Announcements

• Midterm II
  – March 21st (One week from today)
  – In class
  – Same format as last time
  – Will cover all material since Midterm I

• Talk: "Analyzing Intrusions Using Operating System Level Information Flow"
  – Sam King, University of Michigan, Ann Arbor
Internet Protocol Interoperability

- FTP
- HTTP
- NV
- TFTP

- TCP
- UDP

Overlays (running at hosts)

Virtual Network Infrastructure (runs globally)

Networks (run locally)

- Ethernet
- ATM
- FDDI
Internetworks

Router (Gateway)
Internetworks

FDDI Token Ring

Ethernet

Point-to-Point Link (e.g., ISDN)
IP Encapsulation

Example of protocol layers used to transmit from H1 to H8 in network shown on previous slide.
IP Service Model

• Choose minimal service model
  – All nets can implement
  – “Tin cans and a string” extremum

• Features:
  – Best-effort datagram delivery
  – Reliability, etc. as overlays
  – Packet format standardized
IPv4 Packet Format

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>19</td>
<td>31</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Version**
- **Hlen**
- **TOS**
- **Length**
- **Ident**
- **Flags**
- **Offset**
- **TTL**
- **Protocol**
- **Checksum**
- **SourceAddr**
- **DestinationAddr**
- **Options (variable length)**
- **Pad**
- **DATA**
Fields of IPv4 Header

- **Version**
  - Version of IP, example header is IPv4
  - First field so easy to implement case statement
- **Hlen**
  - Header length, in 32-bit *words*
- **TOS**
  - Type of Service (rarely used)
  - Priorities, delay, throughput, reliability
- **Length**
  - Length of datagram, in *bytes*
  - 16 bits, hence max. of 65,536 bytes
- **Fields for *fragmentation* and *reassembly***
  - Identifier
  - Flags
  - Offset
Header fields, continued

- **TTL**
  - Time to live (in reality, hop count)
  - 64 is the current default (128 also used)
- **Protocol**
  - e.g., TCP (6), UDP(17), etc.
- **Checksum**
  - Checksum of header (not CRC)
  - If header fails checksum, discard the whole packet
- **SourceAddr, DestinationAddr**
  - 32 bit IP addresses - global, IP-defined
- **Options**
  - length can be computed using Hlen
IP Datagram Delivery

• Every IP packet (datagram) contains the destination IP address
• The network part of the address uniquely identifies a single network that is part of the larger Internet.
• All hosts and routers that share the same network part of their address are connected to the same physical network.
• Routers can exchange packets on any network they’re attached to.
IP addresses

- Hierarchical, not flat as in Ethernet

```
<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Network</td>
<td>Host</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Network</td>
<td>Host</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Network</td>
<td>Host</td>
</tr>
</tbody>
</table>
```

- Written as four decimal numbers separated by dots:
  158.130.14.2
## Network Classes

<table>
<thead>
<tr>
<th>Class</th>
<th># of nets</th>
<th># of hosts per net</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>126</td>
<td>~16 million</td>
</tr>
<tr>
<td>B</td>
<td>8192</td>
<td>65534</td>
</tr>
<tr>
<td>C</td>
<td>~2 million</td>
<td>254</td>
</tr>
</tbody>
</table>
**IP Forwarding algorithm**

- If (Network # dest == Network # interface) then deliver to destination over interface
- else if (Network # dest in forwarding table) deliver packet to NextHop router
- else deliver packet to default router

- **Forwarding tables**
  - Contain (Network #, NextHop) pairs
  - Additional information, like
    - Built by routing protocol that learns the network topology, adapts to changes
Subnetting

• Problem: IP addressing scheme leads to fragmentation
  – A class B network with only 300 machines on it wastes > 65,000 addresses
  – Need a way to divide up a single network address space into multiple smaller subnetworks.

• Idea: One IP network number allocated to several physical networks.
  – The multiple physical networks are called *subnets*
  – Should be close together (why?)
  – Useful when a large company (or university!) has many physical networks.
Subnet Numbers

- **Solution:** *Subnetting*
  - All nodes are configured with *subnet mask*
  - Allows definition of a *subnet number*
    - All hosts on a physical subnetwork share the same *subnet number*

<table>
<thead>
<tr>
<th>Subnet Mask (255.255.255.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>111111111111111111111111</td>
</tr>
</tbody>
</table>

**Subnetted Address:**

<table>
<thead>
<tr>
<th>Network number</th>
<th>Subnet ID</th>
<th>Host ID</th>
</tr>
</thead>
</table>
Example of Subnetting

Subnet mask: 255.255.255.128
Subnet #: 128.96.34.0

H1  128.96.34.15

R1  128.96.34.1  128.96.34.130

H2  128.96.34.129

R1  128.96.33.1

H3  128.96.33.14

Subnet mask: 255.255.255.128
Subnet #: 128.96.34.128

Subnet mask: 255.255.255.0
Subnet #: 128.96.33.0
Subnets, continued

- Mask is bitwise-ANDed with address
- This is done at routers
- Router tables in this model:
  - <Subnet #, Subnet Mask, NextHop>
- Subnetting allows a set of physical networks to look like a single logical network from elsewhere
Forwarding Algorithm

D = destination IP address
for each forwarding table entry
(SubnetNumber, SubnetMask, NextHop)
    D1 = SubnetMask & D
    if D1 = SubnetNumber
        if NextHop is an interface
            deliver datagram directly to destination
        else
            deliver datagram to NextHop (router)
ARP - Address Resolution Protocol

• Problem:
  – Need mapping between IP and link layer addresses.

• Solution: ARP
  – Every host maintains IP–Link layer mapping table (cache)
  – Timeout associated with cached info (15 min.)

• Sender
  – Broadcasts “Who is IP addr X?”
  – Broadcast message includes sender’s IP & Link Layer address

• Receivers
  – Any host with sender in cache “refreshes” time-out
  – Host with IP address X replies “IP X is Link Layer Y”
  – Target host adds sender (if not already in cache)
ICMP: Internet Control Message Protocol

- Collection of error & control messages
- Sent back to the source when Router or Host cannot process packet correctly
- Error Examples:
  - Destination host unreachable
  - Reassembly process failed
  - TTL reached 0
  - IP Header Checksum failed
- Control Example:
  - Redirect – tells source about a better route
Domain Name System

• System for mapping mnemonic names for computers into IP addresses.
  zeta.cis.upenn.edu → 158.130.12.244

• Domain Hierarchy
• Name Servers
• Name Resolution
Records on Name Servers

• < Name, Value, Type, Class >
• Types
  – A  Host to address mappings
  – NS  Name server address mappings
  – CNAME  Aliases
  – MX  Mail server mappings
• Class IN for IP addresses
Name resolution

client

198.168.0.100

zeta.cis.upenn.edu

Local Name server

198.168.0.1

128.196.128.233

zeta.cis.upenn.edu

Root Name server

zeta.cis.upenn.edu

Upenn Name server

zeta.cis.upenn.edu

CIS Name server

198.168.0.100