inst.eecs.berkeley.edu/~cs61c

CS61C : Machine Structures

Lecture 39
I/O : Networks

2004-12-01

Lecturer PSOE Dan Garcia

www.cs.berkeley.edu/~ddgarcia

Clean to Zombie Bot in 4min ⇒

USA Today and Avantegarde report that it took less than 4 min for an unprotected PC running XP SP1 to be compromised. The Mac and Linux box were attacked but didn't fall.

www.usatoday.com/money/industries/technology/2004-11-29-honeypot_x.htm

Buses in a PC: connect a few devices (2002)

CPU — Memory bus — Memory

PCI Interface

PCI: Internal (Backplane) I/O bus

Ethernet Interface, SCSI Interface

SCSI: External I/O bus

(1 to 15 disks)

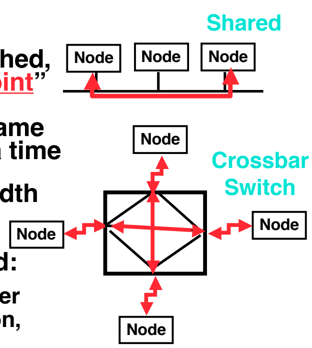
Ethernet Local Area Network

Bus - shared medium of communication that can connect to many devices. Hierarchy!!

- Data rates (P4)
  - Memory: 400 MHz, 8 bytes ⇒ 3.2 GB/s (peak)
  - PCI: 100 MHz, 8 bytes wide ⇒ 0.8 GB/s (peak)
  - SCSI: "Ultra4" (160 MHz), "Wide" (2 bytes) ⇒ 0.3 GB/s (peak)
  - Gigabit Ethernet: ⇒ 0.125 GB/s (peak)

Shared vs. Switched Based Networks

- Shared Media vs. Switched: in switched, pairs ("point-to-point" connections) communicate at same time; shared 1 at a time
- Aggregate bandwidth (BW) in switched network is many times shared:
  - point-to-point faster since no arbitration, simpler interface



Why Networks?

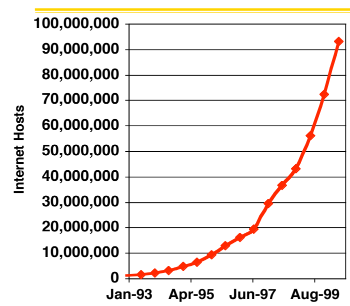
- Originally sharing I/O devices between computers (e.g., printers)
- Then Communicating between computers (e.g, file transfer protocol)
- Then Communicating between people (e.g., email)
- Then Communicating between networks of computers ⇒ File sharing, WWW, …

How Big is the Network (1999)?

~30	Computers in 273 Soda
~400	in inst.cs.berkeley.edu
~4,000	in eecs&cs .berkeley.edu
~50,000	in berkeley.edu
~5,000,000	in .edu
~46,000,000	in US (.com .net .edu .mil .us .org)
~56,000,000	in the world

Source: Internet Software Consortium

Growth Rate



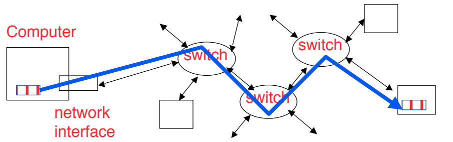
"Source: Internet Software Consortium (http://www.isc.org/)".

Ethernet Bandwidth

1983	3 mb/s
1990	10 mb/s
1997	100 mb/s
1999	1000 mb/s
2004	10 Gig E (to come!)

What makes networks work?

- **links** connecting **switches** to each other and to computers or devices



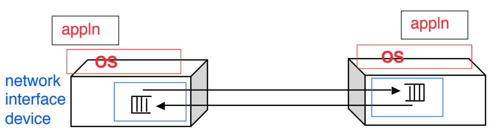
- **ability to name the components and to route packets of information - messages - from a source to a destination**
- **Layering, protocols, and encapsulation as means of abstraction (61C big idea)**

Typical Types of Networks

- **Local Area Network (Ethernet)**
  - Inside a building: Up to 1 km
  - (peak) Data Rate: 10 Mbits/sec, 100 Mbits /sec,1000 Mbits/sec (1.25, 12.5, 125 MBytes/s)
  - Run, installed by network administrators
- **Wide Area Network**
  - Across a continent (10km to 10000 km)
  - (peak) Data Rate: 1.5 Mb/s to 10000 Mb/s
  - Run, installed by telecommunications companies (Sprint, UUNet[MCI], AT&T)
- **Wireless Networks (LAN), ...**

ABCs of Networks: 2 Computers

- **Starting Point**: Send bits between 2 computers



- Queue (First In First Out) on each end
- Can send both ways ("Full Duplex")
- Information sent called a "message"
  - Note: Messages also called packets

A Simple Example: 2 Computers

- **What is Message Format?**
  - Similar idea to Instruction Format
  - Fixed size? Number bits?

Length	Data
8 bit	32 x Length bits

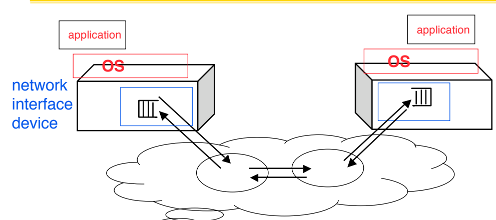
- **Header(Trailer)**: information to deliver message
- **Payload**: data in message
- What can be in the data?
  - anything that you can represent as bits
  - values, chars, commands, addresses...

Questions About Simple Example

- **What if more than 2 computers want to communicate?**
  - Need computer "address field" in packet to know which computer should receive it (destination), and to which computer it came from for reply (source) [just like envelopes!]

Dest. Net ID	Source Net ID	Len	CMD/ Address /Data
8 bits	8 bits	8 bits	32xn bits
Header			Payload

ABCs: many computers



- switches and routers interpret the header in order to deliver the packet
- source encodes and destination decodes content of the payload

Questions About Simple Example

- What if message is garbled in transit?
- Add redundant information that is checked when message arrives to be sure it is OK
- 8-bit sum of other bytes: called "Check sum"; upon arrival compare check sum to sum of rest of information in message. `xor` also popular.

Math 55 talks about what a Check sum is…

Questions About Simple Example

- What if message never arrives?
- Receiver tells sender when it arrives (ack) [ala registered mail], sender retries if waits too long
- Don't discard message until get "ACK" (for ACKnowledgment);
  Also, if check sum fails, don't send ACK

Observations About Simple Example

- Simple questions such as those above lead to more complex procedures to send/receive message and more complex message formats
- Protocol: algorithm for properly sending and receiving messages (packets)

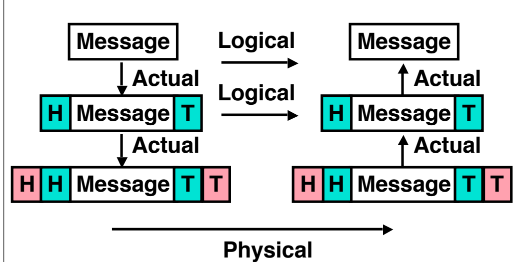
Software Protocol to Send and Receive

- SW Send steps
  - 1: Application copies data to OS buffer
  - 2: OS calculates checksum, starts timer
  - 3: OS sends data to network interface HW and says start
- SW Receive steps
  - 3: OS copies data from network interface HW to OS buffer
  - 2: OS calculates checksum, if OK, send ACK; if not, delete message (sender resends when timer expires)
  - 1: If OK, OS copies data to user address space, & signals application to continue

Protocol for Networks of Networks?

- Internetworking: allows computers on independent and incompatible networks to communicate reliably and efficiently;
  - Enabling technologies: SW standards that allow reliable communications without reliable networks
  - Hierarchy of SW layers, giving each layer responsibility for portion of overall communications task, called protocol families or protocol suites
- Abstraction to cope with complexity of communication vs. Abstraction for complexity of computation

Protocol Family Concept



Protocol Family Concept

- Key to protocol families is that communication occurs logically at the same level of the protocol, called peer-to-peer…

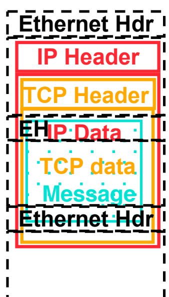
  …but is implemented via services at the next lower level
- Encapsulation: carry higher level information within lower level “envelope”
- Fragmentation: break packet into multiple smaller packets and reassemble

Protocol for Network of Networks

- Transmission Control Protocol/Internet Protocol (TCP/IP)
  - This protocol family is the basis of the Internet, a WAN protocol
  - IP makes best effort to deliver
  - TCP guarantees delivery
  - TCP/IP so popular it is used even when communicating locally: even across homogeneous LAN

TCP/IP packet, Ethernet packet, protocols

- Application sends message
- TCP breaks into 64KB segments, adds 20B header
- IP adds 20B header, sends to network
- If Ethernet, broken into 1500B packets with headers, trailers (24B)
- All Headers, trailers have length field, destination, ...



Overhead vs. Bandwidth

- Networks are typically advertised using peak bandwidth of network link: e.g., 100 Mbits/sec Ethernet (“100 base T”)
- Software overhead to put message into network or get message out of network often limits useful bandwidth
- Assume overhead to send and receive = 320 microseconds (μs), want to send 1000 Bytes over “100 Mbit/s” Ethernet
  - Network transmission time: 1000Bx8b/B /100Mb/s = 8000b / (100b/μs) = 80 μs
  - Effective bandwidth: 8000b/(320+80)μs = 20 Mb/s

Peer Instruction

(T / F) P2P filesharing has been the dominant application on many links!

Suppose we have 2 networks, Which has a higher effective bandwidth as a function of the transferred data size?

- BearsNet
  TCP/IP overhead 300 μs, peak BW 10Mb/s
- CalNet
  TCP/IP overhead 500 μs, peak BW 100Mb/s

```
   TRUE
1: B always
2: C always
3: B small
   C big
4: B big
   C small
5: The same!

   FALSE
6: B always
7: C always
8: B small
   C big
9: B big
   C small
0: The same!
```

And in conclusion…

- Protocol suites allow heterogeneous networking
  - Another form of principle of abstraction
  - Protocols ⇒ operation in presence of failures
  - Standardization key for LAN, WAN
- Integrated circuit (“Moore’s Law”) revolutionizing network switches as well as processors
  - Switch just a specialized computer
- Trend from shared to switched networks to get faster links and scalable bandwidth