Interconnection Networks

Alvin R. Lebeck
CPS 220

Goal: Communication between computers
Warning: Terminology-rich environment
Focus on Networks for Parallel Computing
– today’s System Area Networks exhibit many of the same properties

Terms

Network characterized by

- Topology
  - physical structure of the graph
- Routing Algorithm
  - which paths through network can message flow
- Switching Strategy
  - How data in message traverses its route
  - Circuit Switched vs Packet Switched
- Flow Control
  - When does a packet (or portions of it) move along its route

Organization

- Given topology constructed by linking switches and network interfaces, must deliver packet from node A to node B
- Link: cable with connectors on each end
- Switch: N inputs N outputs (degree N)
- Flit: Minimum # of bits move across link as a single unit
- Packet: unit that requires routing information, some number of flits

Topology

- Structure of the interconnect
- Determines
  - Switch Degree: number of links from a node
  - Diameter: number of links crossed between nodes on maximum shortest path
  - Average distance: number of hops to random destination
  - Bisection: minimum number of links that separate the network into two halves
### Important Topologies

<table>
<thead>
<tr>
<th>Type</th>
<th>Degree</th>
<th>Diameter</th>
<th>Ave Dist</th>
<th>Bisection Diam</th>
<th>Ave D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D mesh</td>
<td>2</td>
<td>N-1</td>
<td>2N/3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2D mesh</td>
<td>4</td>
<td>2(N^1/2-1)</td>
<td>2N^1/2/3</td>
<td>N^1/2</td>
<td>1</td>
</tr>
<tr>
<td>nD mesh (N x k)</td>
<td>2n</td>
<td>n(N^1/n-1)</td>
<td>n(N^1/n)</td>
<td>N^1/n</td>
<td></td>
</tr>
<tr>
<td>Ring</td>
<td>2</td>
<td>N/2</td>
<td>N/4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2D torus</td>
<td>4</td>
<td>N^1/2</td>
<td>N^1/2/2</td>
<td>2N^1/2</td>
<td>1</td>
</tr>
<tr>
<td>hypercube n</td>
<td>n</td>
<td>nLogN</td>
<td>n/2</td>
<td>N/2</td>
<td>1</td>
</tr>
</tbody>
</table>

### Topologies (cont)

<table>
<thead>
<tr>
<th>Type</th>
<th>Degree</th>
<th>Diameter</th>
<th>Ave Dist</th>
<th>Bisection Diam</th>
<th>Ave D</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Tree</td>
<td>3</td>
<td>Slog,N</td>
<td>Slog,N</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>4D Tree</td>
<td>5</td>
<td>Slog,N</td>
<td>Slog,N</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>kD tree</td>
<td>k+1</td>
<td>Log,N</td>
<td>Log,N</td>
<td>N/2</td>
<td>20</td>
</tr>
<tr>
<td>2D butterfly</td>
<td>4</td>
<td>Log,N</td>
<td>Log,N</td>
<td>N/2</td>
<td>20</td>
</tr>
</tbody>
</table>

### Butterfly

- All paths equal length
- Unique path from any input to any output
- Conflicts cause tree saturation

### Benes Network

- Routes all permutations w/o conflict
- Notice similarity to Fat Tree (Fold in half)
- Randomization is major breakthrough

### ABCs of Networks

- **Starting Point**: Send bits between 2 computers

Queue on each end
- Can send both ways (“Bi-directional, Full Duplex”)  
- Rules for communication? “protocol”  
  - Synchronous send
  - Need Request & Response signaling
  - Name for standard group of bits sent: Packet

### A Simple Example

- What is the packet format?
  - Fixed? (for HW Interpretation)
  - Number bytes?

<table>
<thead>
<tr>
<th>Request/Response</th>
<th>Address/Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 bit</td>
<td>32 bits</td>
</tr>
</tbody>
</table>

0: Please send data from Address  
1: Packet contains data corresponding to request

### Questions About Simple Example

- What if more than 2 computers want to communicate?
  - Need node identifier 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, time outs)

- What if multiple processes/machine?
  - Dispatch
  - Queue per process

- Questions such as these lead to more complex protocols and packet formats
General Packet Format

- Header
  - routing and control information
- Payload
  - carries data (non HW specific information)
    - can be further divided (framing, protocol stacks...)
- Error Code
  - generally at tail of packet so it can be generated on the way out

Message v.s. Packet

- A Message may be composed of several packets
- Applications reason about messages
- Network transfers packets
- Small fixed size packets. Problems?
  - Fragmentation and reassembly (SW overhead)
- Variable Size packets. Problems?
  - Congestion

Packet Switched v.s. Circuit Switched

Circuit Switched
- Establish Route then Send Data
- Telephone system

Packet Switched
- Route each packet individually
- Delivery Guarantees
  - Reliable
    - In order, what if not?

Routing

- Store-and-forward
- Cut-through
- Virtual cut-through
- Wormhole

Routing

- Store-and-forward policy: each switch waits for the full packet to arrive in the switch before it is sent on to the next switch

Cut Through

- Cut-through routing: switch examines the header, decides where to send the message, and then starts forwarding it immediately
Virtual Cut-Through

- What to do if output port is blocked?
- Lets the tail continue when the head is blocked, absorbing the whole message into a single switch.
  - Requires a buffer large enough to hold the largest packet.
- Degenerates to store-and-forward with high contention
- Compaq EV7 network

Wormhole

- When the head of the message is blocked the message stays strung out over the network
  - Potentially blocks other messages (needs only buffer the piece of the packet that is sent between switches).
  - CM-5 used it, with each switch buffer being 4 bits per port.
  - Myrinet uses it
- Interaction with Packet Size
- Can cause tree saturation...

Store and Forward vs. Cut-Through

- Advantage
  - Latency reduces from function of:
    - Store and Forward
      - number of intermediate switches times the size of the packet
    - Cut-Through
      - time for 1st part of the packet to negotiate the switches + the packet size ÷ interconnect BW

Switches

- At minimum, must route inputs to outputs

Routing Algorithm

- How do I know where a packet should go?
- Arithmetic
- Source-Based
- Table Lookup
- Adaptive—route based on network state (e.g., contention)

Arithmetic Routing

- For regular topology, simple arithmetic to determine route
- 2D Mesh (Also called NEWS network)
  - packet header contains signed offset to destination
  - switch ++ or -- one field of header (x or y dimension)
  - when x == 0 and y == 0, then at correct processor
- Requires ALU in switch
- Must recompute CRC
Source Based and Table Lookup Routing

Source Based
- Source specifies output port for each switch in route
- Very Simple Switches
  - no control state
  - strip output port off header
- Myrinet uses this

Table Lookup
- Very Small Header, index into table for output port
- Big tables, must be kept up to date...

Deterministic v.s. Adaptive Routing

- Deterministic—follows a pre-specified route
  - mesh: dimension-order routing
    - \((x_1, y_1) \rightarrow (x_2, y_2)\)
    - first \(D_x = x_2 - x_1\)
    - then \(D_y = y_2 - y_1\)
  - hypercube: edge-cube routing
    - \(X = x_0x_1x_2 \ldots x_n\) \(\rightarrow\) \(Y = y_0y_1y_2 \ldots y_n\)
    - \(R = X \text{ xor } Y\)
    - Traverse dimensions of differing address in order
      - tree: common ancestor
  - Adaptive—route determined by contention for output port

Deadlock

- Virtual Channels
  - Not virtual cut-through
  - Add buffers so, flits of wormhole packets can be interleaved
- Up*-Down*
  - Number switches: higher \(=\) farther away from processors
  - route up, make one turn, route down
- Turn Model Routing
  - Restrict order of turns
    - West First
    - North Last
    - Negative First
  - Can increase number of hops

Deadlock Free Routing

- Virtual Channels
  - Not virtual cut-through
  - Add buffers so, flits of wormhole packets can be interleaved
- Up*-Down*
  - Number switches: higher \(=\) farther away from processors
  - route up, make one turn, route down
- Turn Model Routing
  - Restrict order of turns
    - West First
    - North Last
    - Negative First
  - Can increase number of hops

Congestion Control

- Packet switched networks do not reserve bandwidth; this leads to contention
- Solution: prevent packets from entering until contention is reduced (e.g., metering lights)
- Options:
  - End-to-end Flow Control
  - Link-level Flow Control

Link-Level Flow Control

- Packet discarding: If a packet arrives at a switch and there is no room in the buffer, the packet is discarded
  - no communication between switches, requires higher level protocol
- Flow control: between pairs of receivers and senders; use feedback to tell the sender when it is allowed to send the next packet
  - 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 % (ATM Forum)
  - Back-pressure: separate wires to tell to stop
  - high water, low water (stop & go back to source switch) (Myrinet)
  - Window: give the original sender the right to send N packets before getting permission to send more (overlap the latency of the interconnection with the overhead to send and receive a packet)
Link-Level Flow Control

- Transfer single flit when receiver is ready
- Could have long links with many flits in flight

Credit-based (Window) Flow Control

- Receiver gives N credits to sender
  - sender decrements count
  - stops sending if zero
  - receiver sends back credit as it drains its buffer
  - bundle credits to reduce overhead
- Must account for link latency

Water Level

- high water, low water
- stop & go back to source switch (Myrinet)
- can send redundant stop/go

Incoming phits

Stop

Outgoing phits

Case Study Cray T3D

- 1024 switch nodes each connected to 2 processors
- 3D Torus, bidirectional, 300 MB/s
- Link: 16 bits, 8 control bits
- Variable size packet (multiple of 16 bits)
- Logical request & response networks
  - 2 virtual channels each for deadlock
- Stacked dimension routing
- Wormhole for large packets, virtual cut-through for small packets

IBM SP-2 (Vulcan)

- Switch has eight bidirectional 40 MB/s links
- Link: 8 data bits, 1 tag, 1 reverse flow-control
- Flit is 16 bits, phit is 8
- Input FIFO + output FIFO + central buffer 128 8-byte segments