

# Unix System Administration

**Chris Schenk**

Lecture 10 – Thursday Feb 14

CSCI 4113, Spring 2008

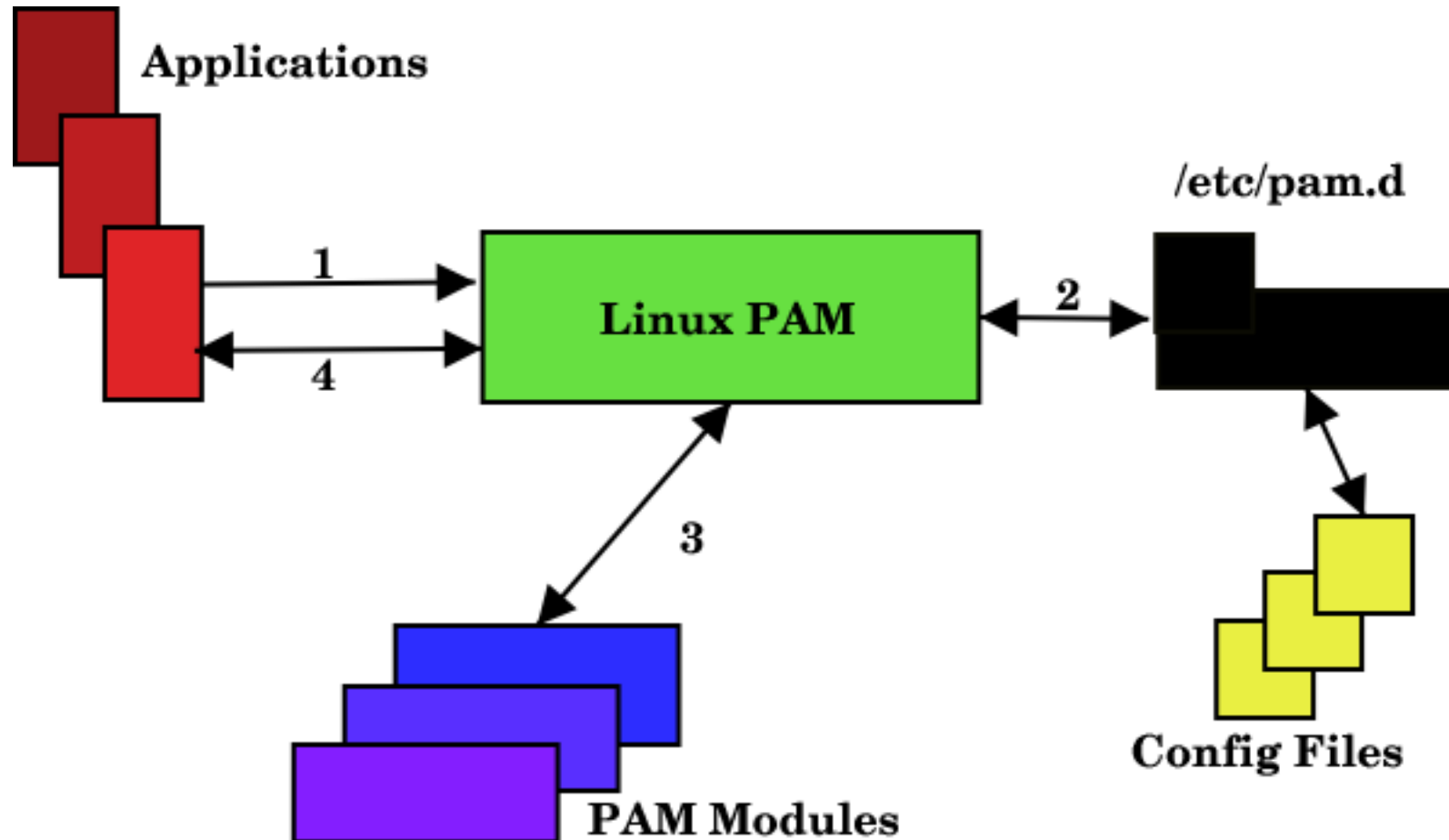
# Pluggable Authentication Modules (PAM)

- Provides a generic interface to multiple auth backends
  - /etc/passwd, ldap, NIS, yp, etc
- Four module types in PAM
  - auth: validating credentials (username/password)
  - account: provides account management for non-auth things (shell, homedir, etc)
  - session: creates the environment for the user
  - password: provides means to change passwords
- All config files exist under /etc/pam.d

# PAM – Modules

- Modules belong to one or more of four types
  - pam\_unix belongs to **all** four module types
  - pam\_nologin belongs to only **account** and **auth**
- Modules take parameters for their execution
  - Different for each module, check the docs!
- The 'control flag' is return value of the module
  - Decides whether or not PAM returns successful
  - required, sufficient, optional, include, requisite

# PAM – Simple Flowchart



# PAM – Other Notes

- PAM provides many modules beyond 'auth'
  - pam\_access: access control lists for groups
  - pam\_cracklib: checks password strength
  - and many more:  
<http://www.kernel.org/pub/linux/libs/pam/>
- Also have ability to jump to or include other files
  - Allow for richer authentication and management schemes
- Programs must have written in support for PAM
  - Must have code that uses PAM's interface

# Time

- Marches infinitely?
  - Not with a 32-bit counter!
- Unix time standard is with a 32-bit counter
  - Starting at the “Unix Epoch”
    - Midnight UTC January 1, 1970
  - Counts number of seconds since the epoch
- Another Y2K problem?
  - 03:14:08 UTC Jan 19, 2038 – 32-bit overflow

# Time Keeps On Slippin'

- Network Time Protocol (NTP)
  - Allows for synchronization of clocks
  - Necessary for certain things to function
  - sudo, nfs, and many more
- Protocol written to deal with latency
  - Network latency when querying a time server!
  - Also tries to learn the drift of your clock
    - So it can adjust without as many queries

# CRON – Periodic Scheduler

- Utility to perform tasks at certain intervals
- Everything starts with a crontab
  - Global crontab for root under `/etc/crontab`
  - Users can install a personal crontab with `crontab -e`
- Time granularity is only down to the minute
  - minutes/hours/days(month)/month/days(week)
- Example: run 8:00pm on weekdays in October
  - `0 20 * 10 1-5 /usr/local/bin/mysweetscript`
- `man 5 crontab` for more info!

# 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

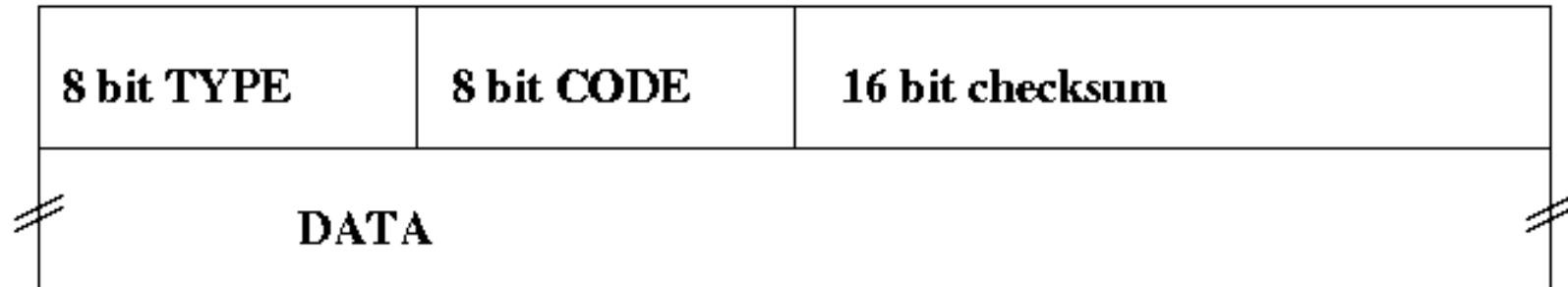
# Byte Boundaries

- 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 . 0**111 1111**
  - 192 . 168 . 1 . 127
- 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 3<sup>rd</sup> 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`

# The route Command

- Controls the route table on your machine
  - All routing between hosts occurs at layer 2!
  - Most hosts only have a single default route (gw)
- Your machine can act as a router
  - Many different ethernet cards with different subnets
  - Need to know where to send packets
- 'route' has different behavior on different \*nix
  - Read those man pages or Google for help
- Can add and delete routes as needed
  - Again, uncommon to do with a normal host

# 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 (lots of packets, but each small)

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

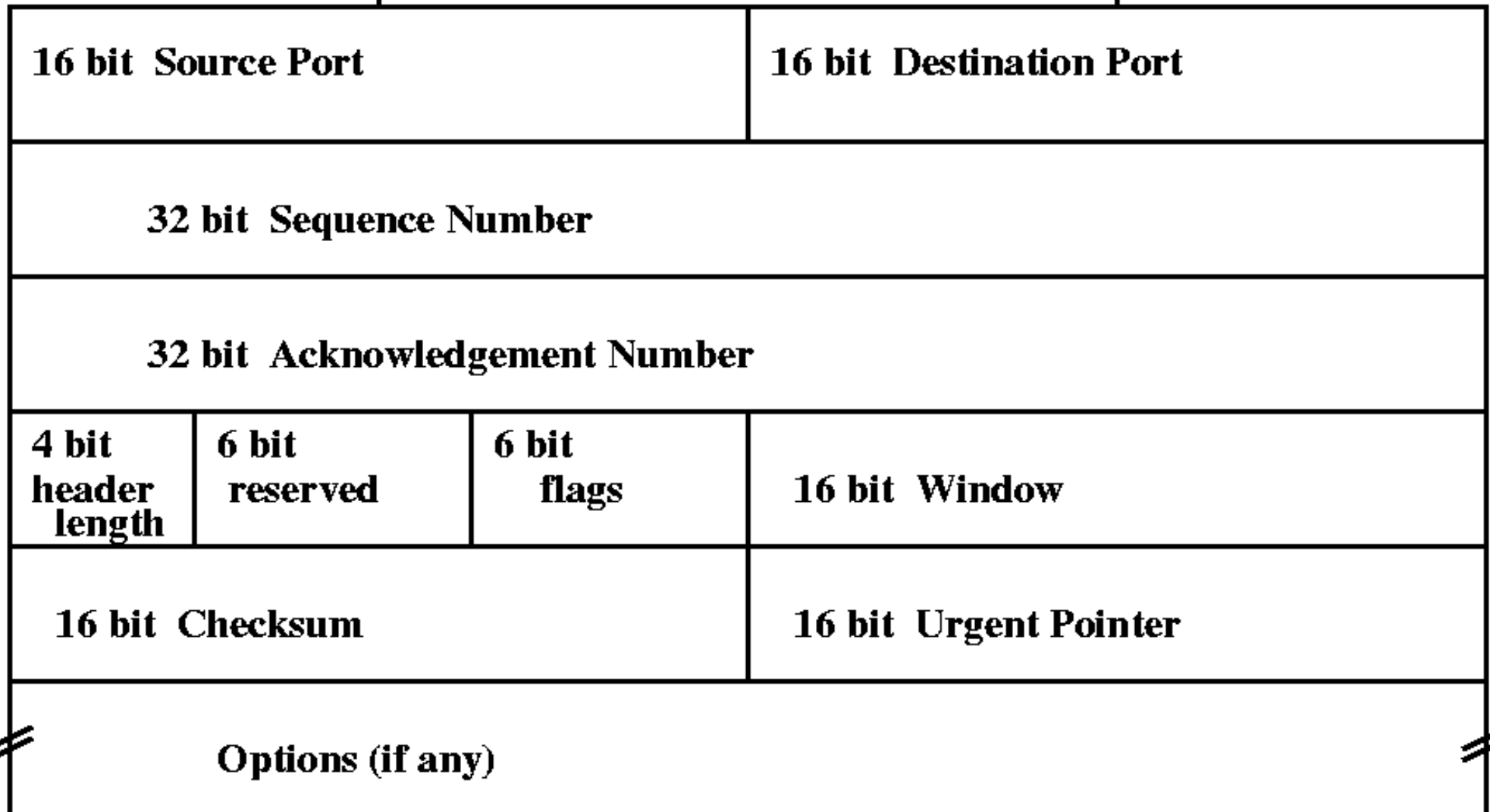
# Traceroute

- 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

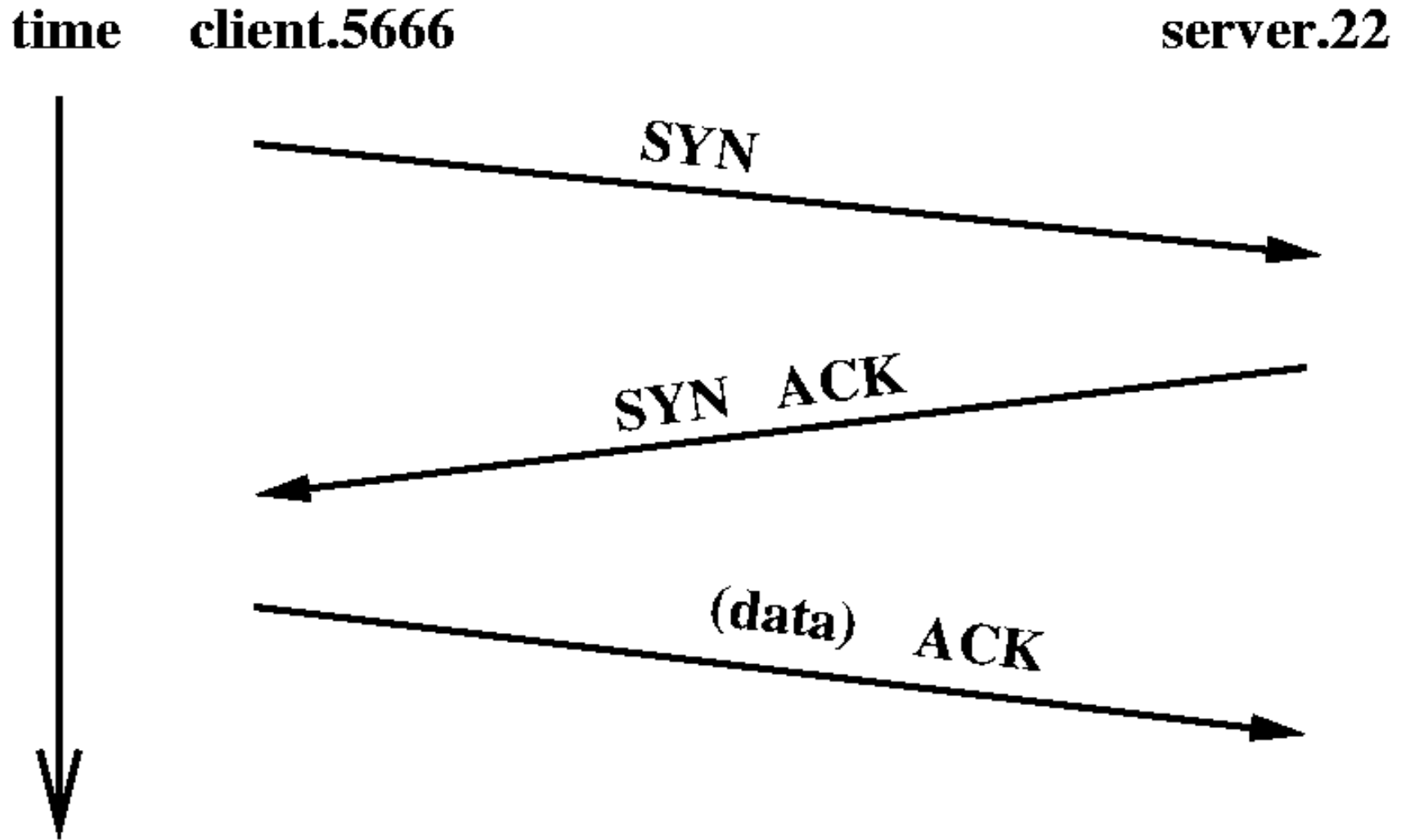
# Transport Layer (3) – TCP

- Transmission Control Protocol
  - Both reliable AND stateful
  - Reliability is done with timeout and re-transmission
  - A connection 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)

# Transport Layer – TCP Header



# Transport Layer – TCP Handshake



# TCP vs UDP Port Closed

- How does a host determine if a port is closed?
  - Behavior differs between TCP and UDP
- TCP SYN received on server's closed port
  - TCP RST sent back to client
    - A layer 3 response
- UDP received on server's closed port
  - ICMP host/port unreachable sent back to client
    - A Layer 2 response

# Transport Layer (3) – 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

# Ephemeral Ports

- A connection requires the endpoint IP-Port pair
  - Connect to google.com at port 80
- Client browser doesn't have to explicitly bind to a local port
  - This is where ephemeral ports come in
- Ephemeral ports range varies between systems
  - Linux: `/proc/sys/net/ipv4/ip_local_port_range`
    - 32768-61000
  - FreeBSD: `sysctl -a | grep net.inet.ip.portrange`
    - 1024-5000

# Basic Unix Network Connectivity

- Three things to get \*nix up and running by hand
  - Obviously as root, works under most unix systems
- Set your ip address and network with 'ifconfig'
  - `Ifconfig eth0 128.138.202.145 netmask 255.255.255.0`
- Add a default route with 'route'
  - `route add default gw 128.138.202.1`
- Add in DNS nameservers to `/etc/resolv.conf`
  - `nameserver 128.138.202.8`
  - `nameserver 128.138.130.30`
  - `search cs.colorado.edu colorado.edu`

# 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)

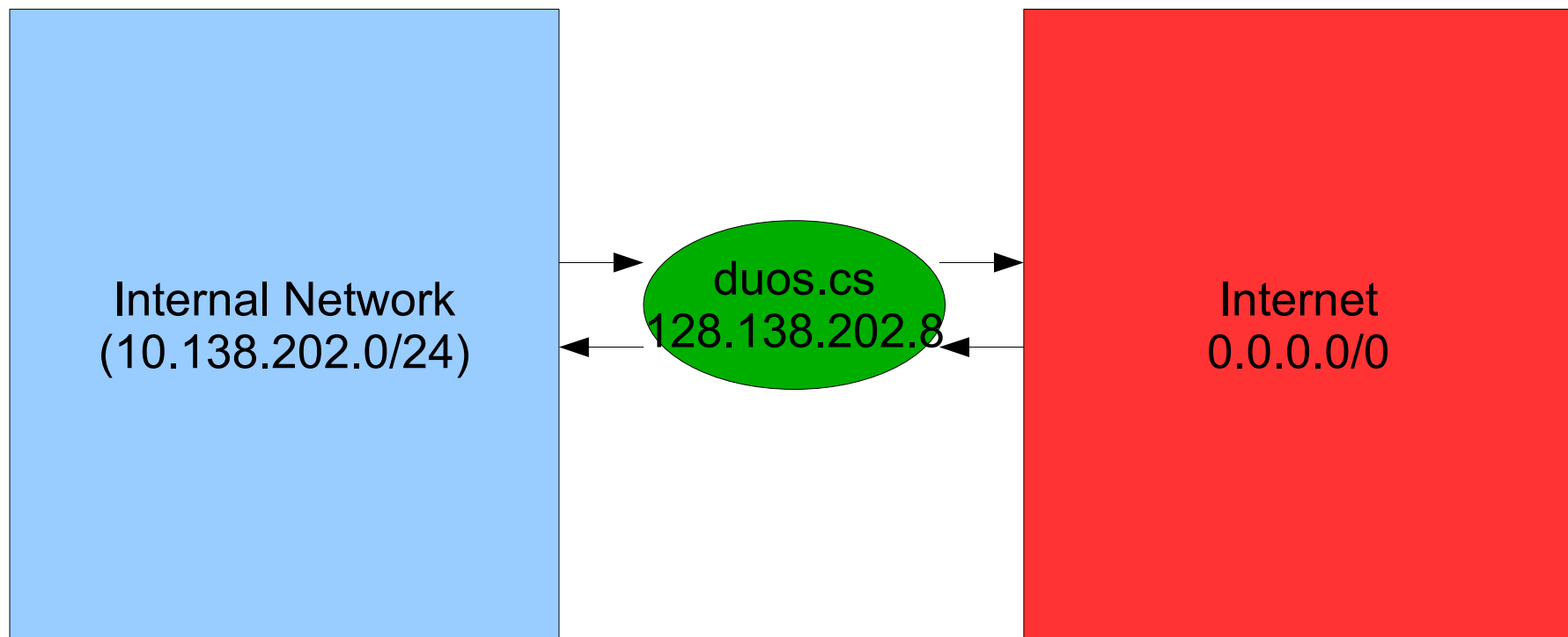
- Graphical packet sniffing tool
  - same behavior as tcpdump, but somewhat easier to see
  - decodes and splits packets into readable formats
    - Link, Network, Transport, and Application layers
- Reads capture files from tcpdump
  - Also uses the same boolean expressions

# 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
  - `-lt` Show tcp listen sockets
  - `-lu` Show udp listen sockets

# Network Address Translation (NAT)

- NAT provides a way to mask machines on a network
  - Single IP address exposed for multiple hosts



# Network Address Translation (cont)

- The SOURCE IP address and port is rewritten to the gateway's IP address and port
  - Kyle is 10.138.202.72 and ephemeral port 37019
  - Rewritten to 128.138.202.8 and ephemeral port 41000
- The gateway machine waits for a response from server
  - Looks at the rewritten ephemeral response to know how to rewrite to the original IP and port of Kyle
  - So any incoming packets for ephemeral port 41000 are rewritten to IP 10.138.202.72 and port 37019 for Kyle