transparency (how visible is the distribution)
- Security
- Reliability
- Performance
- Scalability
- Programming models
- Communications models

Transparency

- **In a true distributed system with transparency:**
  - it would appear as a single system
  - different modes would be invisible
  - jobs would migrate automatically from node to node
  - a job on one node would be able to use memory on another

Distribution and the OS

- **There are various issues that the OS must deal with:**
  - how to provide efficient network communication
  - what protocols to use
  - what is the application interface to remote apps (although this might be a language issue)
  - protection of distributed resources

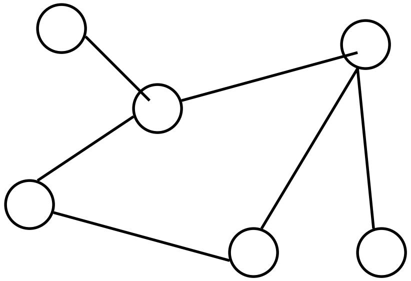
The Network

- There are various network technologies that can be used to interconnect nodes.
- In general, Local Area Networks (LANs) are used to connect hosts within a building. Wide Area Networks (WANs) are used across the country or planet.
- We are at an interesting point, as network technology is about to see an order-of-magnitude performance increase. This will have a huge impact on the kinds of systems we can build.

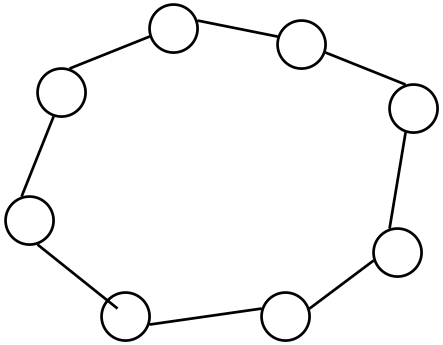
Issues in Networking

- Routing
- Bandwidth and contention
- Latency
- Reliability
- Efficiency
- Cost
- Scalability

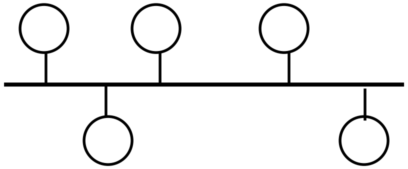
Network Topologies



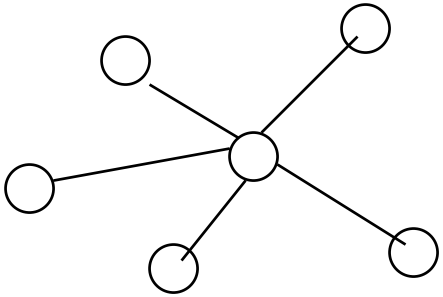
Point to Point



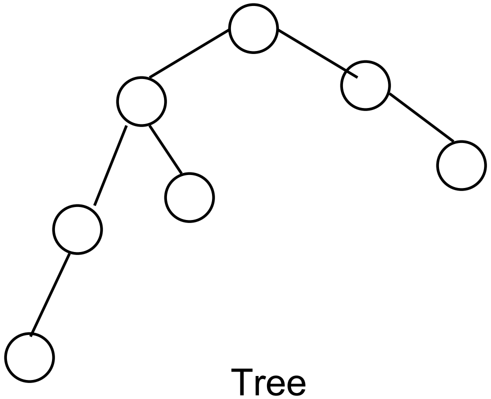
Ring



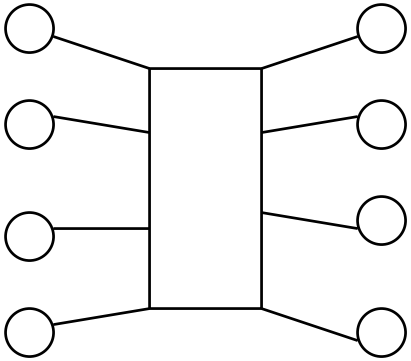
Broadcast



Star



Tree



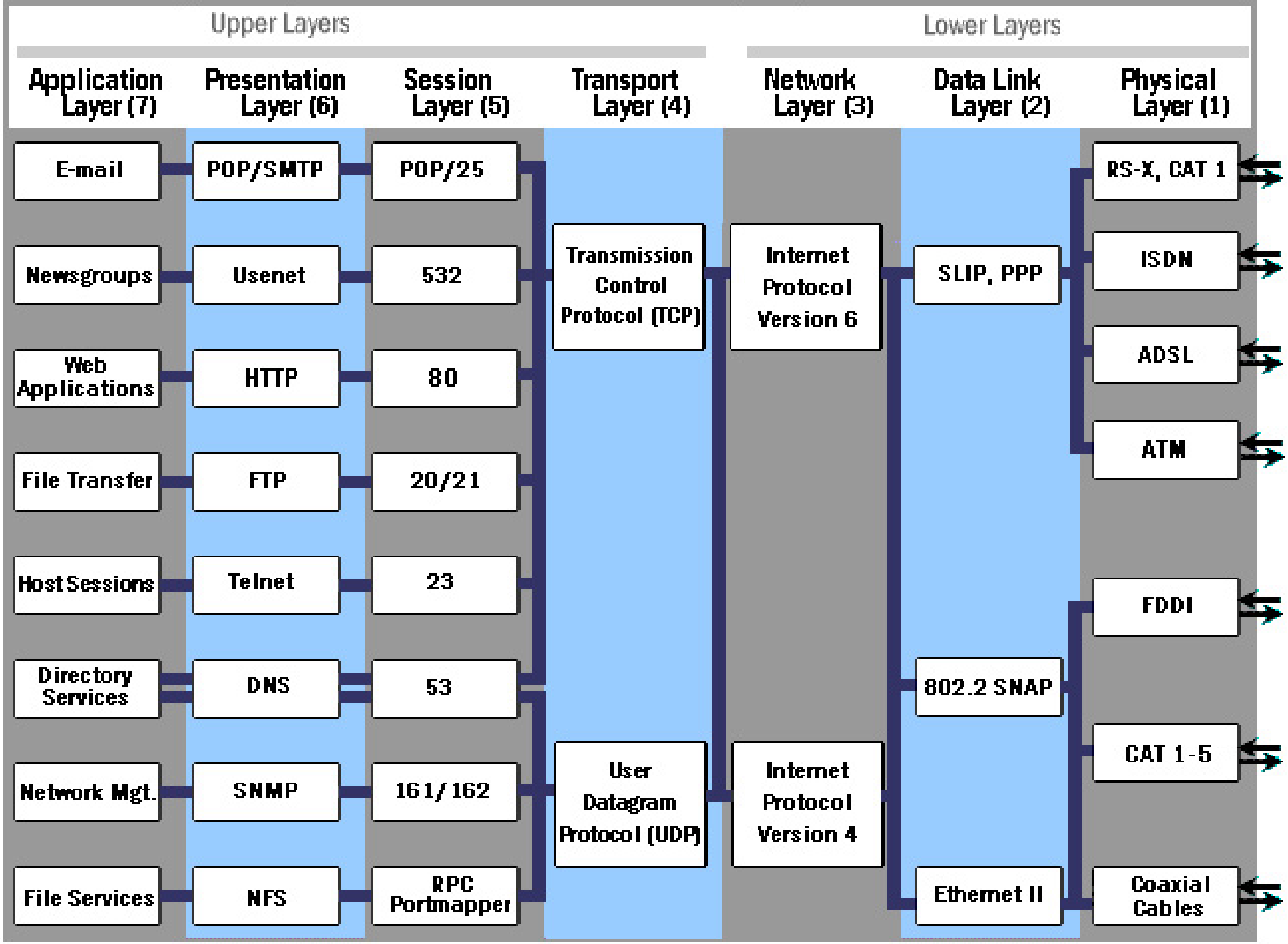
Switch

Traditionally, two ways to handle networking

- **Circuit Switching**
  - what you get when you make a phone call
  - good when you require constant bit rate
  - good for reserving bandwidth (refuse connection if bandwidth not available)
- **Packet Switching**
  - what you get when you send a bunch of letters
  - network bandwidth consumed only when sending
  - packets are routed independently
  - packetizing may reduce delays (using parallelism)

- Phone systems are moving to packet switching because of the Internet and the reduced equipment cost!

Open Systems Interconnection (OSI) Reference Model



Upper Layers

Lower Layers

Application Layer (7)	Presentation Layer (6)	Session Layer (5)
E-mail	POP/SMTP	POP/25
Newsgroups	Usenet	532
Web Applications	HTTP	80
File Transfer	FTP	20/21
Host Sessions	Telnet	23
Directory Services	DNS	53
Network Mgt.	SNMP	161/162
File Services	NFS	RPC Portmapper

Transport Layer (4)

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Network Layer (3)

- Internet Protocol Version 6
- Internet Protocol Version 4

Data Link Layer (2)

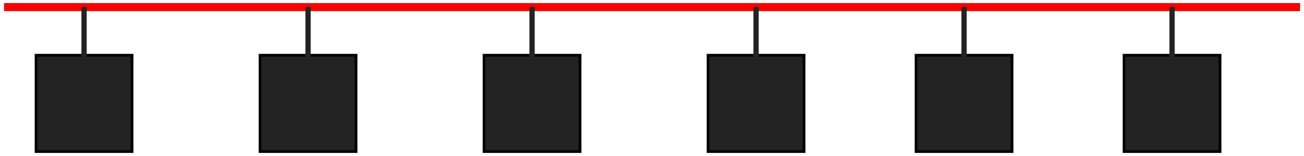
- SLIP, PPP
- 802.2 SNAP
- Ethernet II

Physical Layer (1)

- RS-X, CAT 1
- ISDN
- ADSL
- ATM
- FDDI
- CAT 1-5
- Coaxial Cables

Data link layer: Ethernet

- Broadcast network



- CSMA-CD: Carrier Sense Multiple Access with Collision Detection
  - recall the “standing in a circle, drinking beer and telling stories” analogy
- Packetized – fixed
- Every computer has a unique physical address
  - 00-08-74-C9-C8-7E

Data Link Message

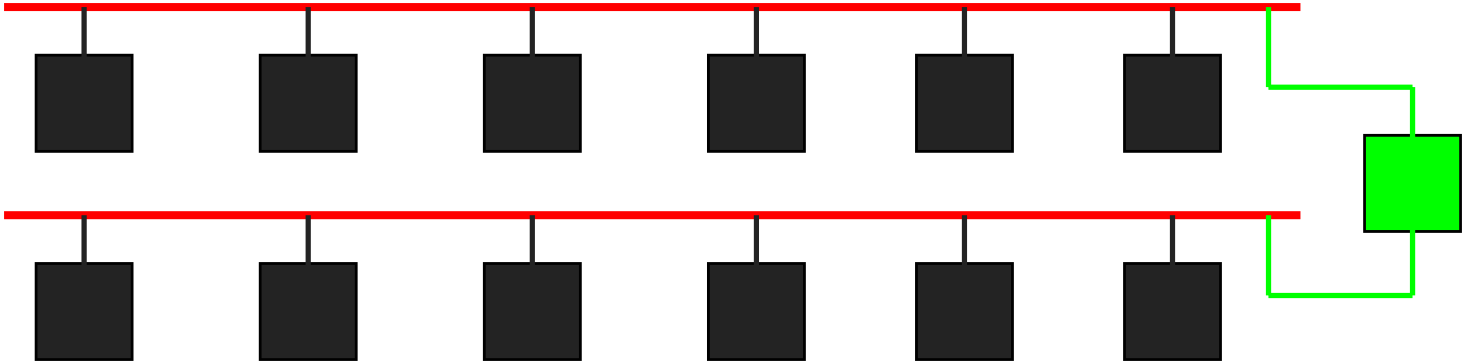
- Packet format

physical address	payload

- Interface listens for its address, interrupts OS when a packet is received

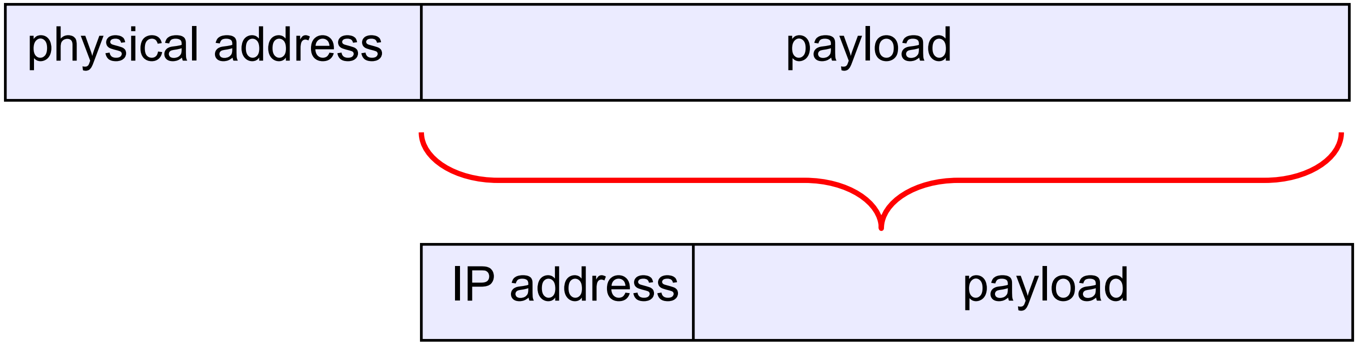
Network layer: IP

- Internet Protocol (IP)
  - routes packets across multiple networks, from source to destination
- Every computer has a unique Internet address
  - 128.208.3.200
- Individual networks are connected by routers that have physical addresses (and interfaces) on each network



IP Level Message

- A really hairy protocol lets any node on a network find the physical address on that network of a router that can get a packet one step closer to its destination
- Packet format

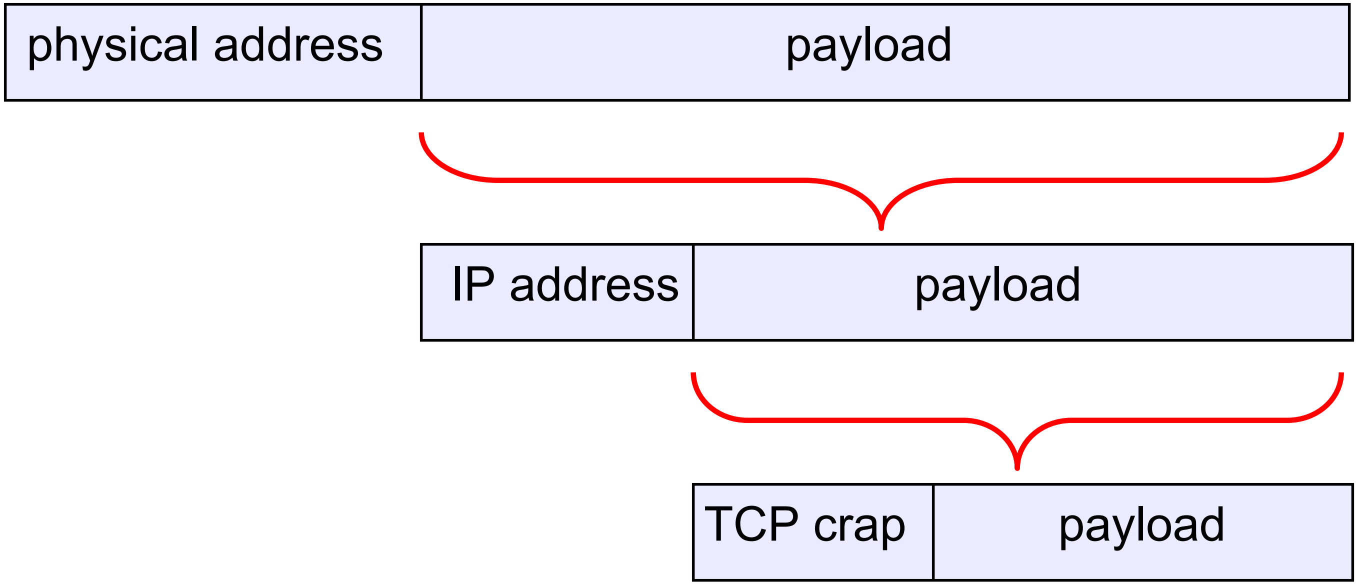


DNS

- A separate really hairy protocol, DNS (the Domain Name Service), maps from intelligible names (cs.washington.edu) to IP addresses (128.208.3.200)
- So to send a packet to a destination
  - use DNS to convert domain name to IP address
  - prepare IP packet, with payload prefixed by IP address
  - determine physical address of appropriate router
  - encapsulate IP packet in Ethernet packet with appropriate physical address
  - blast away!
- Detail: port number gets you to a specific address space on a system

Transport layer: TCP

- TCP: Transmission Control Protocol
  - manages to fabricate reliable multi-packet messages out of unreliable single-packet datagrams
  - analogy: sending a book via postcards – **what's required?**



TCP/IP summary

- Using TCP/IP and lower layers, we can get multi-packet messages delivered reliably from address space A on machine B to address space C on machine D, where machines B and D are many heterogeneous network hops apart, without knowing any of the underlying details
- Higher protocol layers facilitate specific services
  - email: smtp
  - web: http
  - file transfer: ftp
  - remote login: telnet

New applications will define the Internet

- **VOIP (voice over IP)**
- **Streaming real-time video**
- **Multi-player games**
- **Other stuff that you'll invent…**