Announcements

• Midterm 1 will be held on Thursday, Feb. 8th.
  – Example midterms from last year are on the web pages.

• It will cover all the material seen so far in class.
  – True/False
  – Multiple Choice
  – Short answer / essay
  – Problem solving
<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Host</td>
<td>Reference model – not actual implementation.</td>
</tr>
<tr>
<td>Application</td>
<td>Transmits <em>messages</em> (e.g. FTP or HTTP)</td>
</tr>
<tr>
<td>Presentation</td>
<td>Data format issues (e.g. big- vs. little-endian)</td>
</tr>
<tr>
<td>Session</td>
<td>Manages multiple streams of data</td>
</tr>
<tr>
<td>Transport</td>
<td>Process to process protocols</td>
</tr>
<tr>
<td>Network</td>
<td>Routes <em>packets</em> among nodes in network</td>
</tr>
<tr>
<td>Data Link</td>
<td>Packages bit streams into <em>frames</em></td>
</tr>
<tr>
<td>Physical</td>
<td>Transmits raw bits over link</td>
</tr>
</tbody>
</table>

*OSI Reference model*
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
Domain Name Hierarchy

edu
com
gov
mil
org
net
cornell ... upenn
cisco ... yahoo
nasa ... nsf
arpa ... navy ...
cis
seas
wharton ...

2/6/07
CIS/TCOM 551
Hierarchy of Name Servers

- Root Name Server
  - Cornell Name Server
  - Upenn Name Server
    - CIS Name Server
    - SEAS Name Server
    - Wharton Name Server
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

Local Name server

Root Name server

Upenn Name server

CIS Name server

198.168.0.100

zeta.cis.upenn.edu

198.168.0.1

zeta.cis.upenn.edu

128.196.128.233

zeta.cis.upenn.edu

198.168.0.100

zeta.cis.upenn.edu

198.168.0.100

zeta.cis.upenn.edu

198.168.0.100

zeta.cis.upenn.edu

198.168.0.100

zeta.cis.upenn.edu
Protocol Stack Revisited

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

UDP and TCP/IP

So far…
# Application vs. Network

<table>
<thead>
<tr>
<th>Application Needs</th>
<th>Network Char.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliable, Ordered, Single-Copy Message Delivery</td>
<td>Drops, Duplicates and Reorders Messages</td>
</tr>
<tr>
<td>Arbitrarily large messages</td>
<td>Finite message size</td>
</tr>
<tr>
<td>Flow Control by Receiver</td>
<td>Arbitrary Delay</td>
</tr>
<tr>
<td>Supports multiple applications per-host</td>
<td>…</td>
</tr>
</tbody>
</table>
User Datagram Protocol (UDP)

- Simplest transport-layer protocol
- Just exposes IP packet functionality to application level
- *Ports* identify sending/receiving process
  - Demultiplexing information
  - *(port, host)* pair identifies a network process
UDP End-to-End Model

- Multiplexing/Demultiplexing with Port number
Using Ports

- Client contacts Server at a *well-known port*
  - SMPT: port 25
  - DNS: port 53
  - POP3: port 110
  - Unix talk: port 517
  - In unix, ports are listed in `/etc/services`

- Sometimes Client and Server agree on a different port for subsequent communication

- Ports are an abstraction
  - Implemented differently on different OS’s
  - Typically a message queue
Transmission Control Protocol (TCP)

- Most widely used protocol for reliable byte streams
  - Reliable, in-order delivery of a stream of bytes
  - Full duplex: pair of streams, one in each direction
  - Flow and congestion control mechanisms
  - Like UDP, supports ports

- Built on top of IP (hence TCP/IP)
TCP End-to-End Model

- Buffering corrects errors but may introduce delays
Packet Format

- Flags
  - SYN
  - FIN
  - RESET
  - PUSH
  - URG
  - ACK

- Fields

```
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SrcPort</td>
<td>DstPort</td>
<td></td>
</tr>
<tr>
<td>SequenceNum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acknowledgment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL</td>
<td>0</td>
<td>Flags</td>
</tr>
<tr>
<td>Checksum</td>
<td>UrgPtr</td>
<td></td>
</tr>
<tr>
<td>Options (variable)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

DATA
Three-Way Handshake

Active participant (client)  Passive participant (server)

SYN, SequenceNum = x

SYN + ACK, SequenceNum = y,
Acknowledgment = x + 1

ACK, Acknowledgment = y + 1
TCP State Transitions
TCP Receiver

- Maintains a buffer from which application reads
- Advertises < buffer size as the window for sliding window
- Responds with Acknowledge and AdvertisedWindow on each send; updates byte counts when data O.K.
- Application blocked until read() O.K.
TCP Sender

- Maintains a buffer; sending application is blocked until room in the buffer for its write
- Holds data until acknowledged by receiver as *successfully received*
- Implement window expansion and contraction; note difference between *flow* and *congestion* control
TCP Flow & Congestion Control

- Flow vs. Congestion Control
  - Flow control protects the recipient from being overwhelmed.
  - Congestion control protects the network from being overwhelmed.

- TCP Congestion Control
  - Additive Increase / Multiplicative Decrease
  - Slow Start
  - Fast Retransmit and Fast Recovery
Increase and Decrease

- A value CongestionWindow is used to control the number of unacknowledged transmissions.
- This value is increased linearly until timeouts for ACKs are missed.
- When timeouts occur, CongestionWindow is decreased by half to reduce the pressure on the network quickly.
- The strategy is called “additive increase / multiplicative decrease”.
Additive Increase
TCP Sawtooth Pattern
Slow Start

- Sending the entire window immediately could cause a traffic jam in the network.
- Begin “slowly” by setting the congestion window to one packet.
- When acknowledgements arrive, double the congestion window.
- Continue until ACKs do not arrive or flow control dominates.
Slow Start
Network Vulnerabilities

- **Anonymity**
  - Attacker is remote, origin can be disguised
  - Authentication

- **Many points of attack**
  - Attacker only needs to find weakest link
  - Attacker can mount attacks from many machines

- **Sharing**
  - Many, many users sharing resources

- **Complexity**
  - Distributed systems are large and heterogeneous

- **Unknown perimeter**

- **Unknown attack paths**
Syn Flood Attack

• Recall TCP’s 3-way handshake:
  – SYN     ---  SYN+ACK    ---  ACK

• Receiver must maintain a queue of partially open TCP connections
  – Called SYN_RECV connections
  – Finite resource (often small: e.g. 20 entries)
  – Timeouts for queue entries are about 1 minute.

• Attacker
  – Floods a machine with SYN requests
  – Never ACKs them
  – Spoofs the sending address  (Why? Two reasons!)
Reflected denial of service

- Broadcast a ping request
  - For sender’s address put target’s address
  - All hosts reply to ping, flooding the target with responses

- Hard to trace
- Hard to prevent
  - Turn off ping? (Makes legitimate use impossible)
  - Limit with network configuration by restricting scope of broadcast messages
(Distributed) Denial of Service

- Coordinate multiple subverted machines to attack
- Flood a server with bogus requests
  - TCP SYN packet flood
  - > 600,000 packets per second
- Detection & Assessment?
  - 12,800 attacks at 5000 hosts! (in 3 week period during 2001)
  - IP Spoofing (forged source IP address)
  - [http://www.cs.ucsd.edu/users/savage/papers/UsenixSec01.pdf](http://www.cs.ucsd.edu/users/savage/papers/UsenixSec01.pdf)
- Prevention?
  - Filtering?
  - Decentralized file storage?