CSE 398: System Administration

- TCP/IP Networking
  - We will cover just the practical issues
  - Highly recommend taking a networking course

- What is TCP/IP?
- How was TCP/IP developed?

 ©2004 Brian D. Davison

TCP/IP

- Most common networking protocol suite
- Foundation of the Internet
  - 675M+ users online worldwide (Sep 2003)
  - 171M+ hosts online (Jan 2003 number)
- Network applications typically use one of two transport protocols:
  - TCP – Transmission Control Protocol
  - UDP – User Datagram Protocol
- All traffic carried by IP – Internet Protocol

 ©2004 Brian D. Davison

Protocols

- IP
  - Packet-oriented (routers don't care what is in packets or what came before)
- TCP
  - Connection-oriented, two-way, reliable, in-order transport of stream of bytes
  - Congestion control – slow down when congestion is noticed, speed up when resources available
  - Flow control – don't overwhelm receiver
- UDP
  - Unreliable transport of packets

 ©2004 Brian D. Davison

Network layers

- Application layer – end-user applications
- Transport layer – delivery of data between applications
- Network layer – basic communication, addressing, and routing
- Link layer – network hardware and device drivers
- Physical layer – cable or physical medium

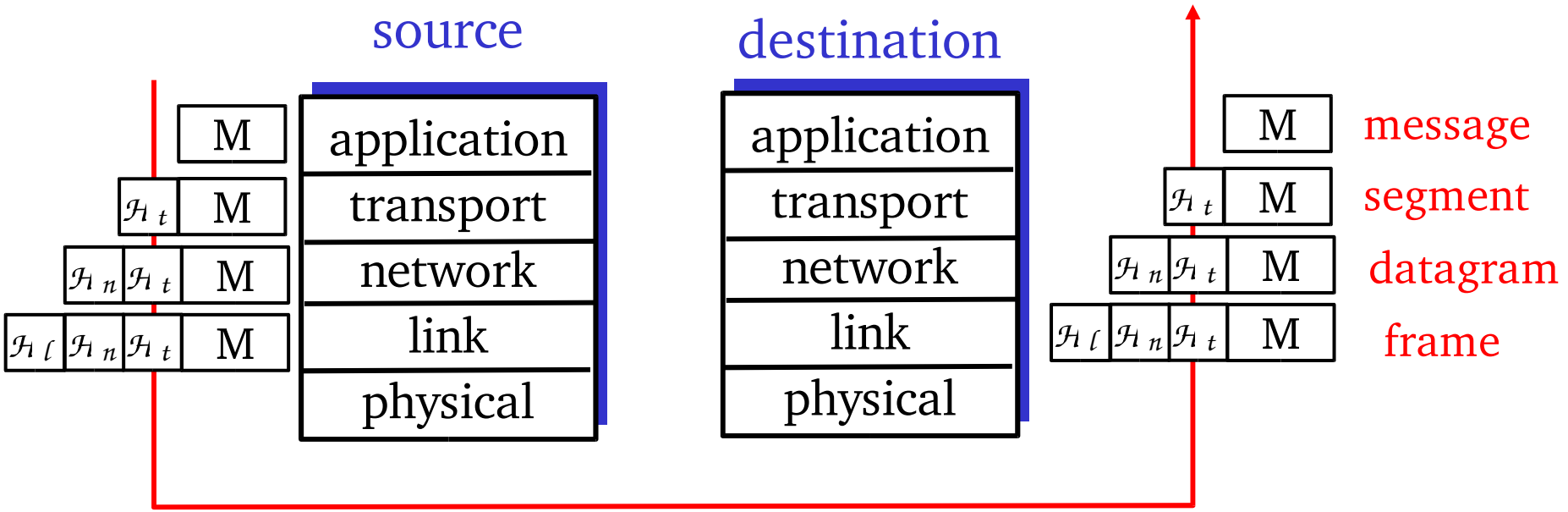
 ©2004 Brian D. Davison

Encapsulation

- As data is sent downward through the stack, it is encapsulated with layer-specific headers
- E.g.
  - App sends 100 bytes
  - UDP adds 8 bytes of header
  - IP adds 20 bytes
  - Ethernet adds 18 bytes

 ©2004 Brian D. Davison

Layers + Encapsulation



 ©2004 Brian D. Davison

Addressing

- Different layers use different addressing
  - App. layer (usu.) allows people to use hostnames
  - IP (network) layer requires IP addresses
  - Link layer requires MAC addresses
    - e.g., Ethernet (48 bits)
      - First 3 bytes are manufacturer ID
      - Last 3 bytes are serial number
- Ports identify process or service on a host
  - List of well-known ports in /etc/services
  - Ports <= 1024 are priviledged ports

 ©2004 Brian D. Davison

Address types

- IP layer and link layer have multiple address types
  - Unicast – single host (network interface)
  - Broadcast – addresses that include all hosts on a particular network
    - All bits in host part of address are ones
  - Multicast – addresses that identify a group of hosts
    - IP addresses with first byte in 224-239

 ©2004 Brian D. Davison

IP Addresses

- IPv4 address has four bytes
  - Split into network and host portions
  - Internet originally used classes of IP addresses

Class	1st byte	Format	Comments
A	1-126	N.H.H.H.	Very early networks, DoD
B	128-191	N.N.H.H.	Large sites, usually subnetted
C	192-223	N.N.N.H.	Smaller sites
D	224-239		Multicast addresses
E	240-255		Experimental

  - www.lehigh.edu = 128.180.2.14
    - Class B (128); host portion is 180.2.14

 ©2004 Brian D. Davison

Subnetting

- Individual networks are often **much** smaller than the class sizes
- Subnetting permits breaking up an allocation into multiple smaller networks
- Lehigh breaks up its Class B into many smaller networks, such as the old EECS nets
  - 128.180.5.*, 128.180.98.*, 128.180.14.*
  - Each can be broken down further

 ©2004 Brian D. Davison

Subnetting Example

- 128.180 under class-full addressing is a Class-B with 65,534 addresses
- Subnetting extends the network address into host portion
- We specify a subnet 128.180.98
  - Using explicit subnet mask 255.255.255.0
  - Alternatively, with network bits specified explicitly
    - 128.180.98.0/24
  - Can also break on non-byte boundaries
    - 128.180.98.128/25
    - 128.180.120.0/22

 ©2004 Brian D. Davison

CIDR

- Classless Inter-Domain Routing
  - Allows for shorter network address than class-specified – obsoletes network classes
  - Requires length field, e.g., 128.180.0.0/16
  - Aggregates smaller networks into single larger one
    - 192.200.254.0 + 192.200.255.0 = 192.200.254.0/23
  - Can now allocate portions of class A and B addresses
  - Aggregated networks reduces routing table growth

 ©2004 Brian D. Davison

Address Shortage

- Before CIDR, concern for enough addresses
  - Class Bs would be gone by 1995
  - Router tables were exploding (growing beyond router capacities)
- CIDR + NAT + name-based virtual hosting greatly slowed down IP allocations
- IPv6 will solve this (16 byte addresses!)

 ©2004 Brian D. Davison

NAT

- Network Address Translation
  - Router intercepts packets, replaces internal network addresses and ports with externally visible addresses and ports
  - Maintains mapping so that external packets are directed to the right internal host
  - Typically uses a single public IP address, many ports, but can (in theory) map arbitrary hosts/ports
  - Capability built into many (cheap) routers, Linux

 ©2004 Brian D. Davison

Private Addresses

- While a NAT can protect your internal addresses from being visible in IP headers, it isn't perfect
  - Some apps will encode addresses in data
  - What if you really want to connect to the external host with an IP address same as an internal host?
- Most use private address space (unroutable)

IP Class	From	To	CIDR Range
A	10.0.0.0	10.255.255.255	10.0.0.0/8
B	172.16.0.0	172.31.255.255	172.16.0.0/12
C	192.168.0.0	192.168.255.255	192.168.0.0/16

 ©2004 Brian D. Davison

Routing

- Given a packet, on which of multiple network interfaces should it be sent?
- UNIX kernel keeps routing table (netstat -rn)
  - Sample from dual-homed host

```
Kernel IP routing table
Destination     Gateway         Genmask         Iface
128.180.98.128  0.0.0.0         255.255.255.128 eth0
192.168.0.0     0.0.0.0         255.255.0.0     eth1
127.0.0.0       0.0.0.0         255.0.0.0       lo
0.0.0.0         128.180.98.248  0.0.0.0         eth0
```

 ©2004 Brian D. Davison

Routing continued

- Routing is static for most systems
  - Established when network card configured
- Additional static routes can be added using the `route` command
- Dynamic routing can be managed using `routed` or `gated`

 ©2004 Brian D. Davison

ARP: Address Resolution Protocol

- Once the routing of a packet has been determined, it must be transmitted to the next gateway or host on the local network
- LAN transmissions use LAN addresses
- ARP is used to discover the hardware address of the target IP address
- ARP sends a LAN broadcast asking who has the desired IP address; the owner responds with a unicast message with answer
  - Results cached in a table (also collected via snooping)

 ©2004 Brian D. Davison

Sample ARP table

```
% arp -a

Net to Media Table: IPv4
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
eri0   proxima              255.255.255.255       08:00:20:86:2f:72
eri0   genie-plab           255.255.255.255       08:00:20:80:83:c4
eri0   128.180.123.254      255.255.255.255       00:d0:02:13:44:00
eri0   ariel                255.255.255.255 SP    00:03:ba:27:95:90
eri0   dhcp123-114          255.255.255.255       00:0c:6e:94:15:de
eri0   dolphin              255.255.255.255       00:30:48:70:93:d1
eri0   benson-plab          255.255.255.255       08:00:20:73:02:be
eri0   diogenes2            255.255.255.255       00:10:4b:d1:9b:8a
eri0   pl333-05             255.255.255.255       00:0c:76:1a:bf:97
eri0   dhcp123-139          255.255.255.255       00:03:47:d3:24:68
eri0   dhcp123-130          255.255.255.255       00:0c:f1:8c:99:e6
eri0   dhcp123-163          255.255.255.255       00:10:dc:e2:d4:cc
eri0   BASE-ADDRESS.MCAST.NET 240.0.0.0     SM    01:00:5e:00:00:00
```

 ©2004 Brian D. Davison

Network Configuration

- Adding a machine to a LAN
  - Assign unique IP address and hostname (per interface)
  - Set up host to configure network interfaces at boot time
  - Set up default route
  - Point to DNS name server
- Files
  - /etc/sysconfig/network-scripts/ifcfg-eth0
  - Hostname, default route, IP address, netmask, broadcast
- DHCP could do all of this automatically

 ©2004 Brian D. Davison

Mapping names to IP addresses

- Three choices: /etc/hosts, NIS, DNS
- Simplest: /etc/hosts

```
% more /etc/hosts
#
# Internet host table
#
127.0.0.1       localhost
128.180.120.15  proxima
128.180.120.9   mailhost
128.180.120.103 ariel   # Added by DHCP
```

- Works when NIS or DNS is broken
  - e.g., at boot time

 ©2004 Brian D. Davison

ifconfig

- Configure network interfaces with ifconfig
  - ifconfig eth0 128.138.240.1 netmask 255.255.255.0 up
  - shows configuration, e.g., for Suns:

```
ariel% ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4> mtu 1500 index 2
        inet 128.180.120.103 netmask fffffc00 broadcast 128.180.123.255
lo0: flags=2000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
eri0: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        inet6 fe80::203:baff:fe27:9590/10
```

 ©2004 Brian D. Davison

ifconfig on linux

```
wume4% /sbin/ifconfig -a
eth0  Link encap:Ethernet  HWaddr 00:40:F4:34:C9:9A
      inet addr:128.180.5.20  Bcast:128.180.5.255  Mask:255.255.255.0
      UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:11986063 errors:355 dropped:0 overruns:0 frame:0
      TX packets:1011576 errors:11 dropped:0 overruns:0 carrier:22
      collisions:227547 txqueuelen:100
      RX bytes:2623049414 (2501.5 Mb)  TX bytes:215019824 (205.0 Mb)
      Interrupt:10 Base address:0xd000

lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:843156 errors:0 dropped:0 overruns:0 frame:0
      TX packets:843156 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:117522440 (112.0 Mb)  TX bytes:117522440 (112.0 Mb)
```

 ©2004 Brian D. Davison

Red Hat configuration files

- /etc/sysconfig/network
  - hostname, default route
- /etc/sysconfig/static-routes
  - static routes
- /etc/sysconfig/network-scripts/ifcfg-XXXX
  - IP address, netmask, broadcast address per interface
  - e.g., eth0, eth1, lo
- Use ifup and ifdown to change interface status, or use /etc/rc.d/inint.d/network

 ©2004 Brian D. Davison

DHCP

- Dynamic Host Configuration Protocol
- Clients **lease** network config from server
  - IP addresses and netmasks
  - Gateways (default routes)
  - DNS name servers
  - Syslog hosts
  - X font servers, proxy servers, NTP servers
  - and more

 ©2004 Brian D. Davison

How DHCP works

- Client broadcasts a “Who am I?” message
- Local DHCP server responds with network configuration lease
- When lease is half over, client renews the lease
  - DHCP server must track lease info (persist through server reboots, etc.)
- DHCP used on most CSE/ECE hosts

 ©2004 Brian D. Davison

dhcpd configuration

```
#dhcpd.conf
#
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.51 192.168.1.60;
   option broadcast-address 192.168.1.255;
   option routers gw.synack.net;
}
subnet 209.180.251.0 netmask 255.255.255.0 {
}
host gandalf {
   hardware ethernet 08:00:07:12:34:56;
   fixed-address gandalf.synack.net;
}
```

 ©2004 Brian D. Davison

Networking hardware

- Ethernet is the core of most networks
  - 10 Mbit 10Base2, 10BaseT
  - 100 Mbit 100BaseTX
  - 1 Gbit 1000BaseT
- Many competing LAN technologies
  - ATM, Token Ring, FDDI
- Wireless
  - 802.11b/a/g

 ©2004 Brian D. Davison

Connecting ethernets

- Hubs/repeaters (physical layer)
  - Retimes and reconstitutes Ethernet frames to all ports
  - Single collision domain
- Switch (link layer)
  - Receives, buffers, and retransmits packets
  - Separate collision domains
  - Learns locations of MAC addresses, selectively forwards frames
- Router (IP/network layer)
  - Connects separate ethernet networks
  - Can connect different LAN/WAN technologies

 ©2004 Brian D. Davison

Connecting to the Internet

- How can you connect your network to the Internet?

 ©2004 Brian D. Davison

Connecting to the Internet

- How can you connect your network to the Internet?
  - Dialup
  - Wireless (cellular, satellite, point-to-point)
  - xDSL
  - Cable
  - IDSN
  - Frame relay
  - T1, T3, OC3, etc.

 ©2004 Brian D. Davison

Remote access

- Connecting users to the organization
  - Check email, access data when traveling
  - Work from home
  - Remote facilities (e.g., stationed at customer site) but need access organization network regularly
- Different needs, different support requirements
  - Establish an SLA
  - Worry about security, performance, costs

 ©2004 Brian D. Davison

Network debugging

- Questions to ask
  - Do you have physical connectivity and a link light?
  - Is your interface configured properly?
  - Is DNS configured properly?
  - Do your ARP tables show other hosts?
  - Can you ping the localhost address (127.0.0.1)?
  - Can you ping other local hosts by IP address?
  - Can you ping other local hosts by hostname?
  - Can you ping hosts on another network?
  - Do high-level commands like telnet and ssh work?

 ©2004 Brian D. Davison

Network tools

- ping
- traceroute
- netstat
- arp (saw earlier)
- tcpdump/ethereal

 ©2004 Brian D. Davison

ping

- If ping works, networking between hosts is likely to be working
- It does not test availability of services
- Some firewalls filter ICMP messages

```
# ping www.lehigh.edu
PING www.lehigh.edu (128.180.2.14) from 128.180.98.216: 56(84) bytes
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=1 time=0.859 ms
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=2 time=0.685 ms
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=3 time=0.648 ms
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=4 time=0.751 ms
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=5 time=0.667 ms
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=6 time=1.00 ms
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=7 time=0.725 ms
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=8 time=0.655 ms
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=9 time=0.663 ms
64 bytes from ws1.CC.Lehigh.EDU (128.180.2.14): icmp_seq=10 time=0.724 ms

--- www.lehigh.edu ping statistics ---
10 packets transmitted, 10 received, 0% loss, time 9091ms
rtt min/avg/max/mdev = 0.648/0.738/1.006/0.109 ms
```

 ©2004 Brian D. Davison

traceroute

- Finds the sequence of gateways traveled
- Works by increasing the TTL of the packet sent
- traceroute -n skips DNS

```
ariel% traceroute www.princeton.edu
traceroute to hulk.princeton.edu (128.112.128.15), 30 hops max, 40 byte
packets
 1 128.180.123.254 (128.180.123.254) 3.406ms  0.940ms 0.676ms
 2 ewfmB-GBE-A.CC.Lehigh.EDU (128.180.18.4) 0.407ms 0.465ms 0.640ms
 3 ewfmd-ewfmb.CC.Lehigh.EDU (128.180.128.82) 0.952ms 0.649ms 0.830ms
 4 ewfmc-ewfmd.CC.Lehigh.EDU (128.180.128.89) 1.005ms 1.687ms 1.464ms
 5 local.lehigh1.magpi.net (198.32.42.145) 5.148ms 4.810ms 4.721ms
 6 phl-02-08.backbone.magpi.net (198.32.42.197) 4.277ms 4.895ms 4.362ms
 7 remote.princeton.magpi.net (198.32.42.66) 33.474ms 20.958ms 20.399ms
 8 gigagate1.Princeton.EDU (128.112.12.21) 25.451ms 9.580ms 20.908ms
 9 hulk.Princeton.EDU (128.112.128.15) 11.067ms * 34.043ms
```

 ©2004 Brian D. Davison

netstat

- Tons o' network statistics

```
# netstat | more
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address     Foreign Address          State
tcp        0      0 someserver:www    td9091978.adsl.ter:1819 ESTABLISHED
tcp        0      0 someserver:www    td9091978.adsl.ter:1817 TIME_WAIT
tcp        0      0 someserver:www    unknown.servercen:53522 TIME_WAIT
tcp        0      0 someserver:www    lj1157.inktomisea:49477 TIME_WAIT
tcp        0      1 someserver:1248   218.15.192.166:smtp     SYN_SENT
tcp        0      0 someserver:www    cable200-75-67-206:3307 FIN_WAIT2
tcp        0      0 someserver:1224   mx02.osn.de:smtp        ESTABLISHED
tcp        0      0 localhos:x11-ssh-offset localhost:57893   ESTABLISHED
tcp        0   4209 localhost:6023    localhost:34263         FIN_WAIT1
```

- Can also show
  - interface configurations, routing tables, counter values

 ©2004 Brian D. Davison

Packet sniffers

- Show you what is really on the network
- Not as useful in a switched environment
- Examples: tcpdump, ethereal

```
# tcpdump
tcpdump: listening on eth0
19:05:08 220.168.28.55.http > wume1.cse.lehigh.edu.64207: P
1001579411:1001579684(273) ack 3591949882 win 65300 <nop,nop,timestamp
64538628 1686086162> (DF)
19:05:08 wume1.cse.lehigh.edu.36560 > G.ROOT-SERVERS.NET.domain:  7131
[1au] PTR? 55.28.168.220.in-addr.arpa. (55) (DF)
19:05:08 G.ROOT-SERVERS.NET.domain > wume1.cse.lehigh.edu.36560:
7131-% 0/4/2 (164) (DF)
19:05:08 wume1.cse.lehigh.edu.36560 > l.gtld-servers.net.domain:
24184 A? NS1.APNIC.NET. (31) (DF)
19:05:08 wume1.cse.lehigh.edu.36560 > ns.ripe.net.domain:  12092 [1au]
PTR? 55.28.168.220.in-addr.arpa. (55) (DF)
```

 ©2004 Brian D. Davison