Announcements

- First project: Due: 25 Sept. 2006
  - [http://www.cis.upenn.edu/~cse331/project1.html](http://www.cis.upenn.edu/~cse331/project1.html)
  - Please e-mail savi@seas.upenn.edu with your project group by Weds. 13 Sept.
  - If you need a group, send e-mail

- Please put "cse331" in the subject of all course-related e-mail

- Prof. Zdancewic will be away Sept. 18 & 20.
  - Class will be taught by Peng Li
Plan for Today:

• Wrap up buffer overflows discussion.

• Take a step back and begin talking about networks.
  – Basic network architecture
  – Terminology
  – Simple performance characteristics
Tool support for C/C++

- Extensions to gcc that do array bounds checking
- Link against "safe" versions of libc (e.g. libsafe)
- Test programs with tools such as Purify, Splint, valgrind
- Compile programs using tools such as:
  - Stackguard and Pointguard (Cowan et al., immunix.org)

- Research compilers:
  - Ccured (Necula et al.)
  - Cyclone (Morrisett et al.)

- Binary rewriting techniques
  - Software fault isolation (Wahbe et al.)
Defeating Buffer Overflows

• Use a typesafe programming language
  – Java/C# are not vulnerable to these attacks
  – Garbage collection eliminates most memory management problems

• Some operating systems move the start of the stack on a per-process basis:
  – E.g. eniac-l
  – Doesn't provide complete protection (but it does make things harder)
The Four Major Networks

- Telephone
- Television
- Radio
- Internet (grew out of ARPANET–late 1960’s)

- Starting to see hybrids…

- Computer networks
  - General purpose programmable hardware
  - Support many different applications
How to build such a network?

- Connectivity
- Efficient Resource Sharing
- Functionality
- Performance
- Security
Requirement: Connectivity

- Goal of a network is to get information from one place to another
  - Source
  - Destination
  - Nodes or Hosts

- Network paths
  - Can be direct or indirect
  - Can be static or dynamic

Specified by an address
Connectivity: Direct Links

Point to Point
- e.g. telephone

Multiple Access
- e.g. Ethernet
Connectivity: Switched Networks

Network "Cloud"

Switches or Hubs

Hosts

Direct Links
Connectivity: Internetworks
Resource Sharing: Multiplexing

- How can multiple hosts share the network if they want to use it at the same time?
  - Sharing links
  - Sharing switches
Multiplexing: STDM & FDM

- Synchronous Time-division Multiplexing (STDM)
  - “Time sharing”
  - Divide time into equal sized quanta
  - Round-robin

- Frequency-division Multiplexing (FDM)
  - Transmit all flows at different frequencies
  - Radio or Television

- Limitations:
  - Wasted resources
  - Maximum # flows can’t be changed
Statistical Multiplexing

- Data is partitioned into *packets*
- Routing decision is made per packet
- Better resource usage than STDM
- Fairness? Congestion?
Functionality

- Different applications require different services

Reliable unicast channel (e.g. file transfer)

Unreliable multicast channel with in-order delivery (e.g. video)
Functionality & Dealing with Failure

- Fairness
- Congestion
- Quality of Service
- Bit or burst errors
- Link or node outages
Performance

• **Bandwidth** (throughput)
  – The number of bits that can be transmitted over the network in a certain period of time.
  – Measured in bits/sec

• **Latency** (delay)
  – How long it takes a single bit to propagate from one end of the network to the other.
  – Measured in seconds

• **Round Trip Time (RTT)**
  – How long it takes for a bit to get from one end of the network to the other and back.
### Connectivity: Direct Link Technologies

<table>
<thead>
<tr>
<th>Technology</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wired Ethernet</td>
<td>10, 100 Mbps, 1, 10 Gbps</td>
</tr>
<tr>
<td>SONET fiber</td>
<td>up to 9.6 Gbps</td>
</tr>
<tr>
<td>CATV</td>
<td>1-6 Mbps, asymmetric</td>
</tr>
<tr>
<td>ADSL</td>
<td>Downstream: 1.5-55.2 Mbps Upstream: 16-640 Kbps</td>
</tr>
<tr>
<td>ISDN</td>
<td>64 Kbps*n with bonding</td>
</tr>
<tr>
<td>POTS</td>
<td>56 Kbps</td>
</tr>
<tr>
<td>Wireless Ethernet</td>
<td>2, 11, 22, ... Mbps</td>
</tr>
<tr>
<td>Infrared IrDA</td>
<td>115 Kbps to 4 Mbps</td>
</tr>
<tr>
<td>CDPD</td>
<td>19.2 Kbps</td>
</tr>
</tbody>
</table>
Performance: Delay x Bandwidth

Delay x Bandwidth determines the number of bits that can be “in flight”.

For efficient resource usage: keep the pipe full.
Key Equations

Latency = Propagation + Transmit + Queue

Propagation = Distance / SpeedOfLight

Transmit = Size / Bandwidth
Total Latency: Direct Link

Data moves through the link at the speed of light.

Time
0 Data ready to be sent
Data moves through the link at the speed of light.

Time

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Data ready to be sent</td>
</tr>
<tr>
<td>( t = \frac{\text{Size}}{\text{Bandwidth}} )</td>
<td>Data in the link</td>
</tr>
</tbody>
</table>
Total Latency: Direct Link

Data moves through the link at the speed of light.

<table>
<thead>
<tr>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Data ready to be sent</td>
</tr>
<tr>
<td>( t = \frac{\text{Size}}{\text{Bandwidth}} )</td>
<td>Data in the link</td>
</tr>
<tr>
<td>( t+k )</td>
<td>Data traveling through the link</td>
</tr>
</tbody>
</table>
Total Latency: Direct Link

Data moves through the link at the speed of light.

Time
0  Data ready to be sent
\( t = \text{Size}/\text{Bandwidth} \)  Data in the link
\( t+k \)  Data traveling through the link
prop = Distance/LightSpeed  First bit arrives at destination
Total Latency: Direct Link

Data moves through the link at the speed of light.

Time
0  Data ready to be sent
\( t = \frac{\text{Size}}{\text{Bandwidth}} \)  Data in the link
\( t+k \)  Data traveling through the link
\( \text{prop} = \frac{\text{Distance}}{\text{LightSpeed}} \)  First bit arrives at destination
\( \text{prop} + t \)  Last bit arrives at destination
Paths Are Made of *Links*

- Links are interconnected by zero or more *network elements*, e.g., switches, routers, hubs, bridges, etc.
- Path delay is sum of link delays plus queuing (switching) delays
- Path throughput = *bottleneck link t’put*
Tradeoffs

• RTT from Penn to Stanford is approx. 100ms
• 1.4 GHz workstation
  – 140 million cycles elapsed in that time

• Data compression
  – Trades machine cycles for bandwidth

• (Question: Why is RTT important?)
Bandwidth vs. Latency

• Which is the better deal:
  – Improve your *bandwidth* from 1 Mbps to 100 Mbps, or
  – Improve your *RTT* from 100 ms to 1 ms?

• The answer depends on what you need to send.
Latency Bound

- Send 1 byte

<table>
<thead>
<tr>
<th>Transmit Time</th>
<th>1 Mbps</th>
<th>100 Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 µs</td>
<td>.08 µs</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Perceived Latency</th>
<th>100 ms</th>
<th>100.008 ms</th>
<th>100.00008 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Mbps</td>
<td>1 ms</td>
<td>1.008 ms</td>
<td>1.00008 ms</td>
</tr>
<tr>
<td>100 Mbps</td>
<td>99%</td>
<td>99%</td>
<td>.008%</td>
</tr>
</tbody>
</table>

99%
## Bandwidth Bound

- Send 25 MB

<table>
<thead>
<tr>
<th>Transmit Time</th>
<th>1 Mbps</th>
<th>3.5 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 Mbps</td>
<td>21 sec</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Perceived Latency</th>
<th>100 ms</th>
<th>1 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Mbps</td>
<td>210.1 sec</td>
<td>210.001 sec</td>
</tr>
<tr>
<td>100 Mbps</td>
<td>21.1 sec</td>
<td>21.001 sec</td>
</tr>
</tbody>
</table>

90%
Some Units and Measurements

- Mbps = $10^6$ bits/sec
- byte = 8 bits
- KB = $2^{10}$ bytes (= 8,192 bits)
- MB = $2^{20}$ bytes (= 8,388,608 bits)
- ms = $10^{-3}$ seconds
- µs = $10^{-6}$ seconds

- Speed of light:
  - Vacuum : $3 \times 10^8$ m/sec
  - Copper or Fiber: $2 \times 10^8$ m/sec