# Ethernet and IP

*A slightly less introductory networking class*

Drew Saunders
Networking Systems
Stanford University

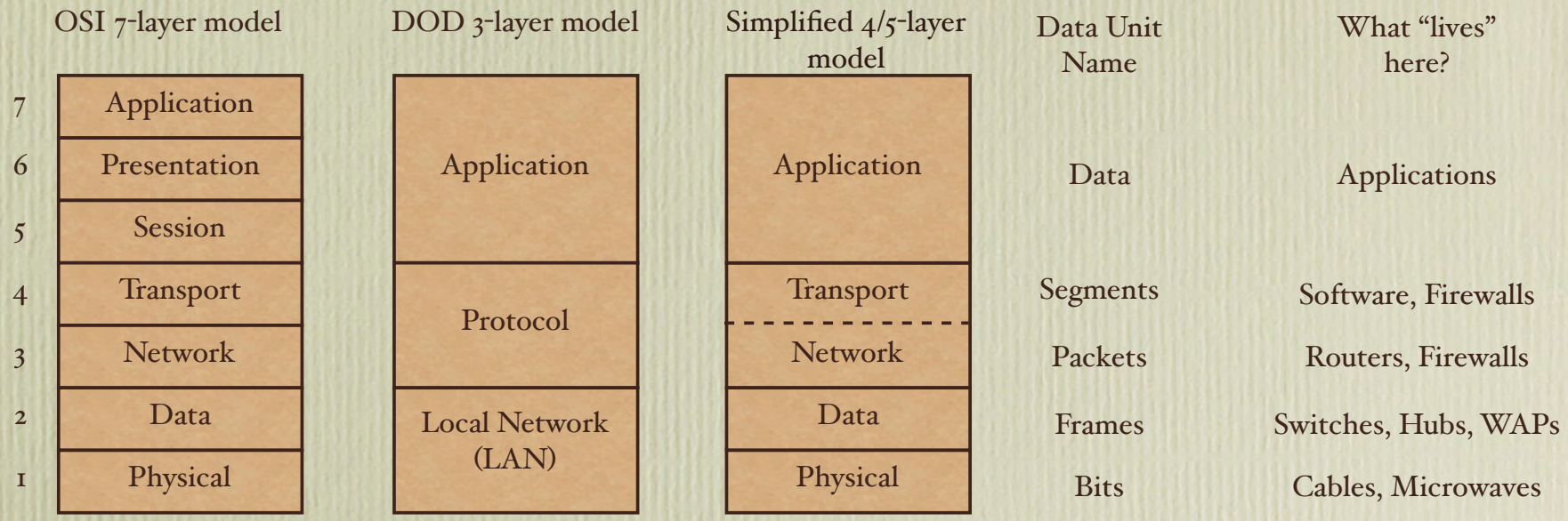
# Goals of Class

- Slightly more in-depth knowledge of Ethernet.
- Internet Protocol, TCP, UDP, ARP, ICMP.
- IP addressing, subnet masks.
- Some troubleshooting of ethernet and IP problems.

# Not goals of class

- Detailed understanding of TCP/IP.
- Server or desktop maintenance or configuration.
- How to use email, web, etc.
- Routing, SUNet details.

# Our friend, the 7-layer networking model.




OSI (Open Systems Interconnection) mnemonic: **A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing

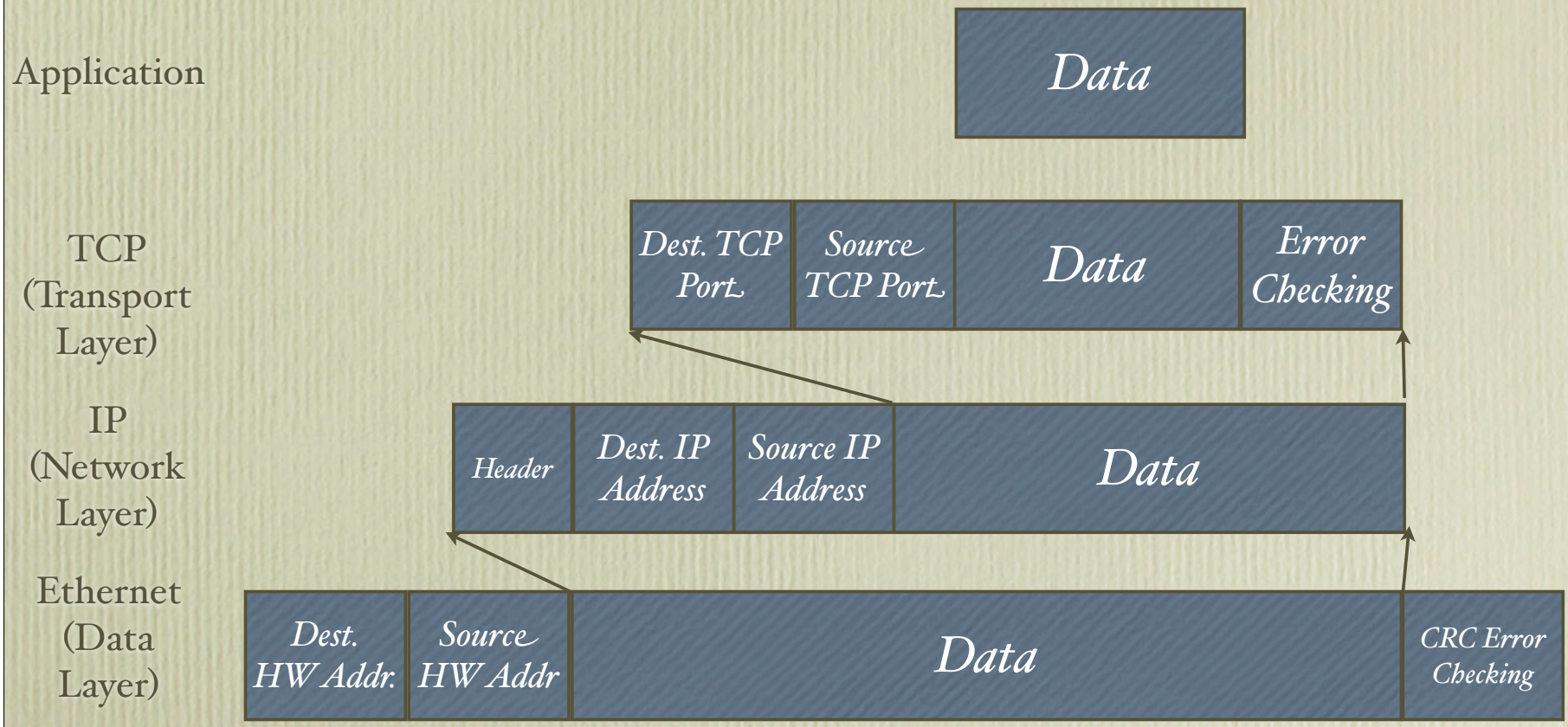
We'll be using the "simplified model," and talking mostly about the Network (IP) and Transport (TCP and UDP) layers, plus a little about Data (Ethernet details).

Although not strictly correct, people tend to call everything from layer 4 down a "packet."

# Networking layers compared to the postal system.

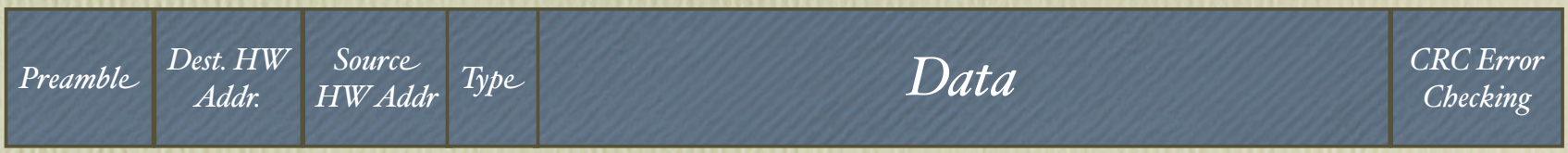
- What are you sending? (Application)
- Where am I going? (Addressing, DNS)
- How do I get there? (Routing)
- What am I packaged in? (Network, Data)
- What carries me? (Data, Physical)
- What are the traffic rules on my road/train/flight? (Data, Physical)

# Encapsulation




Postal system: Letter (Application) gets letterhead (TCP) gets put in an envelope (IP) gets put in a mail bag (Ethernet) gets carried on a mail truck (Cat5, wireless, etc. - Physical layer)
Each layer doesn't have to worry about all the contents inside, each layer can handle its own job and pass it on to the next destination, i.e.: the post office can move whole bags without having to look at each letterhead on each letter in each envelope every time.

# More on Ethernet




- Preamble: 64 alternating 0's and 1's to synchronize the computers.
- Destination: 48-bit (usually done as 12 hex characters) address of the intended receiver, all 1's (FFFF...) for broadcast.
- Source: Your own HW address.
- Type: Ethernet II is the only type really used any more, there are 3 others for legacy Layer 3 protocols (Appletalk, IPX, etc.)
- Data: The contents, which aren't the concern of the ethernet layer, but are to be handed off to the next layer.
- CRC: Cyclic Redundancy Check. When the computer builds the packet, it does some math based on the packet (which is, after all, just 0's and 1's) and puts the result of that calculation at the end. If the receiver does the same math and gets a different result, the packet is rejected and an error is sent back to the sender.

# Common Ethernet Problems

- Physical layer: Is the cable plugged in? Is it broken? Did someone run their chair over it for the last time? “It worked fine yesterday!”
- Speed mis-match. Most modern computers and network equipment can speak at either 10Mbps *or* 100Mbps *or* 1000Mbps, but each end has to agree. Auto-negotiation may fail, so one end or both may have to be forced to a specific speed.
- Duplex mis-match. Most modern computers and network equipment use full duplex only (they can send *and* receive at the same time), older equipment may be half duplex (send *or* receive). Duplex mis-matches don’t cause the link to be down, but cause a lot of chopped up data and retransmits and slowness.
- MDI/MDIX (Medium Dependent Interface (Crossover)) problem. When connecting a switch to switch, or computer to computer, you usually need to use a crossover cable, otherwise, you have transmit “talking” to transmit, and receive “talking” to receive. Auto-MDIX can allow you to use a regular cable, but doesn’t always work.

# IP Packet


- Header: Lots of stuff! Version (IPv4, IPv6); Header length (usually 5 32-bit words); total packet length; unique ID and fragment offset of the datagram to reassemble all the IP packets in the right order; time to live (how many router hops before you give up and toss the packet?); Protocol of the contents (TCP, UDP, etc.); Header Checksum (error checking for the header only).
- IP addresses: Like hardware, destination first, then you announce yourself as the source.
- The total header (including addresses) needs to be a multiple of 32 bits (usually 5), so there can be padding after the source address and before the data.

# IP Addressing

- IP addresses are in the form of 4 octets (32 bits), each from 0-255.
  - IPv6 uses 128 bits and is expressed in hex.
- Like phone numbers, they go from general to specific as you go left to right
  - But unlike phone numbers, the "area code" can be of variable size.
- 171.64.20.23 (the host "networking.stanford.edu"), 171.64. means "Stanford.edu" 20. means "Networking group subnet" and .23 means "networking."
- The division between network, subnet and host is determined by subnet masks and CIDR prefixes. Your host needs to be told its subnet mask.
- There used to be a way of determining network size by network "Class." Classes were A, B, C, D. Nobody uses "classful" routing any more, everything is done as "classless" routing. CIDR is "Classless Inter-Domain Routing" and is pronounced like the apple beverage.

# Subnet Masks

- From a networking point of view, your host knows only two things: “Local to my net” and “give it to the router.”
- The subnet mask tells your host whether another IP number is “Local” or “The Router’s Problem.” It draws the line in the sand for you.
- Subnet masks are also in the form of four octets, but you’ll generally see them in the form of 255.255.255.0 or something like that.
- Convert the subnet mask to binary, you’ll have a string of 32 digits, a long series of 1’s, followed by a series of 0’s. Where the 1’s end and the 0’s begin is your subnet boundary. 255.255.254.0 is 23 “1’s” then 9 “0’s.”
- If your subnet mask is wrong, you may send things “locally” that belong to the router, which will probably fail, or send things to the router that are local, which will work but is excess work for the router.

# CIDR Prefixes

- Since the old method of delimiting an IP range by "Class" is gone, we need a way to designate how large a network is. Subnet masking is what the host (or router) uses, but it's rather tedious to write out, so the subnet mask is turned into a number and referred to as the "CIDR Prefix" even though it goes at the end of the network's IP address designation.
- 255.255.254.0 is 23 1's, so that is a /23 CIDR prefix, and we would write out that IP range as "171.64.20.0/23" which translates into "171.64.20.0 through 171.64.21.255" or 512 IP numbers (but not all are available).
- As the CIDR number gets smaller, the subnet mask line moves "left" and the size of the network gets bigger. 171.64.0.0/16 means "171.64.0.0 through 171.64.255.255" (65,536 IP numbers) and is a handy way for routers to lump huge subnets together and make routing easier. The rest of the world knows Stanford as "171.64.0.0/14" (171.64.0.0 through 171.67.255.255, 262,144 IP numbers) and can more easily route to us with just that simple designation.

# CIDR Prefix to Subnet Cheat-sheet for common Stanford network sizes

| *Prefix* | *Subnet Mask* | *Example Net.* | *SU Router IP #* | *Broadcast IP#* | *Usable IP #s* |
|---|---|---|---|---|---|
| /20 | 255.255.240.0 | 172.31.16.0/20 | 172.31.16.1 | 172.31.31.255 | 4093 |
| /21 | 255.255.248.0 | 172.31.16.0/21 | 172.31.16.1 | 172.31.23.255 | 2045 |
| /22 | 255.255.252.0 | 172.31.16.0/22 | 172.31.16.1 | 172.31.19.255 | 1021 |
| /23 | 255.255.254.0 | 172.31.16.0/23 | 172.31.16.1 | 172.31.17.255 | 509 |
| /24 | 255.255.255.0 | 172.31.16.0/23 | 172.31.16.1 | 172.31.16.255 | 253 |
| /25 | 255.255.255.128 | 172.31.16.128/25 | 172.31.16.129 | 172.31.16.255 | 125 |
| /26 | 255.255.255.192 | 172.31.16.0/26 | 172.31.16.1 | 172.31.16.63 | 61 |
| /27 | 255.255.255.224 | 172.31.16.160/27 | 172.31.16.161 | 172.31.16.191 | 29 |
| /28 | 255.255.255.240 | 172.31.16.144/28 | 172.31.16.145 | 172.31.16.159 | 13 |
| /29 | 255.255.255.248 | 172.31.16.8 | 172.31.16.9 | 172.31.16.15 | 5 |

# “Usable” IP numbers?

- In any subnet, at least 3 IP numbers are taken up. The network itself takes the “bottom” number. In the case above, 171.64.20.0 is the IP number that designates the network itself.
- Just like in ethernet, we need a way to talk to everyone all at once, and that’s a broadcast. That’s the “top” number, or 171.64.21.255 in the example above.
- If you want to ever be able to send IP packets to another network, you’ll need a router, and it needs at least one IP number (and often more). At Stanford, we use the “network+1,” so 171.64.20.1 is the router for 171.64.20.0/23. The router can have any IP number, but most people use one near the bottom or top of the network range.
    - Cisco HSRP (Hot-Swappable Router Protocol) uses two routers that “share” the one IP number, so .1 would be for the active router, while .2 and .3 are the actual two routers. If the active .1 “dies” the other takes over that address.

# Chopping up a /24 net.

| /24 (253 addresses) | /25 (125 ea.) | /26 (61 ea.) | /27 (29 ea.) | /28 (13 ea.) |
|---|---|---|---|---|
| 172.31.16.0/24 | 172.31.16.0/25 | 172.31.16.0/26 | 172.31.16.0/27 | 172.31.16.0/28 |
| | | | | 172.31.16.16/28 |
| | | | 172.31.16.32/27 | 172.31.16.32/28 |
| | | | | 172.31.16.48/28 |
| | | 172.31.16.64/26 | 172.31.16.64/27 | 172.31.16.64/28 |
| | | | | 172.31.16.80/28 |
| | | | 172.31.16.96/27 | 172.31.16.96/28 |
| | | | | 172.31.16.112/28 |
| | 172.31.16.128/25 | 172.31.16.128/26 | 172.31.16.128/27 | 172.31.16.128/28 |
| | | | | 172.31.16.144/28 |
| | | | 172.31.16.160/27 | 172.31.16.160/28 |
| | | | | 172.31.16.176/28 |
| | | 172.31.16.192/26 | 172.31.16.192/27 | 172.31.16.192/28 |
| | | | | 172.31.16.208/28 |
| | | | 172.31.16.224/27 | 172.31.16.224/28 |
| | | | | 172.31.16.240/28 |

# Let's do some subnet math!

*Tape up your glasses and tuck in your pocket protectors!*

- Your host, 172.31.12.20 is on a /27 network. Is 172.31.12.40 on your net?
- Convert 172.31.12.20 to binary, convert 172.31.12.40 to binary.
  - `10101100.0001111.00001100.00010100` and `10101100.0001111.00001100.00101000`
- Write 27 1's in a row, fill out the rest of the 32 bits with 0's, divide into 4 8-bit segments to make it easier to follow.
  - `11111111.11111111.11111111.11100000`
- Where the 1's end and the 0's begin on the subnet mask, if everything to the left is the same, then it's the same subnet. If there's any difference, then it's not the same subnet, and you pass those packets on to the router.

# Stanford's IP numbers

| *IP range* | *CIDR Prefix* | *Netmask* | *Use* |
|---|---|---|---|
| 128.12.0.0 | /16 | 255.255.0.0 | Dorms |
| 171.64.0.0 | /14 | 255.252.0.0 | Stanford's main IP range |
| 171.64.0.0 | /16 | 255.255.0.0 | Used for main campus |
| 171.65.0.0 | /16 | 255.255.0.0 | Medical Center, Hospital |
| 171.66.0.0 | /16 | 255.255.0.0 | Remote, DSL, Modems, some departments |
| 171.67.0.0 | /16 | 255.255.0.0 | Main campus, large blocks |
| 68.65.160.0 | /21 | 255.255.248.0 | Cable Modem |
| 204.63.224.0 | /21 | 255.255.248.0 | Livermore Auxiliary Data Center |

See this list and more at https://www.stanford.edu/services/lnaguide/lnasonly/net_numbers.html

# Private IP numbers

- There are groups of IP numbers that by definition cannot be routed onto the Internet. ISP's (Internet Service Providers) don't allow these numbers to be routed to them from customers. If you want to read the gory details, look up "RFC 1918" on the web.
- Those IP ranges are 10.0.0.0/8 (a huge range), 172.16.0.0/12 and 192.168.0.0/16.
- Stanford uses a lot of private addresses. Conveniently, 172.24.0.0/14 looks similar enough to 171.64.0.0/14 that we can associate a 172.2x.y.z IP space with every 171.6x.y.x IP space. I.e.: 171.66.175.0/24 gets 172.26.175.0/24 associated with it.
- Since these IP numbers can't go off campus, they're very useful for network equipment, printers, etc. Anything that doesn't have someone sitting at it expecting to be able to get to off campus resources might be a good candidate for a private IP number. This is *not* a substitute for security!
- Private IP ranges that are defined in NetDB are (usually) routed on campus, so any host with one of those numbers can be accessed by any other on campus host.

# IPv6

- IPv6 is IP Protocol version 6, as opposed to the current IPv4.
- IPv4 provides ~4.3 billion IP numbers (4,294,967,296 to be exact), but there's a lot of inefficiency, plus a lot of demand. IPv6 provides 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses. That should be sufficient for a very long time.
- IPv4 is out of assignable IP number blocks. The main benefit of IPv6 is a (hopefully) inexhaustible quantity of IP numbers. The main problem with IPv6 is translating between IPv4 and IPv6. As there is no entity that can demand that IPv4 no longer be routed, organizations will have to route both IPv4 and IPv6 or tunnel or translate for many years.
- IPv6 uses CIDR prefixes, just like IPv4. Stanford's IPv6 range is 2607:F6D0::/32, which provides for $2^{96}$ numbers (a whole lot). We'll allocate subnets as /64, or $2^{64}$ numbers (still quite a bit). Pine Hall, where we're testing IPv6 is using 2607:f6d0:c001::/64.
- We don't expect to be needing to do much with IPv6 at Stanford, but are ready when the demand arises.

# NAT (Network Address Translation)

- NAT, Network Address Translation, allows devices (usually routers, sometimes special NAT-only devices) to associate one public IP number with one or more private IP numbers.
- NAT is one of the main reasons why IPv4 has lasted as long as it has and IPv6 isn't as popular.
- At Stanford, we use 10.32.0.0/15 for main campus wireless, which we then associate through NAT with a real 171.66.x.x IP range. This allows us to handle the ever increasing number of wireless users.
- At Stanford, for wireless guest, we use NAT to associate the small public IP range of 68.65.169.128/25 with the huge internal IP range of 10.22.0.0/16.

# ARP:
# Bridging Ethernet and IP

- ARP (Address Resolution Protocol) allows you to associate a layer 3 (Network) address to a layer 2 (Data) address. It can be used by many different layer 2 or 3 protocols, but we'll talk about Ethernet and IP, which are functionally the only protocols that most anyone uses any more.
- Post office analogy: You have a letter going to a big company, you need to associate the person's name (like their IP address) with the company street address (like the ethernet address). The post office (ethernet) only cares about the street address, the company's internal mail people (IP) worry about getting it to the person.
- ARP uses an ethernet broadcast (Hey, everybody!) containing a non-broadcast IP packet (Who's 10.8.24.13?) to get a specific ethernet address associated in its ARP table with that particular IP address.
- Since you announce both your ethernet and IP source in your broadcast, the recipient can put you in its ARP table without needing to do a broadcast.
- You can have multiple IP numbers associated with one ethernet address in your ARP table, but not the other way around.

# TCP and UDP

- These are at layer 4 (Transport) and they route the data to the host process. I.e.: send a web browser (TCP port 80) request to the web server process (at the application layer).
- Port numbers (from 0 to 65535) are the addressing method. Most are assigned, some are dynamic. See http://www.iana.org/assignments/port-numbers
- You need both the destination and your source port, so you can get a reply to your correct application, i.e: when you launch a web browser and go to a web site, you want your computer do display the reply data in your web browser, not your IM client.
- TCP stands for "Transmission Control Protocol." TCP packets require an acknowledgment from the receiver so qualify as a "connection" protocol. TCP is used for reliability, and flow control, and TCP packets can be reassembled into a large piece of data. Like a series of letters with return receipt stickers and "1 of 20" "2 of 20" on them.
- UDP, "User Datagram Protocol" is connectionless (no acknowledgment), and can't be reassembled. Quick and simple. Like a postcard.

# Common TCP applications

- Telnet (port 23). SSH (22). Both require you to know that the character you typed was received, so you get an acknowledgment for each keystroke!
- HTTP (80) and HTTPS (443). Hypertext Transfer Protocol. You want to get the whole web page, which is often huge, so you need your packets chopped up and sent in segments. HTTPS is secure HTTP over SSL (Secure Sockets Layer).
- SMTP (25). Simple Mail Transfer Protocol. You know this better as email.
- Most of the software you usually interact with will use TCP for the reliability granted by the acknowledgment, plus the convenience of handling large data packages and reassembling them in the correct order.

# Common UDP applications

- Remember: UDP is a single packet, no acknowledgment, so it's for things that need to be fast and aren't so upset about an error or two.
- DNS (53). Domain Name System. If you don't get the reply, wait a bit and ask again.
- DHCP (67 and 68). Dynamic Host Configuration Protocol. Needs to be quick and simple, and you can always try again.
- SNMP (161 and 162). Simple Network Management Protocol. Lets you get information and send configurations to remote devices.
- Games. Many prefer speedy responses, so UDP is a good match.
- Except for games, most of these will be more lower level operating system applications instead of applications that the user interacts with regularly. A lot of network tools use SNMP.

# ICMP

- ICMP, "Internet Control Message Protocol."
- Behaves a lot like UDP, but isn't. ICMP is considered a layer 3 (Network) protocol, UDP is layer 4. ICMP is still part of IP, so here's where the layer boundaries get a little fuzzy.
- ICMP is mostly used by the networking stack of the operating system to send errors. ICMP isn't normally used by user applications.
- Ping is about the only user application that uses ICMP.
  - Ping sends an ICMP echo request, and reports if that echo request has been received. A host can ignore ping.
  - Ping can be used to attack a machine, so it limited at the Stanford border.
- Tracert on Windows (not to be confused with traceroute) also uses ICMP, which is a bad design. Trace*route* uses UDP packets so it's not blocked by ICMP filters.

# Common IP layer problems

- Mis-configured computers:
  - Duplicate IP numbers, a.k.a. the "IP Thief." When you ARP to associate an ethernet address with an IP number, and you get two ethernet addresses, who gets the packet?
  - Wrong router information: If you need to leave your local net, and don't know who the router is, you can never get your packets off your local net.
  - Wrong subnet mask: You may not send packets to the router that you erroneously think are "local" and they won't get anywhere.
  - Incorrect DNS information. Not enough time to cover DNS in this class, but you still need the right DNS information.
- DHCP hands out the correct, (and checks to see if it's "stolen"), IP number plus your subnet mask, router, and DNS servers. We like DHCP!

# Questions?