Lecture 7: Interconnection Networks

Prof. Fred Chong
ECS 250A Computer Architecture
Winter 1999

(Adapted from Patterson CS252 Copyright 1998 UCB)

Review: Storage System Issues
- Historical Context of Storage I/O
- Secondary and Tertiary Storage Devices
- Storage I/O Performance Measures
- Processor Interface Issues
- Redundant Arrays of Inexpensive Disks (RAID)
- ABCs of UNIX File Systems
- I/O Benchmarks

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

I/O to External Devices and Other Computers

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
  - Fault tolerance critical as well
- Warning: Terminology-rich environment

Example Major Networks

<table>
<thead>
<tr>
<th>Network Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARPA net</td>
<td>T1, 56Kbps</td>
</tr>
<tr>
<td>T3, 230Kbps</td>
<td></td>
</tr>
<tr>
<td>IP - Internet Protocol</td>
<td>TCP - Transmission Control Protocol</td>
</tr>
<tr>
<td>Token Ring</td>
<td>Ethernet</td>
</tr>
<tr>
<td>FDDI</td>
<td></td>
</tr>
<tr>
<td>CS Net</td>
<td>Relay</td>
</tr>
<tr>
<td>Bitnet</td>
<td>NSF Net</td>
</tr>
<tr>
<td>NSF Net</td>
<td>ATM</td>
</tr>
<tr>
<td>ATM</td>
<td>1.6Mbps</td>
</tr>
<tr>
<td>10 Mbps</td>
<td>4Gbps</td>
</tr>
<tr>
<td>1Gbps</td>
<td></td>
</tr>
<tr>
<td>10 Mbps</td>
<td></td>
</tr>
<tr>
<td>1Gbps</td>
<td></td>
</tr>
<tr>
<td>10 Gbps</td>
<td></td>
</tr>
<tr>
<td>40 Gbps</td>
<td></td>
</tr>
<tr>
<td>100 Gbps</td>
<td></td>
</tr>
</tbody>
</table>
Networks

- 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:
  - MPP networks (SP2): 100s nodes; 5 25 meters per link
  - Local Area Networks (Ethernet): 100s nodes; 5 1000 meters
  - Wide Area Network (ATM): 1000s nodes; 5 5,000,000 meters

More Network Background

- Connection of 2 or more networks: Internetworking

  - LAN: workstations, cost
  - WAN: telecommunications, phone call revenue

  - Try for single terminology

  - Motivate the interconnection complexity incrementally

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?
  - Header/Response Address/Data
  - 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., CRC)

- 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 Revisted

- What is the format of packet?
  - Fixed? Number bytes?
  - Address/Data CRC

<table>
<thead>
<tr>
<th>1 bit</th>
<th>32 bits</th>
<th>4 bits</th>
</tr>
</thead>
</table>
  00: Request—Please send data from Address
  01: Reply—Packet contains data corresponding to request
  10: Acknowledge request
  11: Acknowledge reply

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
  1: If OK, OS copies data to user address space and signals application to continue
  2: OS calculates checksum, if matches send ACK; if not, deletion message (sender resends when timer expires)

  Sequence of steps for SW protocol
  - Example similar to UDP/IP protocol in UNIX

Network Performance Measures

- Overhead: latency of interface vs. Latency: network

Universal Performance Metrics

- Sender Overhead
- Transmission time (size + bandwidth)

- Receiver
- Time of Flight (size + bandwidth)
- Overhead
- Transport Latency
- Total Latency

Example Performance Measures

<table>
<thead>
<tr>
<th>Interconnect</th>
<th>MPP</th>
<th>LAN</th>
<th>WAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>CM-5</td>
<td>Ethernet</td>
<td>ATM</td>
</tr>
<tr>
<td>Bi-direction</td>
<td>N x 5 Mba</td>
<td>1.125 Mba</td>
<td>N x 10 Mba</td>
</tr>
<tr>
<td>Int. Link BW</td>
<td>20 MB/s</td>
<td>1.125 Mba</td>
<td>10 Mba</td>
</tr>
<tr>
<td>Transport Latency</td>
<td>5 µsec</td>
<td>15 µsec</td>
<td>50 to 10,000 µ</td>
</tr>
<tr>
<td>HW Overhead to/from</td>
<td>6/6 µsec</td>
<td>6/6 µsec</td>
<td></td>
</tr>
<tr>
<td>SW Overhead to/from</td>
<td>1.6/12.4 µsec</td>
<td>200/241 µsec</td>
<td>207/300 µsec (TCP/IP on LAN/WAN)</td>
</tr>
</tbody>
</table>

Software overhead dominates in LAN, WAN

Total Latency Example

- 10 Mbit/sec, sending overhead of 230 µsec & receiving overhead of 270 µsec.
- a 1000 byte message (including the header),
  allows 1000 bytes in a single message.
- 2 situations: distance 0.1 km vs. 1000 km
- Speed of light = 299,792.5 km/sec (1/2 in media)
- Latency (6.8 µsec)
  - Latency1000km =
- Long time of flight => complex WAN protocol
Total Latency Example
- 10 Mbit/sec., sending overhead of 230 µsec & receiving overhead of 270 µsec.
- a 1000 byte message (including the header), allows 1000 bytes in a single message.
- 2 situations: distance 100 m vs. 1000 km
  - Speed of light = 299,792.5 km/sec
  - Latency_{100m} = 230 + 0.1km / (50% x 299,792.5) = 1000 x 8 / 10 + 270
  - Latency_{1000km} = 230 + 0.67 km / (50% x 299,792.5) + 1000 x 8 / 10 + 270
  - Latency_{adder} = 230 + 5671 + 800 + 270 = 7971 µsec
- Long time of flight ⇒ complex WAN protocol

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 Mbits?

Overhead, BW, Size
- Delivered BW
- Msg Size
- How big are real messages?

Measurement: Sizes of Message for NFS
- 95% Msgs, 30% bytes for packets $\leq$ 200 bytes
- > 50% data transferred in packets $\approx$ 8KB

Impact of Overhead on Delivered BW
- BW model: Time = overhead + msg size/peak BW
- > 50% data transferred in packets $\approx$ 8KB

Building a Better Butterfly: The Multiplexed Metabutterfly
Chong, Brewer, Leighton, and Knight
Transformations

- Butterfly
- Dilated Butterfly
- Multi-butterfly
- Meta-butterfly
- Multiplex

- lowest cost
- 2X cost
- high complex
- lower complex
- lower cost

- poor
- better
- great
- great
- good

- fault & congestion performance

Outline

- Expander Networks
  - Good, but hard to build
- Hierarchical Construction
  - Much simpler
  - Almost as good in theory
  - Just as good in simulation
- Multiplexing
  - Much better grouping
  - Randomize for efficiency
- The cost of a butterfly the performance of a multibutterfly

Butterfly

Splitter Network

Dilated Butterfly

Wiring Splitters

Dilated
Better
Expansion

- Definition: A splitter has expansion if every set of $S < \alpha M$ inputs reaches at least $\beta S$ outputs in each of $r$ directions, where $\beta < 1$ and $\alpha < 1 / (\beta r)$
- Random wiring produces the best expansion

Faults and Congestion

- Randomly wired
- Extensively studied:
  - [Bassalygo & Pinsker 74] [Upfal 89]
  - [Leighton & Maggs 89] [Arora, Leighton & Maggs 90]
- Tremendous fault and congestion tolerance

Multibutterflies

What’s Wrong?

Wiring Complexity

Relative Complexity

Wires:

Cables:
**K-Extension Properties**

- Preserve Expansion (with high probability):
  \[
  \beta_{new} = \beta - 2
  \]
  \[
  \alpha_{new} = \frac{\alpha^2}{\beta^2 c^4 + 4\alpha}
  \]

- [Brewer, Chong, & Leighton, STOC94]

**Empirical Results**

- Methodology of [Chong & Knight, SPAA92]
  - Uniformly distributed router faults
  - 1024-processor networks with 5 stages
  - shared-memory traffic pattern

- Metabutterflies (with metanode sizes 4, 16, 32) perform as well as the Multibutterfly

**Multiplexing**

- Multiplex cables
  - Not possible with multibutterfly
- Random Destinations
  - Can remove half the wires!
  - 2X performance of comparable butterfly
Multiplexing (Bit-Inverse)
- Over 5X better on bit-inverse
- Multiple logical paths without excess physical bandwidth

Load Balancing
- Why is bit-inverse worse than random?

Unbalanced Loading
- Solutions:
  - balance bit-inverse
  - more wires in first stage
  - more bandwidth in first stages

Randomized Multiplexing
- Within cables, packet destination unimportant
  - Could be random
  - Assign each packet to any output
- Better bandwidth
  - No fixed time slots
  - No extra headers
  
<table>
<thead>
<tr>
<th>Fixed</th>
<th>Extra Headers</th>
<th>Randomized</th>
</tr>
</thead>
</table>

Summary
- Metabutterfly
  - best fault and congestion tolerance
- Multiplexed Metabutterfly
  - comparable cost to butterfly
  - much better fault and congestion tolerance
- K-extensions and Multiplexing
  - applicable to other networks (e.g., fat trees)

Conclusions
- Other Expander-Based Networks
  - Fat-Trees
  - Deterministic Constructions
- Non-Random K-extensions
  - How many permutations?
- Other Networks with Multiplicity
- Expanders are great, but were hard to build
- K-extensions are the solution
  - Allow Fixed Cabling Degree
  - Retain Theoretical Properties
  - Equal Multibutterflies in Simulation
## Interconnect Outline

- Performance Measures
- (Metabutterfly Example)
- Interface Issues

## HW Interface Issues

### Memory Bus
- Low latency, high bandwidth
- Limited length to maintain speed
- Proprietary protocols

### I/O Bus
- Standard protocols
- Slow
- Can be long

## SW Interface Issues

### How to connect network to software?
- Programmed I/O?
- DMA?
- Receiver interrupted or receiver polls?

### Things to avoid
- Invoking operating system in common case
  - Why would designers want to invoke OS for communication?
  - Operating at uncached memory speed
    (e.g., check status of network interface)

## CM-5 Software Interface

### CM-5 example (MPP)
- User does msgs w/out OS
- Time per poll 1.6 µs; time per interrupt 19 µs
- Minimum time to handle message: 0.5 µs
- Enable/disable 4.5/3.5 µs

### As rate of messages arriving changes, use polling or interrupt?
- Solution: Always enable interrupts, have interrupt routine poll until no messages pending
- Low rate => interrupt
- High rate => polling

## HW Interface Issues

- Where to connect network to computer?
  - Cache consistent to avoid flushes?
  - RAM
  - Hard Drive
  - Standard interface card?
  - MPP
  - LAN/WAN

- Ideal: high bandwidth, low latency, standard interface

## Interconnect Issues

- Performance Measures
- Interface Issues
- Network Media
Network Media

- **Twisted Pair:** Copper, 1mm think, twisted to avoid antenna effect (telephone)
- **Coaxial Cable:** Used by cable companies: high BW, good noise immunity
- **Fiber Optics:** Light: 3 parts are cable, light source, light detector. Multimode light disperse (LED), Single mode single wave (laser)

Costs of Network Media (1995)

<table>
<thead>
<tr>
<th>Media</th>
<th>Bandwidth</th>
<th>Distance</th>
<th>Cost/meter</th>
<th>Cost/interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>twisted pair</td>
<td>1 Mb/s</td>
<td>1 km</td>
<td>$0.23</td>
<td>$2</td>
</tr>
<tr>
<td>copper wire</td>
<td>(20 Mb/s)</td>
<td>(0.1 km)</td>
<td>$1.64</td>
<td>$5</td>
</tr>
<tr>
<td>coaxial cable</td>
<td>10 Mb/s</td>
<td>1 km</td>
<td>$1.03</td>
<td>$1000</td>
</tr>
<tr>
<td>multimode</td>
<td>600 Mb/s</td>
<td>2 km</td>
<td>$1.64</td>
<td>$1000</td>
</tr>
<tr>
<td>optical fiber</td>
<td>single</td>
<td>2000 Mb/s</td>
<td>$1.03</td>
<td>$1000</td>
</tr>
<tr>
<td>fiber optic</td>
<td>mode</td>
<td>100 km</td>
<td>$1.64</td>
<td>$1000</td>
</tr>
</tbody>
</table>

Note: more elaborate signal processing allows higher BW from copper (ADSL)

Single mode Fiber measures: BW * distance as 3X/year

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 re-send to avoid repeated collisions; not fair arbitration;
  - OK if low utilization

Switch Topology

- **Structure of the interconnect**
  - Degree: number of links from a node
  - Diameter: max number of links crossed between nodes
  - Average distance: number of hops to random destination
  - Broadcast: minimum number of links that separate the network into two halves (worst case)

- **Warning:** these three-dimensional drawings must be mapped onto chips and boards, which are essentially two-dimensional media
  - Elegant when sketched on the blackboard may look awkward when constructed from chips, cables, boards, and boxes (largely 2D)

- **Networks should not be interesting!**

Important Topologies

<table>
<thead>
<tr>
<th>Type</th>
<th>Degree Diameter</th>
<th>Ave Dist</th>
<th>Ave Diam</th>
<th>Ave Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D mesh</td>
<td>&lt;=2</td>
<td>N+1</td>
<td>N/2</td>
<td>N+1</td>
</tr>
<tr>
<td>3D mesh</td>
<td>&lt;=4</td>
<td>2(N^1/2-1)</td>
<td>2N^1/2</td>
<td>3N^1/2</td>
</tr>
<tr>
<td>3D mesh</td>
<td>&lt;=6</td>
<td>3(N^1/3-1)</td>
<td>3N^1/3</td>
<td>2N^2/3</td>
</tr>
<tr>
<td>nD mesh</td>
<td>&lt;=2n</td>
<td>n(N^1/2-1)</td>
<td>nN^1/2</td>
<td>(n^1/2-1)^n</td>
</tr>
<tr>
<td>Ring</td>
<td>2</td>
<td>N/2</td>
<td>N/2</td>
<td>2</td>
</tr>
<tr>
<td>2D torus</td>
<td>6</td>
<td>N^1/2</td>
<td>N^1/2</td>
<td>2N^1/2</td>
</tr>
<tr>
<td>3D torus</td>
<td>12</td>
<td>N^1/3</td>
<td>N^1/3</td>
<td>3N^1/3</td>
</tr>
<tr>
<td>k-ary n-cube</td>
<td>2k</td>
<td>N^1/n</td>
<td>N^1/n</td>
<td>2k</td>
</tr>
<tr>
<td>Hypercube n</td>
<td>n^1/Log(n)</td>
<td>n^1/Log(n)</td>
<td>N^1/Log(n)</td>
<td>3N^1/n-1</td>
</tr>
<tr>
<td>Hypercube 2^n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Topologies (cont)

<table>
<thead>
<tr>
<th>Type</th>
<th>Degree</th>
<th>Diameter</th>
<th>Ave Dist</th>
<th>Bisection</th>
<th>Ave D</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Tree</td>
<td>3</td>
<td>$2 \log_2 N$</td>
<td>$2 \log_2 N$</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>4D Tree</td>
<td>4</td>
<td>$2 \log_4 N$</td>
<td>$2 \log_4 N$</td>
<td>1.618</td>
<td>19.2</td>
</tr>
<tr>
<td>kD</td>
<td>k-1</td>
<td>$\log_2 N$</td>
<td>$\log_2 N$</td>
<td>1</td>
<td>20</td>
</tr>
</tbody>
</table>

Multistage: nodes at ends, switches in middle

- All paths equal length
- Unique path from any input to any output
- Conflicts that try to avoid
- Don’t want algorithm to have to know paths

Example MPP Networks

<table>
<thead>
<tr>
<th>Name</th>
<th>Number</th>
<th>Topology</th>
<th>Bits</th>
<th>Clock</th>
<th>Link</th>
<th>Bisection</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>nCube/ten</td>
<td>1-1024</td>
<td>10-cube</td>
<td>1</td>
<td>10 MHz</td>
<td>1.2</td>
<td>640</td>
<td>1987</td>
</tr>
<tr>
<td>iPSC/2</td>
<td>16-128</td>
<td>7-cube</td>
<td>1</td>
<td>16 MHz</td>
<td>2</td>
<td>345</td>
<td>1988</td>
</tr>
<tr>
<td>MP-1216</td>
<td>32-512</td>
<td>2D grid</td>
<td>1</td>
<td>25 MHz</td>
<td>3</td>
<td>1,300</td>
<td>1989</td>
</tr>
<tr>
<td>Delta</td>
<td>540</td>
<td>2D grid</td>
<td>16</td>
<td>40 MHz</td>
<td>40</td>
<td>640</td>
<td>1991</td>
</tr>
<tr>
<td>CM-5</td>
<td>32-2048</td>
<td>fat tree</td>
<td>4</td>
<td>40 MHz</td>
<td>30</td>
<td>10,240</td>
<td>1991</td>
</tr>
<tr>
<td>CS-2</td>
<td>32-1024</td>
<td>fat tree</td>
<td>8</td>
<td>70 MHz</td>
<td>50</td>
<td>50,000</td>
<td>1992</td>
</tr>
<tr>
<td>Paragon</td>
<td>4-1024</td>
<td>2D grid</td>
<td>16</td>
<td>100 MHz</td>
<td>200</td>
<td>6,400</td>
<td>1992</td>
</tr>
<tr>
<td>T3D</td>
<td>16-1024</td>
<td>3D Torus</td>
<td>16</td>
<td>150 MHz</td>
<td>300</td>
<td>13,200</td>
<td>1993</td>
</tr>
</tbody>
</table>

No standard MPP topology!
Like everything in architecture, it’s a cost/benefit trade-off.

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
- HW and SW Interface to computer affects overhead, latency, bandwidth
- Topologies: many to chose from, but (SW) overheads make them look alike; cost issues in topologies, not algorithms

Connection-Based vs. Connectionless

- Telephone: operator sets up connection between the caller and the receiver
  - Once the connection is established, conversation can continue for hours
  - Share transmission lines over long distances by using switches to multiplex several conversations on the same lines
  - “Time division multiplexing” divides B/W transmission line into fixed number of slots, with each slot assigned to a conversation
  - Problem: lines busy based on number of conversations, not amount of information sent
  - Mother’s Day
- Advantage: reserved bandwidth
  - Quality of Service guarantees are easy

Connection-Based vs. Connectionless

- Connectionless: every package of information must have an address $\Rightarrow$ packets
  - Each package is routed to its destination by looking at its address
  - Analogy, the postal system (sending a letter)
  - “Split phase buses” are sending packets
Routing Messages

- **Shared Media**
  - Broadcast to everyone (Ethernet)

- **Switched Media needs real routing**, Options:
  - **Source-based routing**: message specifies path to the destination (changes of direction)
  - **Virtual Circuit**: circuit established from source to destination, message picks the circuit to follow
  - **Destination-based routing**: message specifies destination, switch must pick the path
    - deterministic: always follow same path
    - **adaptive**: pick different paths to avoid congestion, failures
    - **Randomized routing**: pick between several good paths to balance network load
  - Which schemes have messages arriving in order?

Deterministic Routing Examples

- **mesh**: dimension-order routing
  - \((x_1, y_1) \rightarrow (x_2, y_2)\)
  - first \(x\) then \(y\)

- **hypercube**: edge-cube routing
  - \(X = x_0 \times x_1 \times x_2 \ldots \times x_n\)
  - \(Y = y_0 \times y_1 \times y_2 \ldots \times y_n\)
  - \(R = X \ xor \ Y\)
  - Traverse dimensions of differing address in order

- **tree**: common ancestor

- Deadlock free?
  - Can we create a cycle from messages in transit?

Store and Forward vs. Cut-Through

- **Store-and-forward policy**: each switch waits for the full packet to arrive in switch before sending to the next switch (good for WAN)

- **Cut-through routing or worm hole routing**: switch examines the header, decides where to send the message, and then starts forwarding it immediately
  - In **worm hole routing**, when head of message is blocked, message stays strung out over the network, potentially blocking other messages (needs only buffer the piece of the packet that is sent between switches). CM-5 uses it, with each switch buffer being 4 bits per port.
  - **Cut through routing** lets the tail continue when head is blocked, accordioning the whole message into a single switch. (Requires a buffer large enough to hold the largest packet).

Congestion Control

- Packet switched networks do not reserve bandwidth; this leads to **contention** (connection based limits input)

- **Solution**: prevent packets from entering until contention is reduced (e.g., freeway on-ramp metering lights)

- **Options**:
  - **Packet discarding**: If packet arrives at switch and no room in buffer, packet is discarded (e.g., UDP)
  - **Flow control**: between pairs of receivers and senders; use feedback to tell sender when allowed to send next packet
  - **Back-pressure**: separate wires to tell to stop
  - **Window**: give original sender right to send if packets before getting permission to send more; overlaps latency of interconnection with overhead to send & receive packet (e.g., TCP), adjustable window
  - **Choke packets**: aka “rate-based”; Each packet received by busy switch in warning state sent back to the source via choke packet. Source reduces traffic to that destination by a fixed % (e.g., ATM)

Practical Issues for Interconnection Networks

- **Standardization advantages**:
  - low cost (components used repeatedly)
  - stability (many suppliers to choose from)

- **Standardization disadvantages**:
  - **Time for committees to agree**
  - **ATM packet size??????**
  - When to standardize?
    - Before anything built?
    - Committee does design?
  - Too early suppresses innovation

- **Perfect interconnect vs. Fault Tolerant?**
  - Will SW crash on single node prevent communication? (MPP typically assume perfect)

- **Reliability (vs. availability) of interconnect**
**Practical Issues**

<table>
<thead>
<tr>
<th>Interconnection</th>
<th>MPP</th>
<th>LAN</th>
<th>WAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>CM-5</td>
<td>Ethernet</td>
<td>ATM</td>
</tr>
<tr>
<td>Standard</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Fault Tolerance</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Hot Insert</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- Standards: required for WAN, LAN!
- Fault Tolerance: Can nodes fail and still deliver messages to other nodes? required for WAN, LAN!
- Hot Insert: If the interconnection can survive a failure, can it also continue operation while a new node is added to the interconnection? required for WAN, LAN!

**Cross-Cutting Issues for Networking**

- Efficient Interface to Memory Hierarchy vs. to Network
  - SPEC ratings => fast to memory hierarchy
  - Writes go via write buffer, reads via L1 and L2 caches
- Example: 40 MHz SPARCstation(SS)-2 vs 50 MHz SS-20, no L2 vs 50 MHz SS-20 with L2$ I/O bus latency: different generations
  - SS-2: combined memory, I/O bus => 200 ns
  - SS-20, no L2$: 2 busses +300ns => 500ns
  - SS-20, w L2$: cache miss+500ns => 1000ns

**Protocols: HW/SW Interface**

- Internetworking: allows computers on independent and incompatible networks to communicate reliably and efficiently;
  - Enabling technologies: SW standards that allow reliable communications without reliable networks
  - Hierarchy of SW layers, giving each layer responsibility for portion of overall communications task, called protocol families or protocol suites
- Transmission Control Protocol/Internet Protocol (TCP/IP)
  - This protocol family is the basis of the Internet
  - IP makes best effort to deliver; TCP guarantees delivery
  - TCP/IP used even when communicating locally: NFS uses IP even though communicating across homogeneous LAN

**Protocol**

- Key to protocol families is that communication occurs logically at the same level of the protocol, called peer-to-peer, but is implemented via services at the lower level
- Danger is each level increases latency if implemented as hierarchy (e.g., multiple check sums)

**TCP/IP packet**

- Application sends message
- TCP breaks into 64KB segments, adds 20B header
- IP adds 20B header, sends to network
- If Ethernet, broken into 1500B packets with headers, trailers
- Header, trailers have length field, destination, window number, version, ...

**Example Networks**

- Ethernet: shared media 10 Mbit/s proposed in 1978, carrier sensing with exponential backoff on collision detection
- 15 years with no improvement; higher BW?
- Multiple Ethernets with devices to allow Ethernets to operate in parallel!
- 10 Mbit Ethernet successors?
  - FDDI: shared media (too late)
  - ATM (too late?)
  - Switched Ethernet
  - 100 Mbit Ethernet (Fast Ethernet)
  - Gigabit Ethernet
Connecting Networks

- **Bridges**: connect LANs together, passing traffic from one side to another depending on the addresses in the packet.
  - operate at the Ethernet protocol level
  - usually simpler and cheaper than routers (no decision)

- **Routers or Gateways**: these devices connect LANs to WANs or WANs to WANs and resolve incompatible addressing.
  - Generally slower than bridges, they operate at the internetworking protocol (IP) level
  - Routers divide the interconnect into separate smaller subnets, which simplifies manageability and improves security

- Cisco is a major supplier; basically special purpose computers

---

**Example Networks**

<table>
<thead>
<tr>
<th>MPP</th>
<th>LAN</th>
<th>WAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM SP-2</td>
<td>100 Mb Ethernet</td>
<td>ATM</td>
</tr>
</tbody>
</table>

- **Length (meters)**: 10
- **Number data lines**: 8
- **Switch?**: Yes
- **Nodes (N)**: 5512
- **Material**: copper
- **Bisection BW (Mbit/s)**: 320
- **Peak Link BW (Mbit/s)**: 284
- **Measured Link BW**: 80

---

**Example Networks (cont’d)**

<table>
<thead>
<tr>
<th>MPP</th>
<th>LAN</th>
<th>WAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM SP-2</td>
<td>100 Mb Ethernet</td>
<td>ATM</td>
</tr>
</tbody>
</table>

- **Latency (usecs)**: 1
- **Send-Receive Ovhd (usecs)**: 39
- **Topology**: Fat tree
- **Connectionless**: Yes
- **Store & Forward**: No
- **Congestion Control**: Carrier Sense
- **Standard**: Yes
- **Fault Tolerance**: Yes

---

**Examples: Interface to Processor**

<table>
<thead>
<tr>
<th>Further I/O CPU</th>
<th>10 Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun &quot;SNA&quot; (ATM)</td>
<td>Mylron (Local) (Myrinet)</td>
</tr>
</tbody>
</table>

**Packet Formats**

- **Fields**: Destination, Checksum(C), Length(L), Type(T)
- **Data/Header Sizes in bytes**: (4 to 20)/4, (0 to 1500)/26, 48/5

---

**Example Switched LAN Performance**

<table>
<thead>
<tr>
<th>Network Interface</th>
<th>Switch</th>
<th>Link BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD Lance Ethernet</td>
<td>Baynetworks 10 Mb/s</td>
<td>EtherCell 28115</td>
</tr>
<tr>
<td>Fore SBA-200 ATM</td>
<td>Fore AX-200 155 Mb/s</td>
<td></td>
</tr>
<tr>
<td>Myricom Myrinet</td>
<td>Myricom Myrinet 640 Mb/s</td>
<td></td>
</tr>
</tbody>
</table>

- On SPARCstation-20 running Solaris 2.4 OS
- Myrinet is an example of “System Area Network”: networks for a single room or floor: 25m limit
  - shorter => wider, less need for optical
  - short distance => source-based routing => simpler switches
  - Compaq-Tandem/Microsoft also sponsoring SAN, called “ServerNet”
Example Switched LAN Performance (1995)

<table>
<thead>
<tr>
<th>Switch</th>
<th>Switch Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baynetworks</td>
<td>52.0 µsecs</td>
</tr>
<tr>
<td>EtherCell 28115</td>
<td></td>
</tr>
<tr>
<td>Fore ASX-200 ATM</td>
<td>13.0 µsecs</td>
</tr>
<tr>
<td>Myricom Myrinet</td>
<td>0.5 µsecs</td>
</tr>
</tbody>
</table>


UDP/IP performance

<table>
<thead>
<tr>
<th>Network</th>
<th>UDP/IP roundtrip, N=8B</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bay, EtherCell</td>
<td>1009 µsecs</td>
<td>+2.18*N</td>
</tr>
<tr>
<td>Fore ASX-200 ATM</td>
<td>1285 µsecs</td>
<td>+0.32*N</td>
</tr>
<tr>
<td>Myricom Myrinet</td>
<td>1443 µsecs</td>
<td>+0.36*N</td>
</tr>
</tbody>
</table>

• From simple linear regression for tests from N = 8B to N = 8192B
• Software overhead not tuned for Fore, Myrinet; EtherCell using standard driver for Ethernet

NFS performance

<table>
<thead>
<tr>
<th>Network</th>
<th>Avg. NFS response LinkBW/Ether</th>
<th>UDP/E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bay, EtherCell</td>
<td>14.5 ms</td>
<td>1</td>
</tr>
<tr>
<td>Fore ASX-200 ATM</td>
<td>11.8 ms</td>
<td>15</td>
</tr>
<tr>
<td>Myricom Myrinet</td>
<td>13.3 ms</td>
<td>64</td>
</tr>
</tbody>
</table>

• Last 2 columns show ratios of link bandwidth and UDP roundtrip times for 8B message to Ethernet

Estimated Database performance (1995)

<table>
<thead>
<tr>
<th>Network</th>
<th>Avg. TPS</th>
<th>LinkBW/E.</th>
<th>TCP/E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bay. EtherCell</td>
<td>77 tps</td>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td>Fore ASX-200 ATM</td>
<td>67 tps</td>
<td>15</td>
<td>1.47</td>
</tr>
<tr>
<td>Myricom Myrinet</td>
<td>66 tps</td>
<td>64</td>
<td>1.46</td>
</tr>
</tbody>
</table>

• Number of Transactions per Second (TPS) for DebitCredit Benchmark; front end to server with entire database in main memory (256 MB)
  – Each transaction => 4 messages via TCP/IP
  – DebitCredit Message sizes < 200 bytes
• Last 2 columns show ratios of link bandwidth and TCP/IP roundtrip times for 8B message to Ethernet

Summary: Networking

• Protocols allow heterogeneous networking
  – Protocols allow operation in the presence of failures
  – Internetworking protocols used as LAN protocols
  – Larger overhead for LAN
• Integrated circuit revolutionizing networks as well as processors
  – Switch is a specialized computer

Parallel Computers

• Definition: "A parallel computer is a collection of processing elements that cooperate and communicate to solve large problems fast."
Almasi and Gottlieb, Highly Parallel Computing , 1989
• Questions about parallel computers:
  – How large a collection?
  – How powerful are processing elements?
  – How do they cooperate and communicate?
  – How is data transmitted?
  – What type of interconnection?
  – What are HW and SW primitives for programmer?
  – Does it translate into performance?
Parallel Processors “Religion”

- The dream of computer architects since 1960: replicate processors to add performance vs. design a faster processor
- Led to innovative organization tied to particular programming models since “unprocessors can’t keep going”
  - e.g., unprocessors must stop getting faster due to limit of speed of light: 1975, ..., 1989
  - Borders religious fervor: you must believe!
  - Fervor damped some when 1990s companies went out of business: Thinking Machines, Kendall Square, ...
- Argument instead is the “pull” of opportunity of scalable performance, not the “push” of unprocessor performance plateau

Opportunities: Scientific Computing

- Nearly Unlimited Demand (Grand Challenge):
  - App: Perf (GFLOPS) Memory (GB)
  - 48 hour weather: 0.1 0.1
  - 72 hour weather: 3 1
  - Pharmaceutical design: 100 10
  - Global Change, Genome: 1000 1000
  - (Figures 1-2, page 25, of Culler, Singh, Gupta [CSG97])
- Successes in some real industries:
  - Petroleum: reservoir modeling
  - Automotive: crash simulation, drag analysis, engine
  - Aeronautics: airflow analysis, engine, structural mechanics
  - Pharmaceuticals: molecular modeling
  - Entertainment: full length movies (“A Bug’s Life”)
  - You should all see A Bug’s Life - it’s hilarious

Opportunities: Commercial Computing

- Transaction processing & TPC-C benchmark
- (see Chapter 1, Figure 1-4, page 28 of [CSG97])
- - small scale parallel processors to large scale
- Throughput (Transactions per minute) vs. Time (1996)
- - Speedup: 1 4 8 16 32 64 112
  - IBM RS6000: 735 1438 3119 1.00 1.96 4.24
  - Tandem Himalaya: 3043 6067 12021 20918 1.00 1.99 3.95 6.87
  - IBM performance hit 1=>4, good 4=>8
  - Tandem scales: 112/16 = 7.0
- Others: File servers, electronic CAD simulation (multiple processes), WWW search engines

What level Parallelism?

- Bit level parallelism: 1970 to -1985
  - 4 bits, 8 bit, 16 bit, 32 bit microprocessors
- Instruction level parallelism (ILP): 1985 through today
  - Pipelining
  - Superscalar
  - VLIW
  - Out-of-Order execution
  - Limits to benefits of ILP
- Process Level or Thread level parallelism; mainstream for general purpose computing?
  - Servers are parallel - spawn thread or process for each request
  - Dual-Pentium - multiprocessing

Parallel Framework Abstractions

- Layers:
  - Programming Model:
    - Multiprogramming: lots of jobs, no communication
    - Shared address space: communicate via memory
    - Message passing: send and receive messages
  - Data Parallel: several agents operate on several data sets simultaneously and then exchange information globally and simultaneously (shared or message passing)
  - Communication Abstraction:
    - Shared address space: e.g., load, store, atomic swap
    - Message passing: e.g., send, receive library calls
  - Debate over this topic (ease of programming, scaling)
  => many hardware designs 1:1 programming model

Performance Metrics

- Memory Latency
- Memory Bandwidth
- Scalability
Architecture Below

Physical Setup of Computers
- UMA
  - Uniform Memory Access
  - Bus-based machine
- NUMA
  - Non-Uniform Memory Access
- Clumps
- New research in asymmetric processing units
  - not all processing units are created equal

Logical Addressing
- Multicomputers
  - Processors send messages to pass data
- Shared Memory
  - Each processor can access any memory

Note: Architecture may be decoupled from programming paradigm

Shared Address Model Summary
- Each processor can name every physical location in the machine
- Each process can name all data it shares with other processes
- Data transfer via load and store
- Data size: byte, word, ... or cache blocks
- Uses virtual memory to map virtual to local or remote physical
- Memory hierarchy model applies: now communication moves data to local processor cache (as load moves data from memory to cache)

What's is the final value of c?

Scenario 1: Scenario 2: Scenario 3:

Initially, c = 90

Proc A:
- ld c
- add c,c,1
- st c

Proc B:
- ld c
- add c,c,1
- st c

Scenario 1: Scenario 2: Scenario 3: