CIS520 – Operating Systems
Handout 17
Networking

- Networking deals with interconnected groups of machines talking with each other. Is a very different field than operating systems. Have a lot of standards stuff because everyone must agree on what to do when connect machines together.
- What is a network? A collection of machines, links and switches set up so that machines can communicate with each other. Some examples:
  - Telephone system. Machines are telephones, links are the telephone lines and switches are the phone switches.
  - Ethernet. Machines are computers, there is one link (the ethernet) and no switches.
  - Internet. Machines are computers, there are multiple links, both long-haul and local-area links. The switches are gateways.

  Message may have to traverse multiple links and multiple switches to go from source to destination.
- Circuit-switched versus Packet-switched networks. Basic disadvantage of circuit-switched networks - cannot use resources flexibly. Basic advantage of circuit-switched networks - deliver a guaranteed resource.
- Basic Networking Concepts:
  - Packetization.
  - Addressing.
  - Routing.
  - Buffering.
  - Congestion.
  - Flow control.
  - Unreliable Delivery.
  - Fragmentation.
- Local Area Networks. Connect machines in a fairly close geographic area. Standard for many years: Ethernet. Standardized by Xerox, Intel and DEC in 1978. Still in wide use.
- Physical hardware technology: coax cable about 1/2 inch thick. Maximum length: 500 meters. Can extend with repeaters. Can only have two repeaters between any two machines, so maximum length is 1500 meters.
- Vampire taps to connect machines to Ethernet. Attach an ethernet transceiver to tap; the transceiver does the connection between the Ethernet and the host interface. The host interface then connects to the host machine.
- Ethernet is 10 Mbps bus with distributed access control. It is a broadcast medium - all transceivers see all packets and pass all packets to host interface. The host interface chooses packets the host should receive and discards others.
- Access scheme: Carrier sense multiple access with collision detection. Each access point senses carrier wave to figure out if machine is idle. To transmit, waits until carrier is idle, then starts transmitting. Each transmission consists of a packet; there is a maximum packet size.

- Collision detection and recovery. Transceivers monitor carrier during transimission to detect interference. Interference can happen if two transceivers start sending at same time. If interference happens, transceiver detects a collision.

- When collision detected, uses a binary exponential backoff policy to retry the send. Adds on a random delay to avoid synchronized retries.

- Is there a fixed bound on how long it will take a packet to get successfully transmitted? Is any packet guaranteed to be transmitted at all?

- Addressing. Each host interface has a hardware address built into it. Addresses are 48 bits long. When change host interface hardware, address changes.

- Are three kinds of addresses:

  - Physical address of one network interface.
  - Broadcast address for the network. (All 1's).
  - Multicast addresses for a subset of machines on network.

- Host interface looks at all packets on the ethernet. It passes a packet on to the host if the address in the packet matches its physical address or the broadcast address. Some host interfaces can also recognize several multicast addresses, and pass packets with those addresses on to the host.

- How do vendors avoid ethernet physical address clashes? Buy blocks of addresses from a central authority.

- Packet (frame) format.

  - Preamble. 64 bits of alternating 1 and 0, to synchronize receivers.
  - Destination address. 48 bits.
  - Source address. 48 bits.
  - Packet type. 16 bits. Helps OS route packets.
  - Data. 368-120000 bits.
  - CRC. 32 bits.

  Ethernet frames are self-identifying. Can just look at frame and know what to do with it. Can multiplex multiple protocols on same machine and network without problems. CRC lets machine identify corrupted packets.

- Token-ring networks. Alternative to ethernet style networks. Arrange network in a ring, and pass a token around that lets machine transmit. Message flows around network until reaches destination. Some problems: long latency, token regeneration.

- ARPANET. Ancestor of current Internet. Long-haul packet-switched network. Consisted of about 50 C30 and C300 BBN computers in US and Europe connected by long-haul leased data lines. All computers are dedicated packet-switching machines (PSNs).

- Interesting fact: ARPANET, like highway system, was initially a DOD project set up officially for defense purposes.

- In original ARPANET, each computer connected to ARPANET connected directly to a PSN. Each packet contained address of destination machine and PSN network routed the packet to that machine. Now this is totally impractical and have a much more complex local structure before get onto Internet.

- Design of Internet driven by several factors.

  - Will have multiple networks. Different vendors compete, plus have different technical tradeoffs for local area, wide area and long haul networks.
  - People want universal interconnection.

Will have multiple networks around the world. An internetwork, or internet, connects the different networks. So, job of internet is to route packets between networks.

- One goal of internet: Network transparency. Want to have a universal space of machine identifiers and refer to all machines on the internet using this universal space of machine identifiers. Do not want to impose a specific interconnection topology or hardware structure.
- Internet architecture. Connect two networks using a gateway machine. The job of the gateway is to route packets from one network to another.
- As network topologies become more complicated, gateways must understand how to route data through intermediate networks to reach final destination on a remote network.
- In Internet, gateways provide all interconnections between physical networks. All gateways route packets based on the network that the destination is on.
- Internet addressing. Each host on the Internet has a unique 32-bit address that is used for all Internet traffic to that host. Each internet address is a (netid, hostid) pair. The network identifies the network that the host is on, the hostid identifies the host within the network.
- Three classes of Internet addresses:
  - Class A. First Bit: 0. Bits 1-7: Netid. Bits 8-31: Hostid. Can have 128 Class A networks.
  - Class B. Bits 0-1: 10. Bits 2-15: Netid. Bits 16-31: Hostid. Can have 16,384 Class B networks.
  - Class C. Bits 0-2: 110. Bits 3-23: Netid. Bits 24-31: Hostid. Can have 2 Gig Class C networks.
  - Class D. (multicast addresses). Bits 0-3: 1110. Used for Internet multicast.
  - Class E. Bits 0-3 1111. Reserved.

  See RFC 990 for spec.
- Interesting point. Whole structure of internet is available in RFC's (request for comments). Available over the Internet - use the net search functionality for RFC and you'll find pointers. Can read them to figure out what is going on.
- Gateways can extract network portion of address quickly. Gateways have two responsibilies:
  - Route packets based on network id to a gateway connected to that network.
  - If they are connected to destination network, make sure the packet gets delivered to correct machine on that network.
- Conceptually, an Internet address identifies a host. Exceptions: gateways have multiple internet addresses, at least one per network that they are connected to.
- Because network id is encoded in Internet address, a machine's internet address must change if it switches networks.
- Dotted Decimal notation: Reading Internet addresses. Four decimal integers, with each integer representing one byte.
  - cs.stanford.edu - 36.8.0.47 (what kind of network is it on).
  - cs.ucsb.edu - 128.111.41.20
  - ecrc.de - 141.1.1.1
  - lcs.mit.edu - 18.26.0.36
  - sri.org - 199.88.22.5
- Who assigns internet addresses? The Network Information Center! A centralized authority. It just allocates network ids, leaving requesting authority to allocate host ids.
- Do example on page 45.

- Mapping Internet addresses to Physical Network addresses. Will discuss case when physical network is an Ethernet. Given a 32 bit Internet address, gateway must map to a 48 bit Ethernet address. Uses Address Resolution Protocol (ARP).
- Gateway broadcasts a packet containing the Internet address of the machine that it wants to send the packet to. When machine receives packet, it sends back a response containing its physical address. Gateway uses physical address to send packet directly to machine.
- Also works for machines on same network even when they are not gateways.
- Use a address resolution cache to eliminate ARP traffic.
- ARP request and response frames have specific type fields. An ARP request has a type field of 0806, responses have 8035. Standard set up by the Ethernet standard authority.
- How does a machine find out its Internet address? Store it on disk, and looks there to find out when it boots up. What if it is diskless? Contacts server and finds it out there using Reverse ARP (RARP). RFC 903 - Ross Finlayson, etc.
- RARP request is broadcasted to all machines on network. RARP server looks at physical address of requestor and sends it a RARP response containing the internet address. Usually have a primary RARP server to avoid excessive traffic.
- Now switch to talking about IP - the Internet Protocol. The internet conceptually has three kinds of services layered on top of each other: Connectionless, unreliable packet delivery service, reliable transport service, and application services. IP is the lowest level - the packet delivery.
- The basic unit of transfer in the Internet is the IP datagram. IP datagram has header and data. Header contains internet addresses and the Internet routes IP datagrams based on Internet addresses in header.
- Internet makes a best effort attempt to deliver each datagram, but does not deal with error cases. In particular, can have:
  - Lost Packets
  - Duplicated Packets
  - Out of order Packets

  Higher level software layered on top of IP deals with these conditions.
- IP packets always travel from gateway to gateway across physical networks. If the IP packet is larger than the physical network frame size, the IP packet will be fragmented: chopped up into multiple physical packets. IP is designed to deal with this situation and provides for fragmentation.
- Once a packet has been fragmented, must be reassembled back into a complete packet. Usually reassembled only when fragments reach final destination. But, could build a system that reassembled fragments when got to a physical network with a larger frame size.
- Why is there a need for possibility of fragmentation? No good way to impose a uniform packet size on all networks. Some networks may support large packets for performance, while others can only route small packets. Should not prevent some networks from using large packets just because there exists a network somewhere in the world that can not handle large packets. But must be able to route large packets through a network that only handles small packets - network transparency.
- Important fields in IP header:
  - VERS: protocol version.
  - LEN: length of header, in 32-bit words.
  - TOTAL LEN: total length of IP packet.
  - SOURCE IP ADDRESS: IP address of source machine.
  - DEST IP ADDRESS: IP address of destination machine.

- TTL: time to live. How many hops the packet may take without getting removed from Internet. Every time a gateway forwards the packet, it decrements this field. Required to deal with things like cycles in routing, etc.
  - IDENT: packet indentifier. Unique for each source. Typically, source maintains a global counter it increments for every IP datagram sent.
  - FLAGS: A do not fragment flag (dangerous) and a more fragments flag - 0 marks end of datagram.
  - FRAGMENT OFFSET - gives offset of this fragment in original datagram.

- How to reassemble a fragmented packet? Allocate a buffer for each packet. Use IDENT and SOURCE IP ADDRESS to identify the original datagram to which the fragment belongs. Use the FRAGMENT OFFSET field to write each fragment into correct spot in the buffer. Use more fragments flag to find end of original datagram. Use some mechanism to make sure all fragments arrived before consider datagram complete.

- Routing IP datagrams. There are multiple possible paths between hosts in an internet. How to decide which path for which datagram?

- Routing for hosts on same network. Realize that are on same network by looking a netid field of Internet address, and just use underlying physical network.

- Routing for hosts on different networks. Gateways pass datagrams from network to network until reach a gateway connected to destination network.

- Each gateway must decide next gateway to send datagram to.

  - Source routing. The source specifies the route in the datagram. Useful for debugging and other cases in which Internet should be forced to use a certain route.
  - Host-specific routes. Can specify a specific route for each host. Used mostly for debugging.
  - Table driven routing. Each gateway has a table indexed by destination network id. Each table entry tells where to send datagrams destined for that network. Do example on page 82.
  - Default routes. Specify a default next gateway to be used if other routing algorithms don't give a route.

  Most routers use a combination of table driven routing and default routing. They know how to route some packets, and pass others along to a default router. Eventually, all defaults point to a router that knows how to route ALL packets.

- How are routing tables acquired and maintained? There are a lot of different protocols, but the basic idea is that the gateways send messages back and forth advertising routes. Each advertisement says that a specific network is reachable via N hops. Some protocols also include information about the different hops. The gateways use the route advertisements to build routing tables.

- Internet was originally designed to survive military attacks. It has lots of physical redundancy and its routing algorithm is very dynamic and resilient to change. If a link goes away, the network should be able to route around the failure and still deliver packets. So, routing tables change in response to changes in the network.

- In practice doesn't always work as well as designed. Chief threat to Internet links these days is backhoes, not bombs. Common error is routing all of the links that are supposed to give physical redundancy in the same fiber run, so are vulnerable to one backhoe.

- In original internet, partition gateways into two groups. Core and noncore gateways. Core gateways have complete information about routes. Original core gateways used a protocol called GGP (Gateway to Gateway Protocol) to update routing tables.

- GGP messages allow gateways to exchange pairs of messages. Each message advertise that the sender can reach a given network N in D hops. Receiver compares its current route to the new route through the sender, and updates its tables to use the new route if it is better.

- Famous case: Harvard gateway bug. Memory fault caused it to advertise a 0 hop route to everybody!
- Problem with GGP - distributed shortest path algorithm may take a long time to converge.
- Later algorithm (SPF) replicated a complete database of network topology in every gateway. Gateway runs a local shortest path computation to build its tables.
- In current Internet, there is no longer any central backbone or authority. Instead, have internet providers. The whole system has switched over to private enterprise.
- A top-down view of system. There are 4 Network Access Providers. Each NAP is a very fast router connected via high-capacity lines to other gateways and NAPs. Lines may be T3 (644 Mb/s) lines. Typically big communications companies (MCI, Sprint, ATT) own the lines. Lines are typically fiber.
- Organizations go to internet providers to get access to the internet. An internet provider buys a bunch of routers (usually from Cisco) and leases a bunch of lines. The internet provider must also buy access to a NAP or to a gateway that leads to a NAP. The routers talk a route advertisement protocol and implement some routing algorithm.
- The internet provider can then turn around and sell internet access to whoever wants to buy it. UCSB buys its internet access from CERFNET, and it pays $23,000 per year for its internet access. All of the UC schools will band together and buy internet access from MCI, getting more bandwidth but at a higher price.
- Check out `http://www.cerf.net` to see an Internet topology.
- Organizations tend to chop their communications up into multiple networks, so there are too many networks in the world to give every network an Internet address. For example, the UCSB CS department has more than 10 networks.
- The solution is subnetting. Internet views whole organization as having one network. The organization itself chops the host part of IP address up into a pair of local network and local host. For example, UCSB has one class B Internet network. The third byte of every IP address identifies a local network, and the fourth byte is the host on that network.
- All IP packets from outside come to one UCSB gateway (by default). As far as the Internet is concerned, all of UCSB has only one network.
- Inside UCSB, there is a set of networks connected by routers. These routers interpret the IP address as containing a local network identifier and a host on that network, and route the packet within the UCSB domain. The routers periodically advertise routes using a protocol called RIP.
- This is an example of hierarchical routing. Internet routes to UCSB gateway based on Internet network id, then routers within UCSB route based on the subnet id.
- traceroute command tells you the gateways packets go through to get to a given location. Here are a few:

```
cheetah > traceroute minnie (CSIL Lab, UCSB)
traceroute to minnie (128.111.42.17), 30 hops max, 40 byte packets
 1  toons (128.111.49.2)  17 ms *  3 ms
 2  minnie (128.111.42.17)  3
cheetah > traceroute ecrc.de (Munich, Germany)
traceroute to ecrc.de (141.1.1.1), 30 hops max, 40 byte packets
 1  lo-galaxy (128.111.49.1)  6 ms  3 ms  3 ms
 2  ecigw1-41 (128.111.41.1)  4 ms  12 ms  4 ms
 3  cerfgw (128.111.254.201)  6 ms  12 ms  7 ms
 4  uclagw.cerf.net (134.24.107.104)  24 ms  22 ms  22 ms
 5  sdsc-ucla.cerf.net (134.24.101.100)  44 ms  263 ms  116 ms
 6  nynap-sdsc-atm-ds3.cerf.net (134.24.17.200)  141 ms  111 ms  126 ms
 7  sprintl.sprint.ep.net (192.157.69.9)  124 ms  147 ms *
 8  sl-pen-1-F0/0.sprintlink.net (144.228.60.1)  170 ms  114 ms  122 ms
 9  sl-dc-6-H2/0-T3.sprintlink.net (144.228.10.33)  123 ms  122 ms  132 ms
10  icm-dc-2b-F1/0.icp.net (144.228.20.103)  126 ms  212 ms  119 ms
11  icm-dc-1-F0/0.icp.net (198.67.131.36)  122 ms  214 ms  156 ms
12  icm-ecrc-1-S0-1984k.icp.net (198.67.129.18)  218 ms  209 ms  223 ms
13  ECRC-RBS.ECRC.DE (193.23.5.97)  280 ms  536 ms  315 ms
14  ECRC-GW.ECRC.DE (192.109.251.254)  297 ms  215 ms  236 ms
15  ecrc.de (141.1.1.1)  219 ms *  343 ms
cheetah > traceroute rain.org (Santa Barbara, CA)
traceroute to rain.org (198.68.144.2), 30 hops max, 40 byte packets
 1  lo-galaxy (128.111.49.1)  7 ms  3 ms  3 ms
 2  ecigw1-41 (128.111.41.1)  5 ms  5 ms  5 ms
 3  cerfgw (128.111.254.201)  7 ms  6 ms  6 ms
 4  uclagw.cerf.net (134.24.107.104)  57 ms  40 ms  22 ms
 5  sdsc-ucla.cerf.net (134.24.101.100)  44 ms  54 ms  57 ms
 6  ucop-sdsc.cerf.net (134.24.52.112)  84 ms  62 ms  61 ms
 7  sl-ana-3-S2/6-T1.sprintlink.net (144.228.73.81)  85 ms  81 ms  75 ms
 8  sl-ana-1-F0/0.sprintlink.net (144.228.70.1)  162 ms  139 ms *
 9  sl-fw-6-H2/0-T3.sprintlink.net (144.228.10.29)  125 ms  138 ms  158 ms
10  sl-fw-3-F0/0.sprintlink.net (144.228.30.3)  141 ms  184 ms  121 ms
11  sl-rain-network-1-S0-T1.sprintlink.net (144.228.171.2)  175 ms  165 ms  153 ms
12  coyote.rain.org (198.68.144.2)  149 ms  192 ms  184 ms
cheetah > traceroute cs.orst.edu (Corvallis, Oregon)
traceroute to cs.orst.edu (128.193.32.1), 30 hops max, 40 byte packets
 1  lo-galaxy (128.111.49.1)  7 ms  3 ms  3 ms
 2  ecigw1-41 (128.111.41.1)  5 ms  10 ms  10 ms
 3  cerfgw (128.111.254.201)  8 ms  9 ms  8 ms
 4  * uclagw.cerf.net (134.24.107.104)  37 ms *
 5  uci-la-smds.cerf.net (134.24.95.1)  38 ms  33 ms  30 ms
 6  * * ucop-sf-ds3-smds.cerf.net (134.24.9.112)  64 ms
 7  border3-hssi1-0.SanFrancisco.mci.net (149.20.64.9)  70 ms  71 ms  51 ms
 8  core-fddi-0.SanFrancisco.mci.net (204.70.2.161)  75 ms *  103 ms
 9  core-hssi-2.Seattle.mci.net (204.70.1.49)  74 ms  79 ms  81 ms
10  border1-fddi0-0.Seattle.mci.net (204.70.2.146)  87 ms  72 ms  142 ms
11  nwnet.Seattle.mci.net (204.70.52.6)  68 ms  272 ms  238 ms
12  seabr1-gw.nwnet.net (192.147.179.5)  225 ms  103 ms  75 ms
13  seattle1-gw.nwnet.net (198.104.194.195)  84 ms *  162 ms
14  portland1-gw.nwnet.net (192.80.12.81)  102 ms  137 ms  161 ms
15  osu-gw.nwnet.net (198.104.196.121)  186 ms  151 ms  202 ms
16  orst3-gw.ORST.EDU (192.147.167.1)  127 ms *  91 ms
17  ece-gw-out.ece.ORST.EDU (128.193.8.40)  86 ms *  95 ms
18  CS.ORST.EDU (128.193.32.1)  100 ms  89 ms *
```