CSE 265: System and Network Administration

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

- What is TCP/IP?
- Layers, addresses, NAT
- Protocols: ARP, DHCP
TCP/IP

• Most common networking protocol suite
• Foundation of the Internet
  – 1.3B+ users online worldwide (Dec 2007)
  – 541M+ hosts online (Jan 2008 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
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 but quick/easy transport of individual packets
IP 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
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
## Layers + Encapsulation

<table>
<thead>
<tr>
<th>Layer</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>L1</td>
<td>L1</td>
</tr>
<tr>
<td>Link</td>
<td>L2</td>
<td>L2</td>
</tr>
<tr>
<td>Network</td>
<td>L3</td>
<td>L3</td>
</tr>
<tr>
<td>Transport</td>
<td>L4</td>
<td>L4</td>
</tr>
<tr>
<td>Application</td>
<td>L5</td>
<td>L5</td>
</tr>
</tbody>
</table>

- Source: **M**
- Destination: **H**
- Message: **H**
- Segment: **H**
- Datagram: **H**
- Frame: **H**

The diagram illustrates the encapsulation process where each layer adds its header (H) to the message (M) as it moves from the source to the destination.
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 privileged ports
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
IP Addresses

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

<table>
<thead>
<tr>
<th>Class</th>
<th>1st byte</th>
<th>Format</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>128-191</td>
<td>N.N.H.H.</td>
<td>Large sites, usually subnetted</td>
</tr>
<tr>
<td>C</td>
<td>192-223</td>
<td>N.N.N.H.</td>
<td>Smaller sites</td>
</tr>
<tr>
<td>D</td>
<td>224-239</td>
<td>N.N.N.N.H.</td>
<td>Multicast addresses</td>
</tr>
<tr>
<td>E</td>
<td>240-255</td>
<td></td>
<td>Experimental</td>
</tr>
</tbody>
</table>

- www.lehigh.edu = 128.180.2.57
  - Class B (128.180); host portion is .2.57
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
  - Each can be broken down further
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
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
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!)
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
NAT: Network Address Translation

All datagrams leaving local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)
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)

<table>
<thead>
<tr>
<th>IP Class</th>
<th>From</th>
<th>To</th>
<th>CIDR Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10.0.0.0</td>
<td>10.255.255.255</td>
<td>10.0.0.0/8</td>
</tr>
<tr>
<td>B</td>
<td>172.16.0.0</td>
<td>172.31.255.255</td>
<td>172.16.0.0/12</td>
</tr>
<tr>
<td>C</td>
<td>192.168.0.0</td>
<td>192.168.255.255</td>
<td>192.168.0.0/16</td>
</tr>
</tbody>
</table>
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).
Sample ARP table

% /sbin/arp -a

davison.cse.lehigh.edu (128.180.121.225) at 00:11:43:A0:0F:D8 [ether] on eth0
wume2.cse.lehigh.edu (128.180.121.222) at 00:08:54:1E:44:D4 [ether] on eth0
pan.cse.lehigh.edu (128.180.120.90) at 00:14:4F:0F:9C:1A [ether] on eth0
wume1.cse.lehigh.edu (128.180.121.221) at 00:08:54:1E:44:D0 [ether] on eth0
chiron.cse.lehigh.edu (128.180.120.87) at 00:14:4F:21:44:D8 [ether] on eth0
xena.cse.lehigh.edu (128.180.120.86) at 00:14:4F:21:52:E0 [ether] on eth0
hydra.cse.lehigh.edu (128.180.120.89) at 00:14:4F:21:53:F2 [ether] on eth0
kato.eecs.lehigh.edu (128.180.120.6) at 08:00:20:C4:20:08 [ether] on eth0
noon.cse.lehigh.edu (128.180.121.219) at 00:0F:1F:F9:C1:68 [ether] on eth0
wume-lab2.cse.lehigh.edu (128.180.122.153) at 00:18:8B:24:5A:F4 [ether] on eth0
lu-gw.eecs.lehigh.edu (128.180.123.254) at 00:00:0C:07:AC:00 [ether] on eth0
nix.cse.lehigh.edu (128.180.120.88) at 00:14:4F:21:44:C4 [ether] on eth0
ceres.cse.lehigh.edu (128.180.120.91) at 00:14:4F:23:F9:80 [ether] on eth0
rosie.eecs.lehigh.edu (128.180.120.4) at 08:00:20:B1:FC:F3 [ether] on eth0
wume-lab1.cse.lehigh.edu (128.180.122.152) at 00:18:8B:24:5D:E2 [ether] on eth0
morning.cse.lehigh.edu (128.180.120.43) at 00:C0:9F:38:CD:51 [ether] on eth0
wume-lab6.cse.lehigh.edu (128.180.122.157) at 00:0A:E6:5D:48:03 [ether] on eth0
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
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
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=2000841<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
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)
RH/Fedora 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/init.d/network`
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
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 almost all hosts at Lehigh
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;
}