

Unix System Administration

Chris Schenk

Lecture 08 – Tuesday Feb 13

CSCI 4113, Spring 2007

ARP Review



Host A
128.138.202.50
00:0B:DB:A6:76:18



Host B
128.138.202.53
00:11:43:70:45:81



Switch



Host C
128.138.202.71
06:57:AA:FB:8B:3C



Gateway
128.138.202.1
00:0B:DB:01:5F:A7

Ping Host A (128.138.202.50)



Host A
128.138.202.50
00:0B:DB:A6:76:18



Host B
128.138.202.53
00:11:43:70:45:81



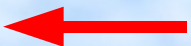
Switch



Host C
128.138.202.71
06:57:AA:FB:8B:3C



Gateway
128.138.202.1
00:0B:DB:01:5F:A7



% ping 128.138.202.50

Who owns 128.138.202.50?



Host A
128.138.202.50
00:0B:DB:A6:76:18



Host B
128.138.202.53
00:11:43:70:45:81



Switch

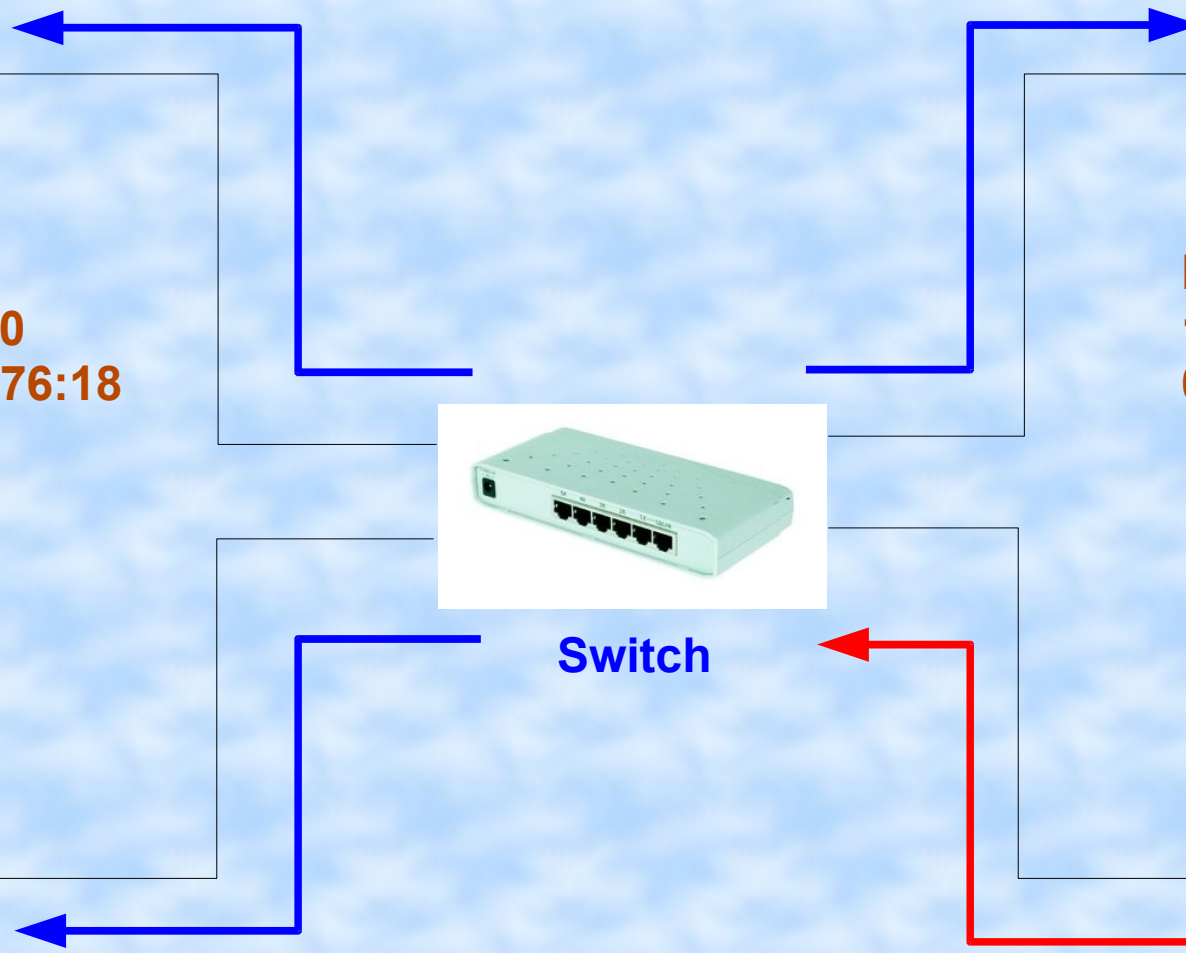


Host C
128.138.202.71
06:57:AA:FB:8B:3C



Gateway
128.138.202.1
00:0B:DB:01:5F:A7

**ARP broadcast to
FF:FF:FF:FF:FF:FF**



Host A owns 128.138.202.50



Host A
128.138.202.50
00:0B:DB:A6:76:18



Host B
128.138.202.53
00:11:43:70:45:81



Switch

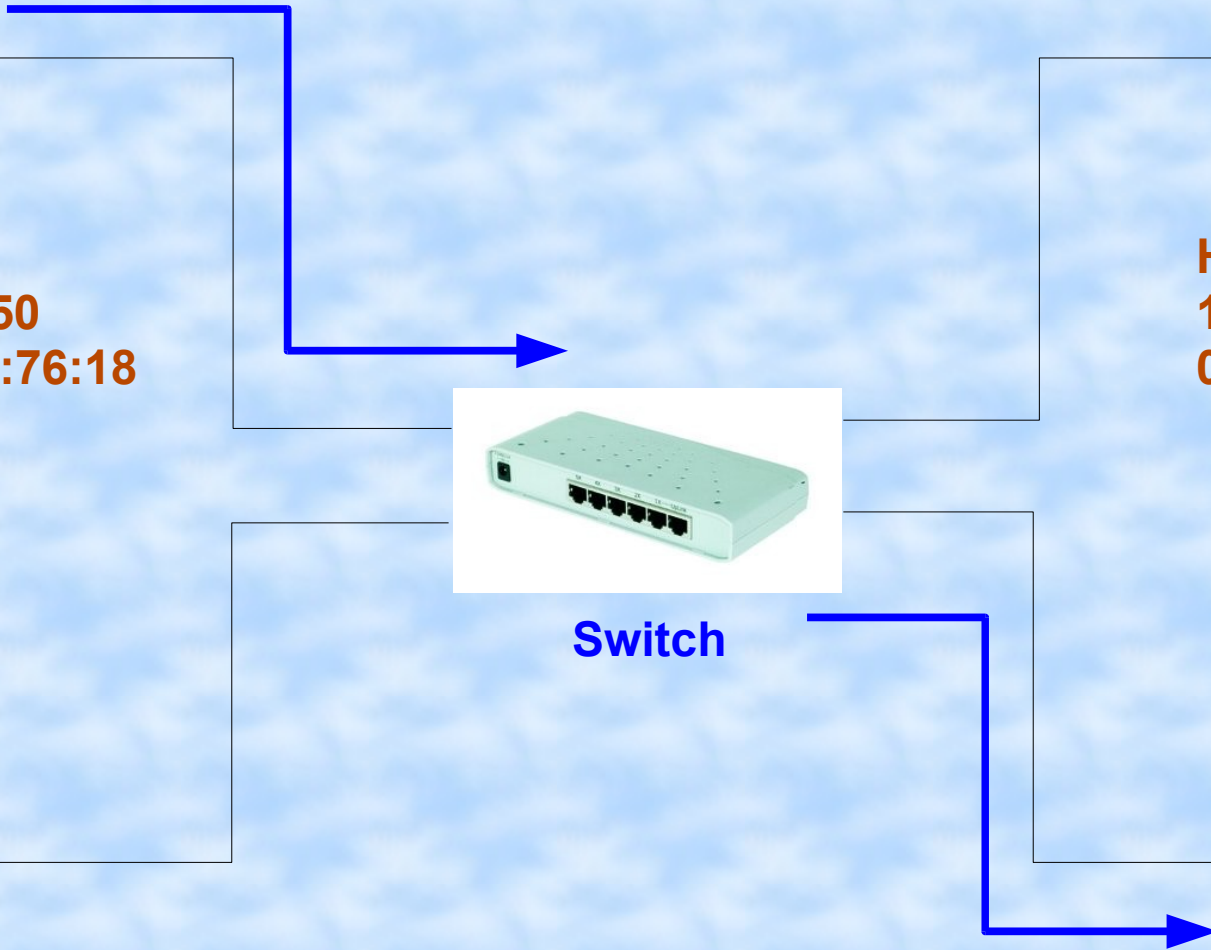


Host C
128.138.202.71
06:57:AA:FB:8B:3C



Gateway
128.138.202.1
00:0B:DB:01:5F:A7

ARP reply to
00:0B:DB:01:5F:A7



Link Layer – ARP and RARP

- ARP – Address Resolution Protocol
 - Used to resolve IP address -> MAC address mappings
 - Used in all switching hardware!
 - Local broadcast segments only (LANs)
 - This is how packets are routed on a LAN!
- RARP – Reverse Address Resolution Protocol
 - Used (sometimes) to obtain an IP address at boot time
 - **rarpd** server with MAC address -> IP address mappings

Layer 1 Notes

- MAC == Media Access Control
 - This is a protocol!
 - 48-bit addresses burned into ROMs
 - Can be overridden by the user
- No authentication of any kind
 - ARP cache poisoning
 - Gratuitous ARP replies to FF:FF:FF:FF:FF:FF
 - Man-In-The-Middle attacks (acting as gateway)
 - Access to local network is always a risk!
- How can we monitor for bad activity?
 - Lots of ARP requests much sooner than timeout

Network Layer (2)

- Internet Protocol (IP) is defined as:
 - Unreliable
 - no guarantee that a packet reaches the destination
 - Stateless (connectionless)
 - No notion of a multi-packet IP session
 - See a packet, route it, and forget about it
- How are packets handled in terms of MTU splitting?
 - information duplicated in each packet

Network Layer – IP Header

BYTE

0	4-bit version	4-bit hdr length	8-bit type-of-service	16-bit total length in bytes	
5	16-bit ID number			3-bit flags	13-bit fragment offset
9	8-bit time-to-live		8-bit protocol	16-bit header checksum	
13	32-bit Source IP address				
17	32-bit Destination IP address				
	// IP Options (if any) //				
	// IP DATA //				

Layer 2 – IP Addresses

- IP version 4
 - 32-bit unsigned integer
 - written in decimal as a “dotted quad” - 128.138.202.19
- Historical IP address classes

Class	1st Byte	Net bits	
A	1 – 126	8	N.h.h.h
B	128 – 191	16	N.N.h.h
C	192 – 223	24	N.N.N.h
D	224 – 239	multicast	
E	240 – 254	research / reserved	

Network Classes

- 32-bit internet addresses originally divided simplistically
 - Class A – 8 net bits, 24 host bits – 16777214 hosts
 - Class B – 16 net bits, 16 host bits – 65534 hosts
 - Class C – 24 net bits, 8 host bits – 254 hosts
 - Why do we lose two addresses per network?
- Companies originally given a classed network
 - IP space ownership similar to old phone #s
 - Companies owned phone numbers
- Large problems with classed networks
 - IP space limitations, routing nightmare

IP Addresses – CIDR

- Classless Inter-Domain Routing
 - Solution to sucky 'classed' based splitting
 - Allowed for shorter routing tables on backbone routers
 - Net/Host boundary can be arbitrary
 - Not just on byte boundaries like the classed networks
 - Host portion still contiguous set of least-significant bits
 - Written as: <IP addr> / <net-bits>
 - 128.138.202.19 / 24
 - / **24** means 24 bits designate the NET portion of the address

CIDR – An Exercise in Base-2

- 128.138.202.0/24
- 'Net' or 'Network' bits
 - Number of bits most significant to least
 - Number of 'net' bits denoted by '/24'
- 'Host' bits
 - What's leftover from the net bits subtracted from 32
 - 32 total bits in an IP address
 - $32 - 24 == 8$ host bits $== 254$ hosts
 - Total number of allowed hosts:
 - $2 ^ (32 - \text{'net bits'}) - 2$

IP Addresses – Netmasks

- 32-bit quantity (same as IP address)
 - Consists of all ones (1) for the NET portion of the address
 - Necessary for CIDR to function
 - MASKS the HOST portion to find only the 'net' bits
- Example
 - Host: `csel.cs.colorado.edu`
 - IP: `128.138.202.19`
 - Subnet: `128.138.202.0/24`
 - Netmask: `255.255.255.0`

IP Addresses – Subnets

- A **subnet** is a range of IP addresses
 - Let's look at a new subnet:
 - Subnet: 128.138.242.0/23
 - the subnet has eight (9) bits for the HOST portion
 - a HOST portion of zeroes designates the NET itself
 - 128.138.242.0
 - a HOST portion of all ones (1) designates the broadcast address for the subnet
 - Subnet address: 128.138.242.0
 - Broadcast address: 128.138.243.255

IP Addresses – CS Example

- Consider 128.138.242.0 and 128.138.243.255 in binary
 - 1000 0000 . 1000 1010 . 1111 0010 . 0000 0000
 - 128 . 138 . 242 . 0
 - 1000 0000 . 1000 1010 . 1111 0011 . 1111 1111
 - 128 . 138 . 243 . 255
- A CIDR address of 128.138.242.0/23 means
 - the NET portion is 23 bits:
 - 1000 0000 . 1000 1010 . 1111 0010 . 0000 0000
 - and the HOST portion is 9 bits:
 - 1000 0000 . 1000 1010 . 1100 1011 . 0000 0000

IP Addresses – CS (cont)

- **/23 (9 HOST bits) subnet – 128.138.242.0/23**
 - Subnet: 128.138.242.0/23
 - Netmask: 255.255.254.0
 - 1111 1111 . 1111 1111 . 1111 1110 . 0000 0000
 - IP Broadcast: 128.138.243.255
- **Another /24 subnet – 128.138.237.0/24**
 - Engineering Wireless!
 - Subnet: 128.138.237.0/24
 - Netmask: 255.255.255.0
 - IP Broadcast: 128.138.237.255

Subnets within Subnets

- University of Colorado given 128.138.0.0/16
 - That exact address and CIDR mask used in internet routers
- CS department given some of this space
 - 128.138.242.0 – 128.138.243.255 (512 addresses)
 - subnetted into six / **26** networks (6 HOST bits, 64 addresses) and one / **25** network (7 HOST bits, 128 addresses)
 - One 128.138.242.0 / 23 route used by CU routers
 - They have no idea about our internal subnets
 - More efficient to have a single route

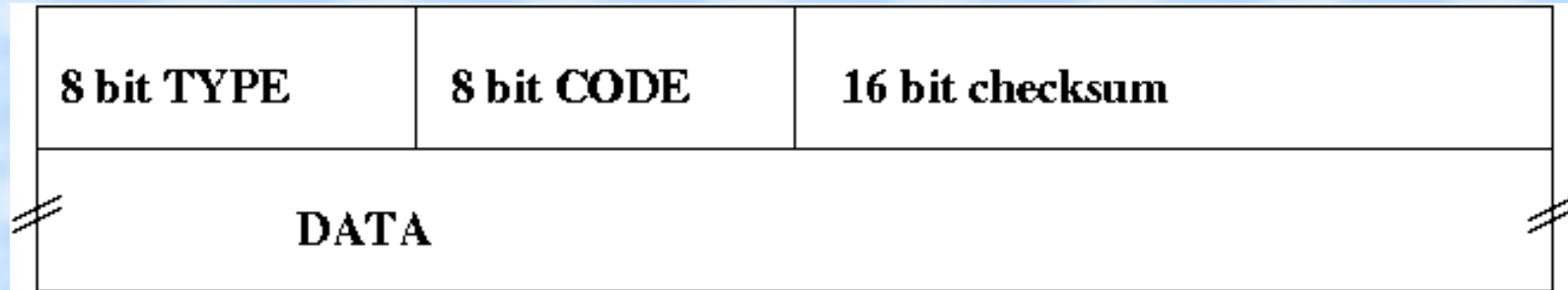
Byte Boundaries

- Base-2 byte boundaries **MUST** be observed
 - Can't arbitrarily split up subnets into smaller ones
 - Net and Host bits must be contiguous and static
- 192.168.1.0/25 (25 net bits, 7 host bits) yields:
 - 1100 0000 . 1010 1000 . 0000 0001 . 0000 0000
 - 192 . 168 . 1 . 0
 - 1000 0000 . 1000 1010 . 0000 0001 . **0111 1111**
 - 192 . 168 . 1 . 128
- This is your exercise in base-2
 - Hosts: $2^{(32 - 7)} - 2 == 126$

Byte Boundaries (cont)

- 192.168.1.0/25 (and /24) works
- 192.168.1.0/23 does NOT!
 - Violates base-2 byte boundaries
 - The last bit in the 'net' portion changes
- 10.0.2.0/23 works
- 10.0.2.0/22 does NOT!
 - /22 networks must have a 3rd octet divisible by 4
- How do you know what is valid and what isn't?
 - Practice and base-2 division

Network Layer (2) – ICMP



- Internet Control Message Protocol
 - Part of network layer but relies on IP for delivery
- Possible types are:
 - Echo reply (0), Dest Unreachable (3)
 - Redirect (5), Echo request (8)
- For some types, CODE gives more info
 - try pinging the CU gateway 128.138.240.1

The ifconfig Command

- View network interface parameters
 - output varies by OS
 - Use -a to see info about all interfaces
 - use netstat -i if -a doesn't work
 - Specify an interface to view that one only (ifconfig eth0)
- Configure network interface parameters
 - Typically:
 - `ifconfig <int> down`
 - `ifconfig <int> <ip> netmask <netmask> up`

Network Layer (2) – Summary

- Contains IP Addresses
 - Used for general internet routing
 - CIDR routing and byte boundaries
- Unreliable protocols at this layer
 - IP and ICMP
- Protocols listed in a file under /etc
 - /etc/protocols
 - Same values used in the 8-bit 'protocol' field

Transport Layer (3)

- Port numbers are 16-bit unsigned int values
 - Used to differentiate applications and services
 - IP address is the **host (device)**, Port is the **process**
 - Transport layer protocols must use both source and dest ports
 - Important services use well-known port numbers
 - Usually found within /etc/services
 - Very long list full of ports you've never heard of
 - Restricted ports (ports < 1024) require root
 - *Very* mild security

Transport Layer - UDP

- User Datagram Protocol
 - **Unreliable** and **stateless** like IP
 - Applications must write-in reliability if needed
 - Good for short, one-time things like
 - NTP
 - DNS queries
 - Streaming video

16 bit SOURCE port number	16 bit DESTINATION port number
16 bit length (incl hdr) in bytes	16 bit hdr chksm – unused

Traceroute

- Used to 'map' routes between you and the destination
- Traceroute sends arbitrary UDP packets to dest
 - Dest port is set to an unlikely value
 - IP time-to-live (TTL) field is incremented at each hop
 - ICMP time exceeded (11) error returned from routers
 - ICMP port unreachable (code 3) returned from dest

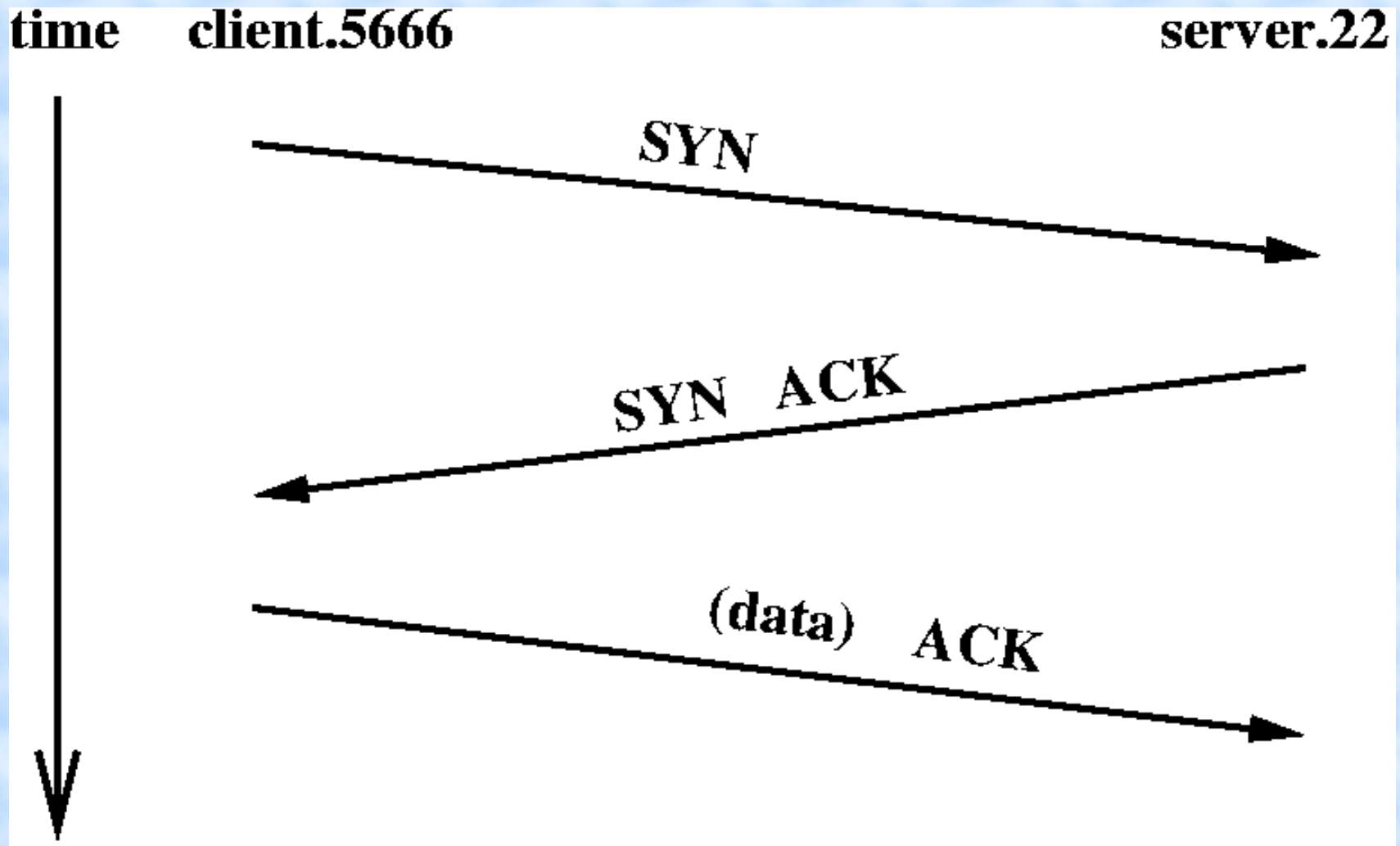
Transport Layer – TCP

- Transmission Control Protocol
 - Both reliable AND stateful
 - Reliability is done with timeout and re-transmission
 - State is defined by a source IP address, port-number pair and a destination IP address, port-number pair
 - Full Duplex means data flows in both directions (accomplished using piggyback ACKs)
 - Streaming (bulk) or interactive (TTY) data flow

Transport Layer – TCP Header

16 bit Source Port		16 bit Destination Port	
32 bit Sequence Number			
32 bit Acknowledgement Number			
4 bit header length	6 bit reserved	6 bit flags	16 bit Window
16 bit Checksum		16 bit Urgent Pointer	
Options (if any)			

Transport Layer – TCP Handshake



Transport Layer – TCP (cont)

- Other features of TCP
 - MSS – Maximum Segment Size (65507 bytes)
 - similar to MTU
 - Delayed ACKs (piggybacking)
 - Sending acks with data
 - Sliding window
 - More efficiency in sending data
 - Network congestion avoidance
 - and others

TCPDump

- Used to display TCP/IP packet headers
 - usually as directly received from an active network via an interface in PROMISCUOUS mode
 - Packets are specified by specifying a boolean expression
- Useful options:
 - `-i <interface>` listen on a specific interface
 - `-e` give link-layer header also
 - `-n` give IP addresses (no DNS)
 - `-t` don't give timestamps
 - `-v` give more header fields (verbose)

TCPDump (cont)

- TCPDump boolean expressions
 - If none given, all packets are displayed
 - Expressions consist of primitives combined with **and, or, not**
 - large variety of primitives (see the manpage)
 - parentheses may be used (but must be escaped)
 - Example primitives:
 - `src host csel`
 - `port ftp and not port ftp-data`
 - `dst port http and host csel`

Ethereal (Wireshark)

- Graphical packet sniffing tool
 - same behavior as tcpdump, but somewhat easier to see
 - decodes and splits packets into readable formats
 - Link, Network, Transport, and Application layers
- Reads capture files from tcpdump
 - Also uses the same boolean expressions

Netstat

- Has many uses
 - View the system's routing table
 - View active or listening TCP and UDP services
 - Other network stats
 - Implementation varies by OS
- Some options
 - `-i` Show stats for all interfaces
 - `-a` Show server socket state too
 - `-n` Don't do IP->name (DNS) lookups
 - `-r` Show system's routing table
 - `-vaten` I find this one most useful