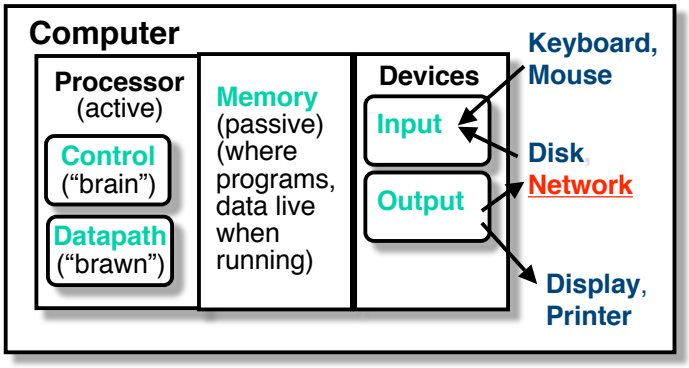
CS61C – Machine Structures
Lecture 37 – Networks

April 24, 2006
John Wawrzynek

No Machine is an Island!



- Like Disk – very important high-bandwidth I/O device.
- Like Disk – a few special uses by OS:
  - File system sometimes extended to other machines, remote printing, …
  - and many uses by user programs (applications)

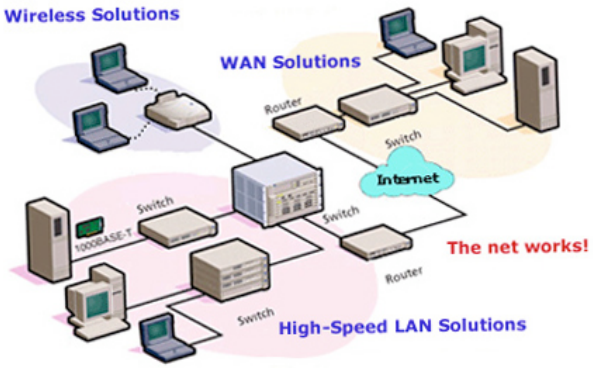
Dominate Network Applications

Email	IMAP	client/server
World Wide Web	firefox/apache	client/server
Media file sharing	*Kazaa*	peer to peer
Online Gaming	*Counter-strike*	peer to peer
Remote "terminal"	ssh	client/server
Media distrib./streaming	*iTunes*	client/server
Software distribution	MS update	client/server
Cluster computing	Beowulf/Now	peer to peer

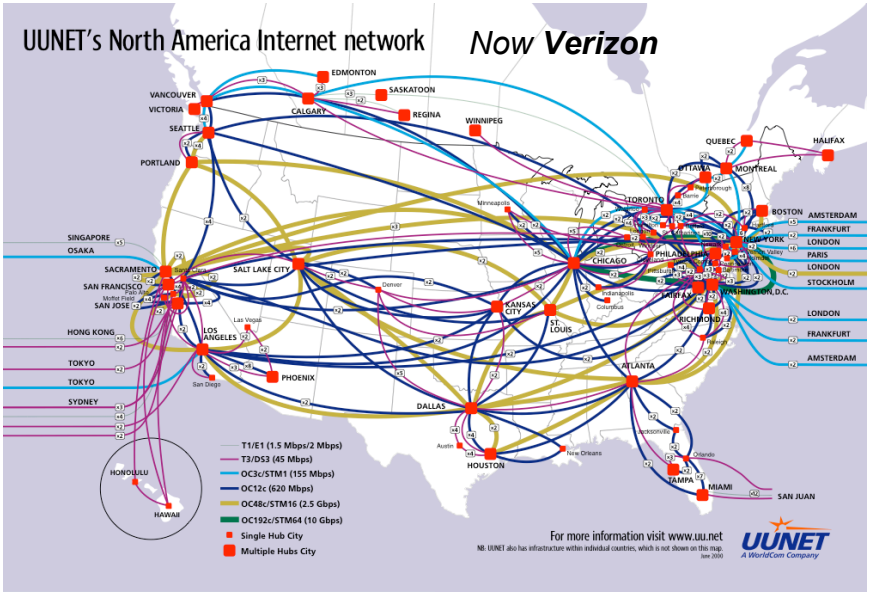
- This set of applications is the dominate use for machines today. *Maybe we should call them* communicators *instead of* computers*!*
- Underneath, in everyone of these applications, one machine communicates directly with another.

Internet

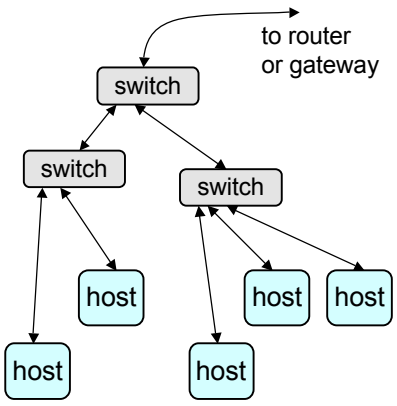
- The Internet is a network of networks of networks …
- Each level can differ in its bandwidth, communication protocols, and physical media (twisted wires, coaxial cables, fiber optics, wireless radios)



Internet

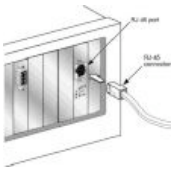


Local Area Network (LAN) Basics

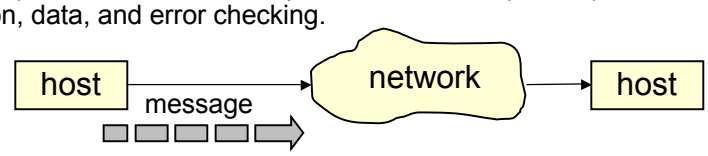


- A LAN is made up physically of a set of switches, wires, and hosts. Routers and gateways provide connectivity out to other LANs and to the internet.
- Ethernet defines a set of standards for data-rate (10/100/1000 Mbps), and signaling to allow switches and computers to communicate (IEEE 802.3)
- Most Ethernet implementations these days are "switched" (point to point connections between switches and hosts, no contention or collisions).

Ethernet



- An Ethernet interface card or Network Interface Card (NIC) is used to bring the network into the host:
- Information travels in variable sized blocks, called Ethernet Frames (or packets), each frame includes preamble, header (control) information, data, and error checking.

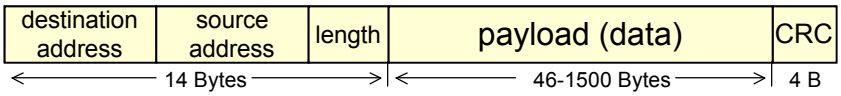


- Link level protocol on Ethernet is called the Medium Access Control (MAC) protocol. It defines the format of the packets.
- Frame format:

Preamble (8 bytes)	MAC header	Payload	CRC

  - Preamble is a fixed pattern used by receivers to synchronize their clocks to the data.
  - Payload is the actual information the host is sending.

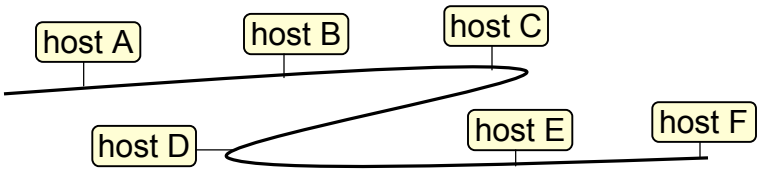
Ethernet (802.3) Frame Format



- MAC protocol *encapsulates* a payload by adding a 14 byte header before the data and a 4-byte cyclic redundancy check (CRC) after the data.
- Each network hardware device is assigned a unique address (called MAC address), assigned globally.
- A 6-Byte **destination address**, specifies either a single recipient node (unicast mode), a group of recipient nodes (multicast mode), or the set of all recipient nodes (broadcast mode).
- A 6-Byte **source address**, is set to the sender's globally unique node address. Its main function is to allow address learning which may be used to configure the filter tables in switches.
- A 2-byte **length** field, indicates the number of bytes in the payload field.
- The 4-Byte **CRC** provides error detection in the case where line errors result in corruption of the MAC frame. Any frame with an invalid CRC should discarded by the MAC receiver without further processing.

Ethernet Control – old style CSMA/CD

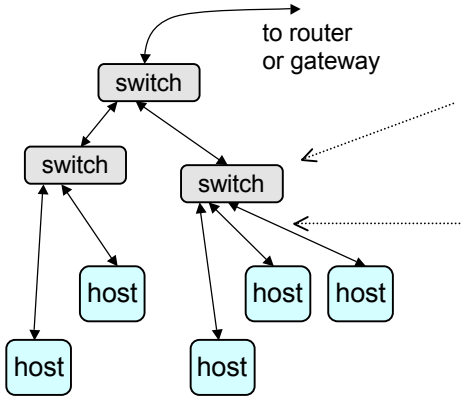
- To keep cost down, inventers of Ethernet wanted no switches – just hosts and Ethernet interfaces.
- They used a protocol called Carrier Sense Multiple Access/Collision Detect (CSMA/CD):



- A host wanting to transmit senses whether the line is idle and therefore available to be used. If it is, the host begins to transmit its frame and listens as it does. If another device has tried to send at the same time, a *collision* occurs and the frames are discarded.
- Each device then waits a random amount of time and retries. If another collision occurs it waits longer before trying again (*exponential backoff*).

Switched Ethernet

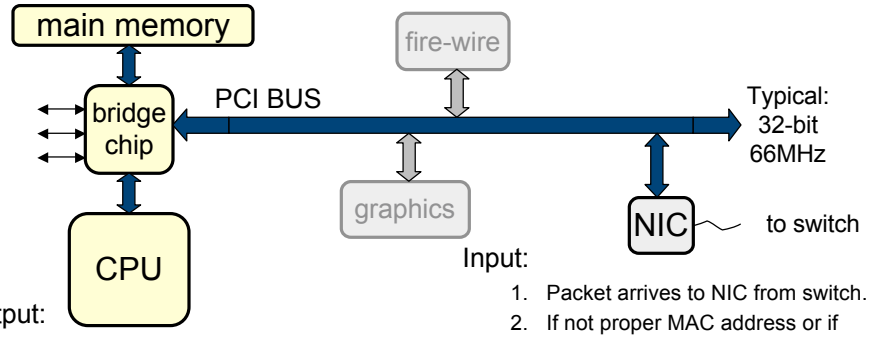
- Modern style Ethernet uses *buffering* and *flow-control* to handle collisions in the network.



Buffers in switch allow one packet to wait while the other uses the output.

Receivers (switches and hosts) use special “pause” and “resume” frames to control sender.

NIC connection into Machine



Output:

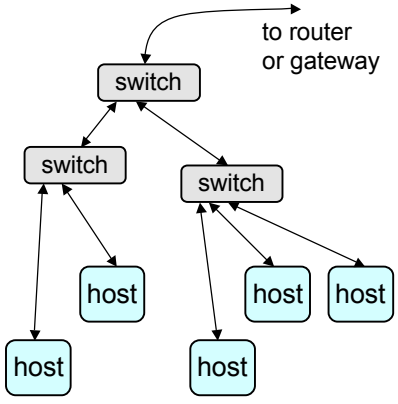
1. CPU prepares payload and leaves in memory buffer.
2. If NIC buffer space available, then initiate DMA transfer to NIC.
3. If no space wait for interrupt.
4. NIC transfers frame out to switch, then interrupts CPU with "buffer space" update.

Input:

1. Packet arrives to NIC from switch.
2. If not proper MAC address or if CRC error ignore
3. Else save payload in buffer and interrupt CPU
4. CPU initiates DMA transfer to memory buffer.
5. DMA finishes and notifies CPU
6. CPU processes packet according to high-level protocol and application.

So far …

- Ethernet (IEEE 802.3):
  - Good for routing within local area network (LAN).
  - **Difficult for truly global routing**, every switch everywhere would need to store all MAC addresses – (we really need some kind of address hierarchy).
  - **Unreliable**:
    - No automatic retransmission on error.
    - No acknowledgements – sender doesn't know if receiver got the data.



TCP/IP

A suite of protocols for global host addressing and reliable transmission on the internet.

- TCP/IP is an example of a layered protocol: each layer builds upon the layer below it, adding new functionality.
- The protocol stack is the collection of protocol that make up the suite:

protocol for transferring files / delivering mail	P2
protocol for routing and reliability	P1
protocol for sending and receiving data using specific hardware	P0

- Each protocol layer *encapsulates* the layer above it:

packet format:

P0 header	P1 header	P2 header	data

- Stacks are modular, so they can easily change when a new hardware model is adapted or needs of applications change. (Replace one module).

TCP/IP

- TCP/IP is a 4-layer protocol:

Application layer:	FTP, SMTP, HTTP
Transport layer:	TCP, UDP
Network layer:	IP
Link Layer:	IEEE 802.x, PPP, SLIP

- Link level examples:
  - IEEE 802.3 for Ethernet, 802.5 for token-ring, 802.11 for wireless,
  - Used with dial-up modems: Serial line IP (SLIP), Point-to-Point protocol (PPP).

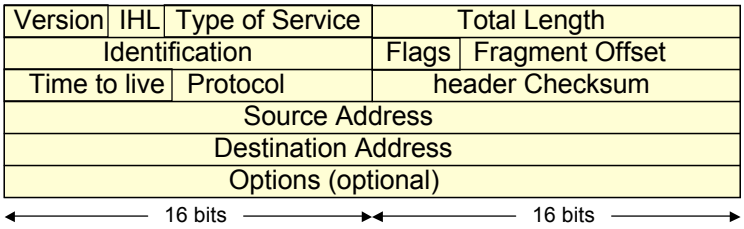
IP (Internet Protocol)

Extends the idea of host address from MAC to a hierarchical "soft" address. All hosts take on an IP address.

- *The job of IP is to enable data to be transmitted between networks* (adds very little in the context of a LAN over what is possible with MAC addresses).
- Features of IP:
  - *Connectionless* – no concept of a job or session. Every packet treated individually.
  - *In-order delivery not ensured*.
  - *Unreliable* protocol.

The link layer (Ethernet) needs to know the unique address (MAC) of the specific place to next deliver the message. TCP/IP suite include ARP (address resolution protocol) to map from IP address to MAC address. Protocol works by broadcasting a request on the network – if a host sees its IP address, it replies with its MAC. If the IP is outside this subnet, then the router (connecting out) will reply).

IP Packets



- *Protocol* field: says which high-level protocol sent the data – used by destination to pass packet to right protocol module.
- TTL (time to live): Initialized by the sender (usually 64) then decr. by 1 by every router the packet passes through. When reaches 0, the packet is discarded and the sender is notified with the Internet control message protocol (ICMP). This keeps packets from getting stuck in loops. (Also, used by traceroute).

- *Internet Addressing*: every host directly connected to the internet has a unique address (issued by IANA, iana.org).
- Internet addresses are 32-bits long written as 4-Bytes separated by periods. Range:

1.0.0.1 to 223.255.255.255

IP Routing

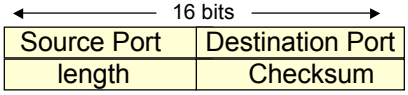
- Local routing is done according to the specifics of the LANs own protocol.
- Routing to outside networks is done through *routers (these are either hosts with multiple NICs and special routing software, or special router hardware*.)
  - Each host on the LAN is assigned a default router, used to connect it to outside.

- A router examines every packet and compares the destination address with a table of address.
  1. If it finds an exact match, it forwards the packet to the address associated with that entry in the table.
  2. If the router doesn't find a match, it runs through to the table looking for a match just on the network ID. If a match is found, the packet is sent on to the address associated with that entry.
  3. If no match, the router sends it to the default, next-hop router, if present.
  4. If no default router present, the router sends an ICMP "host unreachable" messsage back to the sender.

- Routers build up their tables in multiple ways:
  - Static – read from a file on startup.
  - Dynamically, by broadcasting ICMP router solicitation messages to which other routers respond.
  - Other protocols are used to discover the shortest path to a location.
  - Routers are updated periodically in response to traffic conditions and availability of a route.

Transport Layer

Two most popular transport protocols are TCP and UDP.



UDP Header

- UDP – User Datagram Protocol
  - Port numbers represent a software port.
  - They identify which protocol module sent (or is to receive) the data.
  - Standard port numbers exist:
    Telnet: port 23, Simple Mail Transfer Protocol: port 25
  - UDP and TCP use the port numbers to determine which application layer protocol should receive the data.
  - UDP isn't reliable, but appropriate for many applications like real-time audio and video (where if data is lost it is better to do without it than to send it again.) Also, gets used for online games.

TCP – Transmission Control Protocol

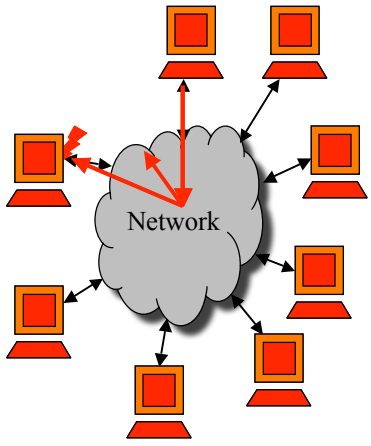
- Transport layer protocol used by most internet applications: FTP, HTTP, Telnet, …

- Connection-oriented: 2 hosts, one a client, and the other a server must establish a connection before any data can be transferred between them (SYN/ACK handshake). Once done the connection must be closed (FIN flag).
- TCP sends data using IP in blocks called segments.
- TCP includes mechanisms for ensuring data which arrives out of sequence is put back into the order it was sent.
- TCP implements flow-control, so a sender app. Cannot overwhelm a receiver app with data.
- TCP provides reliability: When data is received correctly, TCP sends an acknowledgement back to the sender. If the sender doesn't receive an ack within a certain period, the data is recent. For efficiency, the sender will usually send multiple segments without waiting for acks. It keeps track of what segments have or have not been acked – keeping a copy of those that have not, in case they need to be resent.
- ACKs are piggy-backed on data segments for efficiency.

What Are Computer Worms?

- Self replicating network programs
  - Exploit vulnerabilities to infect remote machines
  - Victim machines continue to propagate the infection
- Three main stages
  1. Detect new targets
  2. Attempt to infect new targets
  3. Transfer the worm and activate the code on the victim machine
- Often fully autonomous
  - Spread without any user-interaction required
  - Can be very fast: Slammer infected all vulnerable hosts in 10 minutes

Worm versus Virus: Worm's self propagate through the network, no human interaction or exchange of files required.



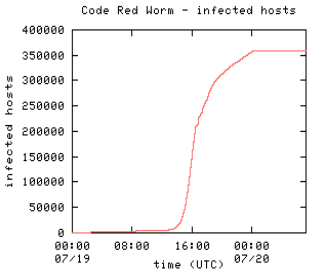
How Do Worms Find Targets?

Most common technique called: Random Scanning

- Repeat Forever:
  - Pick a "random" IP address, if vulnerable, infect it
    - From single host, launch many threads to try more machine addresses simultaneously
  - Other techniques exist
- Fraction of the net infected (a):
  - Function of time and worm's speed
  - "Logistic" function
    - Initial growth is exponential
- Speed (K) depends on:
  - Rate of scanning
  - Number of vulnerable machines
  - Size of address space

$$a = \frac{e^{K(t-T)}}{1 + e^{K(t-T)}}$$

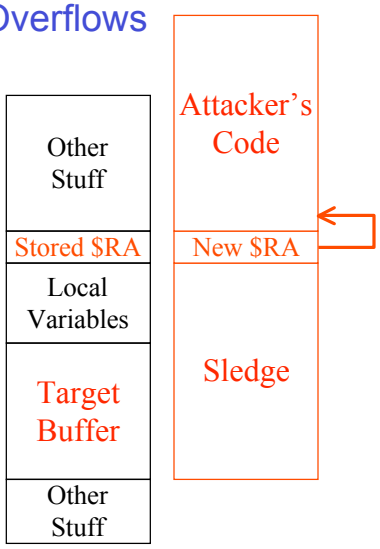
$$K = \frac{\text{Scan Rate * Vuln Machines}}{\text{Address Space Size}}$$



How Do Worms Infect Targets?

Most common technique: Buffer Overflows

- The worm needs to somehow attack the victim machine
  - Take control of the victim
  - Transfer over the body of the worm
- Common Vulnerability: Stack Overflow
  - Victim program has an unchecked buffer on the stack
  - Attacking string overwrites the stack
    - Sledge → dummy data for overwriting
    - Overwritten return address → points to code
    - Injected code → Attacking code to execute
  - Now function returns to the attacker's code
    - The worm now uses this to transfer over the rest of the worm and to start running on the victim



Worm Conclusions

- Example vulnerable applications (these have been, at least partially, patched):
  - Apache and IIS web servers. Code Red attacked IIS.
  - Blaster and its variants attacked Windows RPC (Remote Procedure Call) service, a "default-on" part of the OS.
- To date, most worms have been relatively benign. Most damage comes from flooding the network with scan messages and panic of system administrators. The day will come when a worm will carry a harmful payload:

  Delete files, Re-flash EPROM, (worse if host controls external devices!)
- What can you do?

  As a user: Patch your machine often. Do it today! (You're not just protecting yourself, but the entire network).

  (Many worm writers don't expose vulnerabilities themselves, but wait for MS to announce a patch, then hope that you will not get around to patching your machine.)

  Live behind a "firewall" – blocks traffic on most ports. Some people find this too limiting.

  As a programmer: learn to write secure software.