Generic Multiprocessor Architecture

Node: processor(s), memory system, plus communication assist:

- Network interface and communication controller.

- Scalable network.
  - Function of a parallel machine network is to efficiently transfer information from source node to destination node in support of network transactions that realize the programming model.
  - Network performance should scale up as its size is increased.
Cost of Communication

Given amount of comm (inherent or artifactual), goal is to reduce cost

- Cost of communication as seen by process:
  \[ C = f \times (o + l + \frac{n}{B} + t_c - overlap) \]

  - \( f \) = frequency of messages
  - \( o \) = overhead per message (at both ends)
  - \( l \) = network delay per message
  - \( n \) = data sent per message
  - \( B \) = bandwidth along path (determined by network, NI, assist)
  - \( t_c \) = cost induced by contention per message
  - \( overlap \) = amount of latency hidden by overlap with comp. or comm.

  - Portion in parentheses is cost of a message (as seen by processor)
  - That portion, ignoring overlap, is latency of a message
  - Goal: reduce terms in latency and increase overlap
Network Representation & Characteristics

- A parallel machine interconnection network is a graph \( V = \{\text{switches and nodes}\} \) connected by communication channels or links \( C \subseteq V \times V \).

- Each channel has width \( w \) bits and signaling rate \( f = 1/\tau \) (\( \tau \) is clock cycle time).
  - Channel bandwidth \( b = wf \) bits/sec
  - Phit (physical unit) data transferred per cycle (usually channel width \( w \)).
  - Flit - basic unit of flow-control (minimum data unit transferred across a link).

- Number of input (output) channels is switch or node degree.

- Sequence of switches and links followed by a message in the network is a route.

- Routing Distance - number of links or hops on route.

- A network is generally characterized by:
  - Topology.
  - Flow Control Mechanism.
  - Routing Algorithm.
  - Switching Strategy.
Network Characteristics

• Topology:
  – Physical interconnection structure of the network graph:
    • Node Degree: Number of channels per node.
    • Network diameter: Longest minimum routing distance between any two nodes in hops.
    • Average Distance between all pairs of nodes.
    • Bisection width: Minimum number of links whose removal disconnects the graph and cuts it in half.
    • Symmetry: The property that the network looks the same from every node.
    • Homogeneity: Whether all the nodes and links are identical or not.

  – Type of interconnection:
    • Static or Direct Interconnects: Nodes connected directly using static point-to-point links.
    • Dynamic or Indirect Interconnects: Switches are usually used to realize dynamic links between nodes:
      – Each node is connected to specific subset of switches. (e.g. Multistage Interconnection Networks, MINs).
      – Blocking or non-blocking, permutations realized.
    • Shared-, broadcast-, or bus-based connections. (e.g. Ethernet-based).
Network Characteristics

• Routing Algorithm and Functions:
  – The set of paths that messages may follow.
  – Request/message combining capabilities.

• Switching Strategy:
  – Circuit switching vs. packet switching.

• Flow Control Mechanism:
  – When a message or portions of it moves along its route:
    • Store & Forward Routing,
    • Cut-Through or Worm-Hole Routing.
  – What happens when traffic is encountered at a node:
    • Link/Node Contention handling.
    • Deadlock prevention.

• Broadcast and Multicast Capabilities.
• Communication Latency.
• Link bandwidth.
Network Characteristics

- Hardware/software implementation complexity/cost.
- Network throughput: Total number of messages handled by network per unit time.
- Aggregate Network bandwidth: Similar to network throughput but given in total bits/sec.
- Network hot spots: Form in a network when a small number of network nodes/links handle a very large percentage of total network traffic and become saturated.
- Network scalability:
  - The feasibility of increasing network size, determined by:
    - Performance scalability: Relationship between network size in terms of number of nodes and the resulting network performance.
    - Cost scalability: Relationship between network size in terms of number of nodes/links and network cost/complexity.
Network Requirements For Parallel Computing

- Minimum network latency even when approaching network capacity.
- High sustained bandwidth that matches or exceeds the communication requirements for given computational rate.
- High network throughput: Network should support as many concurrent transfers as possible.
- Low Protocol overhead.
- Minimum network cost.
- Maximum Network Scalability: Network performance should scale up with network size.
Communication Network Performance:

**Network Latency**

Unloaded Network Latency = routing delay + channel occupancy

Time to transfer $n$ bytes from source to destination:

$$\text{Time}(n)_{s-d} = \text{overhead} + \text{routing delay}$$

$$+ \text{channel occupancy} + \text{contention delay}$$

channel occupancy $= (n + n_e) / b$

$b = \text{channel bandwidth, bytes/sec}$

$n = \text{payload size}$

$n_e = \text{packet envelope: header, trailer.}$
Flow Control Mechanisms: Store&Forward Vs. Cut-Through Routing

Store & Forward Routing

Unloaded network latency for n byte packet:

\[ h\left(\frac{n}{b} + \Delta\right) \text{ vs } n/b + h \Delta \]

\[ h = \text{distance in hops} \quad \Delta = \text{switch delay} \]
Communication Network Performance: Network Latency

- For an unloaded network (no contention delay) the network latency to transfer an n byte packet (including packet envelope) across the network:
  
  Unloaded Network Latency = routing delay + channel occupancy

- For store-and-forward routing:
  - Unloaded Network Latency = $T_{sf}(n, h) = h(n/b + \Delta)$

- For cut-through routing:
  - Unloaded Network Latency = $T_{ct}(n, h) = n/b + h\Delta$

  $h =$ distance in hops  \hspace{1cm} $\Delta =$ switch delay
Reducing Network Latency

• Use cut-through routing:
  – Unloaded Network Latency = $T_{sf}(n, h) = h(\frac{n}{b} + \Delta)$

• Reduce number of hops $h$ in route:
  – Map communication patterns to network topology
    e.g. nearest-neighbor on mesh and ring; all-to-all
  • Applicable to networks with static or direct point-to-point interconnects: Ideally network topology matches problem communication patterns.

• Increase link bandwidth $b$.

• Reduce switch routing delay $\Delta$. 
Available Bandwidth

- Factors affecting local bandwidth available to a single node:
  - Accounting for Packet density: \( b \times n / (n + n_e) \)
  - Also Accounting for Routing delay: \( b \times n / (n + n_e + w\Delta) \)
  - Contention:
    - At endpoints.
    - Within the network.

- Factors affecting throughput or Aggregate bandwidth:
  - Network bisection bandwidth:
    - Sum of bandwidth of smallest set of links when removed partition the network into two unconnected networks of equal size.
  - Total bandwidth of all the channels: \( C_b \) bytes/sec, \( C_w \) bits per cycle or \( C \) phits per cycle.
  - Suppose \( N \) hosts each issue a packet every \( M \) cycles with average routing distance \( h \) and average distribution:
    - Each message occupies \( h \) channels for \( t = n/w \) cycles
    - Total network load = \( Nh_t / M \) phits per cycle.
    - Average Link utilization = Total network load / Total bandwidth
    - Average Link utilization: \( \rho = MC/Nh_t < 1 \)
Two packets trying to use the same link at same time.
  - May be caused by limited available buffering.
  - Possible resolutions:
    - Increased buffer space.
    - Drop one or more packets.
    - Use an alternative route (requires an adaptive routing algorithm or a better static routing to distribute load more evenly).

Most networks used in parallel machines block in place
  - Link-level flow control.
  - Back pressure to the source to slow down flow of data.

Closed system: Offered load depends on delivered.
Network Saturation

Indications of Network Saturation

- High queuing
- Delays

Latency

Delivered Bandwidth

Delivered Bandwidth

Latency

Saturation

Link utilization = 1

Saturation

Offered Bandwidth

Saturation

Delivered Bandwidth
Deadlock In Store & Forward Networks

Deadlock prevention:
Multiple virtual channels mapped onto one physical channel.
Sample Static Network Topologies

- Linear
- Ring
- 2D Mesh
- Hypercube
- Binary Tree
- Fat Binary Tree
- Fully Connected
Static Point-to-point Connection Network Topologies

- Direct point-to-point links are used.
- Suitable for predictable communication patterns matching topology.

Fully Connected Network: Every node is connected to all other nodes using $N-1$ direct links

$$\frac{N(N-1)}{2} \text{ Links} \rightarrow O(N^2) \text{ complexity}$$

Node Degree: $N-1$

Diameter = 1

Average Distance = 1

Bisection Width = $(N/2)^2$

Linear Array:

$N-1$ Links $\rightarrow O(N)$ complexity

Node Degree: 1-2

Diameter = $N-1$

Average Distance = $2/3N$

Bisection Width = 1

Ring:

$N$ Links $\rightarrow O(N)$ complexity

Node Degree: 2

Diameter = $N/2$

Average Distance = $1/3N$

Bisection Width = 2

Examples: Token-Ring, FDDI, SCI, FiberChannel Arbitrated Loop, KSR1

Route A $\rightarrow$ B given by relative address $R = B-A$
Static Network Topologies Examples:
Multidimensional Meshes and Tori

$d$-dimensional array or mesh:
- $N = k_{d-1} \times \ldots \times k_0$ nodes
- described by $d$-vector of coordinates $(i_{d-1}, \ldots, i_0)$
- Where $0 \leq i_j \leq k_j$ for $0 \leq j \leq d-1$

$d$-dimensional $k$-ary mesh: $N = k^d$
- $k = \sqrt[d]{N}$
- described by $d$-vector of radix $k$ coordinate.
- Diameter = $d(k-1)$

$d$-dimensional $k$-ary torus (or $k$-ary $d$-cube):
- Edges wrap around, every node has degree $2d$ and connected to nodes that differ by one (mod $k$) in every dimension.
Multidimensional Meshes and Tori Properties

Routing:
- Relative distance: \( R = (b_{d-1} - a_{d-1}, \ldots, b_0 - a_0) \)
- Traverse \( r_i = b_i - a_i \) hops in each dimension.
- Dimension-order routing.

Average Distance:
- \( d \times 2k/3 \) for mesh.
- \( dk/2 \) for cube.

Degree:
- \( d \) to \( 2d \) for mesh.
- \( 2d \) for cube.

Bisection bandwidth:
- \( k^{d-1} \) bi-directional links when \( k \) is even.

- Physical layout?
  - 2D in \( O(N) \) space.
Static Connection
Networks Examples:
2D Mesh
(2-dimensional $k$-ary mesh)

For an $k \times k$ 2D Mesh:

- Node Degree: 2-4
- Network diameter: $2(k-1)$
- No of links: $2N - 2k$
- Bisection Width: $k$
- Where $k = \sqrt{N}$
- Example: 1824 node Intel Paragon: 16 x 114 2D mesh
Static Connection Networks Examples: Hypercubes

- Also called binary $n$-cubes.
- Dimension $= n = \log_2 N$
- Number of nodes $= N = 2^n$
- Diameter: $O(\log_2 N)$ hops
- Good bisection BW: $N/2$
- Complexity:
  - Number of links: $N(\log_2 N)/2$
  - Node degree is $n = \log_2 N$
Message Routing Functions Example
Dimension-order Routing

Network Topology:
3-dimensional static-link hypercube
Nodes denoted by \( C_2 C_1 C_0 \)

Routing by least significant bit \( C_0 \)

Routing by middle bit \( C_1 \)

Routing by most significant bit \( C_2 \)
Static Connection Networks Examples: Trees

- Diameter and average distance are logarithmic.
  - $k$-ary tree, height $d = \log_k N$
  - Address specified $d$-vector of radix $k$ coordinates describing path down from root.
- Fixed degree $k$.
- Route up to common ancestor and down:
  - $R = B \ XOR \ A$
  - Let $i$ be position of most significant 1 in $R$, route up $i+1$ levels
  - Down in direction given by low $i+1$ bits of $B$
- H-tree space is $O(N)$ with $O(\sqrt{N})$ long wires.
- Low Bisection BW = 1
Static Connection Networks Examples: Fat-Trees

- “Fatter” higher bandwidth links (more connections in reality) as you go up, so bisection BW scales with number of nodes N.
- Example: Network topology used in Thinking Machine CM-5
Embedding k-ary d-cubes In Two Dimensions

- Embed multiple logical dimension in one physical dimension using long interconnections.
Embedding A Binary Tree Onto A 2D Mesh

\( \text{A} = \) Additional nodes added to form the tree
Embedding A Ring Onto A 2D Torus
Dynamic Connection Networks

• Switches are usually used to implement connection paths or virtual circuits between nodes instead of fixed point-to-point connections.

• Dynamic connections are established based on communication demands.

• Such networks include:
  – Bus systems.
  – Multi-stage Interconnection Networks (MINs):
    • Omega Network.
    • Baseline Network
    • Butterfly Network, etc.
  – Crossbar switch networks.
Dynamic Networks Definitions

- **Permutation networks:** Can provide any one-to-one mapping between sources and destinations.

- **Strictly non-blocking:** Any attempt to create a valid connection succeeds. These include Clos networks and the crossbar.

- **Wide Sense non-blocking:** In these networks any connection succeeds if a careful routing algorithm is followed. The Benes network is the prime example of this class.

- **Rearrangeably non-blocking:** Any attempt to create a valid connection eventually succeeds, but some existing links may need to be rerouted to accommodate the new connection. Batcher's bitonic sorting network is one example.

- **Blocking:** Once certain connections are established it may be impossible to create other specific connections. The Banyan and Omega networks are examples of this class.

- **Single-Stage networks:** Crossbar switches are single-stage, strictly non-blocking, and can implement not only the N! permutations, but also the $N^N$ combinations of non-overlapping broadcast.
Dynamic Network Building Blocks:

Crossbar-Based Switches

Cross-bar

Input Buffer

Control
Routing, Scheduling

Output Buffer

Transmitter

Input Ports

Receiver

Output Ports
Switch Components

• Output ports:
  – Transmitter (typically drives clock and data).

• Input ports:
  – Synchronizer aligns data signal with local clock domain.
  – FIFO buffer.

• Crossbar:
  – Switch fabric connecting each input to any output.
  – Feasible degree limited by area or pinout, $O(n^2)$ complexity.

• Buffering (input and/or output).

• Control logic:
  – Complexity depends on routing logic and scheduling algorithm.
  – Determine output port for each incoming packet.
  – Arbitrate among inputs directed at same output.
## Switch Size And Legitimate States

<table>
<thead>
<tr>
<th>Switch Size</th>
<th>Legitimate States</th>
<th>Permutation Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 \times 2$</td>
<td>$4$</td>
<td>$2$</td>
</tr>
<tr>
<td>$4 \times 4$</td>
<td>$256$</td>
<td>$24$</td>
</tr>
<tr>
<td>$8 \times 8$</td>
<td>$16,777,216$</td>
<td>$40,320$</td>
</tr>
<tr>
<td>$n \times n$</td>
<td>$n^n$</td>
<td>$n!$</td>
</tr>
</tbody>
</table>
Permutations

• For \( n \) objects there are \( n! \) permutations by which the \( n \) objects can be reordered.
• The set of all permutations form a permutation group with respect to a composition operation.
• One can use cycle notation to specify a permutation function. For Example:

The permutation \( \pi = (a, b, c)(d, e) \) stands for the bijection mapping:

\[ a \to b, \ b \to c, \ c \to a, \ d \to e, \ e \to d \]

in a circular fashion. The cycle \((a, b, c)\) has a period of 3 and the cycle \((d, e)\) has a period of 2. Combining the two cycles, the permutation \(\pi\) has a cycle period of \(2 \times 3 = 6\). If one applies the permutation \(\pi\) six times, the identity mapping \(I = (a)(b)(c)(d)(e)\) is obtained.
Perfect Shuffle

- Perfect shuffle is a special permutation function suggested by Harold Stone (1971) for parallel processing applications.
- Obtained by rotating the binary address of an one position left.
- The perfect shuffle and its inverse for 8 objects are shown here:

```
Perfect Shuffle

000 --> 000
001 --> 001
010 --> 010
011 --> 011
100 --> 100
101 --> 101
110 --> 110
111 --> 111

Inverse Perfect Shuffle

000 --> 000
001 --> 001
010 --> 010
011 --> 011
100 --> 100
101 --> 101
110 --> 110
111 --> 111
```
Multi-Stage Networks: The Omega Network

- In the Omega network, perfect shuffle is used as an inter-stage connection pattern for all $\log_2 N$ stages.
- Routing is simply a matter of using the destination's address bits to set switches at each stage.
- The Omega network is a single-path network: There is just one path between an input and an output.
- It is equivalent to the Banyan, Staran Flip Network, Shuffle Exchange Network, and many others that have been proposed.
- The Omega can only implement $N^{N/2}$ of the $N!$ permutations between inputs and outputs, so it is possible to have permutations that cannot be provided (i.e. paths that can be blocked).
  - For $N = 8$, there are $8^4/8! = 4096/40320 = 0.1016 = 10.16\%$ of the permutations that can be implemented.
- It can take $\log_2 N$ passes of reconfiguration to provide all links. Because there are $\log_2 N$ stages, the worst case time to provide all desired connections can be $(\log_2 N)^2$. 
Multi-Stage Networks:
The Omega Network

Fig 2.24 page 92

Kai Hwang ref.

See handout
MINs Example: Baseline Network

Fig 2.25  page 93

Kai Hwang ref.

See handout
MINs Example: Butterfly Network

- Complexity: \( \frac{N}{2} \times \log N \)
- Exactly one route from any source to any destination node.
- \( R = A \ XOR \ B \), at level \( i \) use ‘straight’ edge if \( r_i = 0 \), otherwise cross edge
- Bisection \( \frac{N}{2} \)
- Diameter \( \log N \)
Relationship Between Butterfly Network & Hypercubes

- The connection patterns in the two networks are isomorphic.
  - Except that Butterfly always takes $\log_2 n$ steps.
Traditional Network Scaling: Latency(P)

- Assumes equal channel width:
  - Independent of node count or dimension.
  - Dominated by average distance.

Message transmission time (single channel occupancy)
Unloaded Latency with Equal Bisection Width

- N-node hypercube has $N$ bisection links.
- 2d torus has $2N^{1/2}$
- Fixed bisection $\Rightarrow w(d) = N^{1/d}/2 = k/2$
- not shown: 1 M nodes, $d=2$ has $w=512$ And avg. 1023 hops.

$n = 40$ bytes, $\Delta = 2$
Summary of Static Network Characteristics

Table 2.2  page 88

Kai Hwang ref.

See handout
Summary of Dynamic Network Characteristics

Table 2.4  page 95
Kai Hwang ref.
See handout
Example Networks: Cray MPPs

- **T3D**: Short, Wide, Synchronous (300 MB/s).
  - 3D bidirectional torus up to 1024 nodes, dimension order, virtual cut-through, packet switched routing.
  - 24 bits: 16 data, 4 control, 4 reverse direction flow control
  - Single 150 MHz clock (including processor).
  - flit = phit = 16 bits.
  - Two control bits identify flit type (idle and framing).
    - No-info, routing tag, packet, end-of-packet.

- **T3E**: long, wide, asynchronous (500 MB/s)
  - 14 bits, 375 MHz
  - flit = 5 phits = 70 bits
    - 64 bits data + 6 control
  - Switches operate at 75 MHz.
  - Framed into 1-word and 8-word read/write request packets.
## Parallel Machine Examples

<table>
<thead>
<tr>
<th>Machine</th>
<th>Topology</th>
<th>Cycle Time (ns)</th>
<th>Channel Width (bits)</th>
<th>Routing Delay (cycles)</th>
<th>Flit (data bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>nCUBE/2</td>
<td>Hypercube</td>
<td>25</td>
<td>1</td>
<td>40</td>
<td>32</td>
</tr>
<tr>
<td>TMC CM-5</td>
<td>Fat-Tree</td>
<td>25</td>
<td>4</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>IBM SP-2</td>
<td>Banyan</td>
<td>25</td>
<td>8</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>Intel Paragon</td>
<td>2D Mesh</td>
<td>11.5</td>
<td>16</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>Meiko CS-2</td>
<td>Fat-Tree</td>
<td>20</td>
<td>8</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>CRAY T3D</td>
<td>3D Torus</td>
<td>6.67</td>
<td>16</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>DASH</td>
<td>Torus</td>
<td>30</td>
<td>16</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>J-Machine</td>
<td>3D Mesh</td>
<td>31</td>
<td>8</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Monsoon</td>
<td>Butterfly</td>
<td>20</td>
<td>16</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>SGI Origin</td>
<td>Hypercube</td>
<td>2.5</td>
<td>20</td>
<td>16</td>
<td>160</td>
</tr>
<tr>
<td>Myricom</td>
<td>Arbitrary</td>
<td>6.25</td>
<td>16</td>
<td>50</td>
<td>16</td>
</tr>
</tbody>
</table>