Review: A Little Queuing Theory

- Queuing models assume state of equilibrium: input rate = output rate
- Notation:
  - \( r \): average number of arriving customers/second
  - \( T_{\text{ser}} \): average time to service a customer (traditionally \( \mu = 1/T_{\text{ser}} \))
  - \( u \): server utilization (0..1): \( u = r \times T_{\text{ser}} \)
  - \( T_q \): average time/customer in queue
  - \( T_{\text{sys}} \): average time/customer in system: \( T_{\text{sys}} = T_q + T_{\text{ser}} \)
  - \( L_q \): average length of queue: \( L_q = r \times T_q \)
  - \( L_{\text{sys}} \): average length of system: \( L_{\text{sys}} = r \times T_{\text{sys}} \)
- Little's Law: \( \text{Length}_{\text{system}} = \text{rate} \times \text{Time}_{\text{system}} \)
  (Mean number customers = arrival rate \times mean service time)

Review: I/O Benchmarks

- Scaling to track technological change
- TPC: price performance as normalizing configuration feature
- Auditing to ensure no foul play
- Throughput with restricted response time is normal measure
- Benchmarks to measure Availability, Maintainability?

Review: Availability benchmarks

- Availability benchmarks can provide valuable insight into availability behavior of systems
  - reveal undocumented availability policies
  - illustrate impact of specific faults on system behavior
- Methodology is best for understanding the availability behavior of a system
  - extensions are needed to distill results for automated system comparison
- A good fault-injection environment is critical
  - need realistic, reproducible, controlled faults
  - system designers should consider building in hooks for fault-injection and availability testing
- Measuring and understanding availability will be crucial in building systems that meet the needs of modern server applications
  - this benchmarking methodology is just 1st step towards goal

Networks

- Goal: Communication between computers
- Eventual Goal: treat collection of computers as if one big computer, distributed resource sharing
- Theme: Different computers must agree on many things
  - Overriding importance of standards and protocols
  - Error tolerance critical as well
- Warning: Terminology-rich environment

Facets people talk a lot about:
- direct (point-to-point) vs. indirect (multi-hop)
- topology (e.g., bus, ring, DAG)
- routing algorithms
- switching (aka multiplexing)
- wiring (e.g., choice of media, copper, coax, fiber)

What really matters:
- latency
- bandwidth
- cost
- reliability
Interconnections (Networks)

- Examples (see Figure 7.19, page 633):
  - Wide Area Network (ATM): 100-1000s nodes; ~ 5,000 kilometers
  - Local Area Networks (Ethernet): 10-1000 nodes; ~ 1-2 kilometers
  - System/Storage Area Networks (FC-AL): 10-100s nodes; ~ 0.025 to 0.1 kilometers per link

SAN: Storage vs. System

- Storage Area Network (SAN): A block I/O oriented network between application servers and storage
  - Fibre Channel is an example
- Usually high bandwidth requirements, and less concerned about latency
  - In 2001: 1 Gbit bandwidth and millisecond latency OK
- Commonly a dedicated network (that is, not connected to another network)
- May need to work gracefully when saturated
- Given larger block size, may have higher bit error rate (BER) requirement than LAN

SAN: Storage vs. System

- System Area Network (SAN): A network aimed at connecting computers
  - Myrinet is an example
- Aimed at High Bandwidth AND Low Latency.
  - In 2001: > 1 Gbit bandwidth and ~ 10 microseconds
- May offer in order delivery of packets
- Given larger block size, may have higher bit error rate (BER) requirement than LAN

ABCs of Networks

- Starting Point: Send bits between 2 computers
- Queue (FIFO) on each end
- Information sent called a "message"
- Can send both ways ("Full Duplex")
- Rules for communication? "protocol"
  - Inside a computer:
    » Loads/Stores: Request (Address) & Response (Data)
    » Need Request & Response signaling

A Simple Example

- What is the format of message?
  - Fixed? Number bytes?
  - 0: Please send data from Address
  - 1: Packet contains data corresponding to request
- Header/Trailer: information to deliver a message
- Payload: data in message (1 word above)
Questions About Simple Example

- What if more than 2 computers want to communicate?
  - Need computer "address field" (destination) in packet
- What if packet is garbled in transit?
  - Add "error detection field" in packet (e.g., Cyclic Redundancy Chk)
- What if packet is lost?
  - More "elaborate protocols" to detect loss (e.g., NAK, ARQ, time outs)
- What if multiple processes/machine?
  - Queue per process to provide protection
- Simple questions such as these lead to more complex protocols and packet formats => complexity

A Simple Example Revisited

- What is the format of packet?
  - Fixed? Number bytes?
  - Request/Response Address/Data
  - 1 bit
  - 32 bits
  - 4 bits

<table>
<thead>
<tr>
<th>Request/</th>
<th>Address/Data</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:</td>
<td>Request—Please send data from Address</td>
<td></td>
</tr>
<tr>
<td>01:</td>
<td>Reply—Packet contains data corresponding to request</td>
<td></td>
</tr>
<tr>
<td>10:</td>
<td>Acknowledge request</td>
<td></td>
</tr>
<tr>
<td>11:</td>
<td>Acknowledge reply</td>
<td></td>
</tr>
</tbody>
</table>

Software to Send and Receive

- SW Send steps
  1: Application copies data to OS buffer
  2: OS calculates checksum, starts timer
  3: OS sends data to network interface HW and says start
- SW Receive steps
  3: OS copies data from network interface HW to OS buffer
  2: OS calculates checksum, if matches send ACK; if not, deletes message (sender resends when timer expires)
  1: If OK, OS copies data to user address space and signals application to continue
- Sequence of steps for SW: protocol
  - Example similar to UDP/IP protocol in UNIX

Network Performance Measures

- Overhead: latency of interface vs. Latency: network

San 252 Administrivia

- HW #1 due Saturday: send electronically to TA
- Pick a partner, project by Monday; send electronically to me, TA
- I’ll be available Monday afternoon to talk

Universal Performance Metrics

<table>
<thead>
<tr>
<th>Sender</th>
<th>Transmission time (size + bandwidth)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(processor busy)</td>
<td>Time of Flight</td>
</tr>
<tr>
<td>Receiver</td>
<td>Transmission time (size + bandwidth)</td>
</tr>
<tr>
<td>(processor busy)</td>
<td></td>
</tr>
<tr>
<td>Transport Latency</td>
<td></td>
</tr>
<tr>
<td>Total Latency</td>
<td></td>
</tr>
</tbody>
</table>

Total Latency = Sender Overhead + Time of Flight + Message Size x BW + Receiver Overhead

Includes header/trailer in BW calculation?
Total Latency Example

- 1000 Mbit/sec, sending overhead of 80 µsec & receiving overhead of 100 µsec.
- 10000 byte message (including the header), allows 10000 bytes in a single message
- 3 situations: distance 1000 km v. 0.5 km v. 0.01
- Speed of light ~ 300,000 km/sec (1/2 in media)
- \[ \text{Latency}_{0.01\text{km}} = \]
- \[ \text{Latency}_{0.01\text{km}} = \]
- \[ \text{Latency}_{1000\text{km}} = \]

Universal Metrics

- Apply recursively to all levels of system
- inside a chip, between chips on a board, between computers in a cluster, ...
- Look at WAN v. LAN v. SAN

Simplified Latency Model

- Total Latency = Overhead + Message Size / BW
- Overhead = Sender Overhead + Time of Flight + Receiver Overhead

Example: show what happens as vary
- Overhead: 1, 25, 500 µsec
- BW: 10, 100, 1000 Mbit/sec (factors of 10)
- Message Size: 16 Bytes to 4 MB (factors of 4)
- If overhead 500 µsec, how big a message > 10 Mb/s?

Overhead, BW, Size

Delivered BW

Msg Size

Packet size

- 95% Ms, 30% bytes for packets ~ 200 bytes
- > 50% data transferred in packets = 8KB

Impact of Overhead on Delivered BW

- BW model: Time = overhead + msg size/peak BW
Interconnect Issues

• Performance Measures
• Network Media

Network Media

Twisted Pair: Copper, 1mm think, twisted to avoid attenna effect (telephone)
“Cat 5” is 4 twisted pairs in bundle

Coaxial Cable: Used by cable companies: high BW, good noise immunity

Copper core Insulator Braided outer conductor Buffer

Fiber Optics: Light: 3 parts are cable, light source, light detector.

Note fiber is unidirectional; need 2 for full duplex

Twisted Pair:

Copper core Insulator Braided outer conductor Buffer

Fiber Optics:

Transmitter + L.E.D. – Laser Diode

Receiver Photodetector

Silica core Cladding

 Laboratories

Copper cable: 1mm thick, twisted to avoid antenna effect.

“Cat 5” is 4 twisted pairs in a bundle.

Used by cable companies: high bandwidth, good noise immunity.

Light: 3 parts are cable, light source, light detector. Note fiber is unidirectional; need 2 for full duplex.

Twisted Pair:

Copper core Insulator Braided outer conductor Buffer

Fiber Optics:

Transmitter + L.E.D. – Laser Diode

Receiver Photodetector

Silica core Cladding

Fiber Optics:

Transmitter + L.E.D. – Laser Diode

Receiver Photodetector

Silica core Cladding

Wave Division Multiplexing Fiber

• Send N independent streams on single fiber!
• Just use different wavelengths to send and demultiplex at receiver
• WDM in 2000: 40 Gbit/s using 8 wavelengths
• Plan to go to 80 wavelengths => 400 Gbit/s!
• A figure of merit: BW* max distance (Gbit-km/sec)
• 10X/4 years, or 1.8X per year

Compare Media

• Assume 40 2.5" disks, each 25 GB, Move 1 km
• Compare Cat 5 (100 Mbit/s), Multimode fiber (1000 Mbit/s), single mode (2500 Mbit/s), and car
• Cat 5: 1000 x 1024 x 8 Mb / 100 Mb/s = 23 hrs
• MM: 1000 x 1024 x 8 Mb / 1000 Mb/s = 2.3 hrs
• SM: 1000 x 1024 x 8 Mb / 2500 Mb/s = 0.9 hrs
• Car: 5 min + 1 km / 50 kph + 10 min = 0.25 hrs
• Car of disks = high BW media

Interconnect Issues

• Performance Measures
• Network Media
• Connecting Multiple Computers
Connecting Multiple Computers

- Shared Media vs. Switched: pairs communicate at same time, “point-to-point” connections
- Aggregate BW in switched network is many times shared
  - point-to-point faster since no arbitration, simpler interface
- Arbitration in Shared network?
  - Central arbiter for LAN?
  - Listen to check if being used (“Carrier Sensing”)
  - Listen to check if collision (“Collision Detection”)
  - Random resend to avoid repeated collisions; not fair arbitration;
  - OK if low utilization

Summary: Interconnections

- Communication between computers
- Packets for standards, protocols to cover normal and abnormal events
- Performance issues: HW & SW overhead, interconnect latency, bisection BW
- Media sets cost, distance
- Shared vs. Switched Media determines BW

Projects

- See [www.cs/~pattrsn/252S01/suggestions.html](http://www.cs/~pattrsn/252S01/suggestions.html)