Network Properties, Scalability and Requirements For Parallel Processing

Scalable Parallel Performance: Continue to achieve good parallel performance "speedup" as the sizes of the system/problem are increased. Scalability/characteristics of the parallel system network play an important role in determining performance scalability of the parallel 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.**
    - Latency grows slowly with network size $N$. e.g $O(\log_2 N)$ vs. $O(N^2)$
    - Total available bandwidth scales up with network size. e.g $O(N)$
  - **Network cost/complexity should grow slowly in terms of network size.**
    - e.g. $O(N\log_2 N)$ as opposed to $O(N^2)$

Two Aspects of Network Scalability: **Performance and Cost/Complexity**

1. **Network performance scalability**
   - i.e. $O(\log_2 N)$ vs. $O(N^2)$

2. **Network cost/complexity scalability**
   - e.g. $O(N\log_2 N)$ vs. $O(N^2)$

(PP Chapter 1.3, PCA Chapter 10)

$N = $ Size of Network
Network Requirements For Parallel Computing

1. **Low network latency** even when approaching network capacity.
2. **High sustained bandwidth** that matches or exceeds the communication requirements for given computational rate.
3. **High network throughput**: Network should support as many concurrent transfers as possible.
4. **Low Protocol overhead**.
5. **Cost/complexity and performance Scalable**:
   - **Cost/Complexity Scalability**: Minimum network cost/complexity increase as network size increases.
     - In terms of number of links/switches, node degree etc.
   - **Performance Scalability**: Network performance should scale up with network size.
     - Latency grows slowly with network size.
     - Total available bandwidth scales up with network size.

For A given network Size

To reduce communication overheads

As network Size Increases

Scalable network

Two Aspects of Network Scalability: **Performance and Complexity**
Cost of Communication

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

- Cost of communication as seen by process:

\[ C = f \times \left( o + l + \frac{n}{B} + t_c - \text{overlap} \right) \]

- \( f \) = frequency of messages
- \( o \) = overhead per message (at both ends)
- \( l \) = network delay per message
- \( n \) = data sent for per message
- \( B \) = bandwidth along path (determined by network, NI, assist)
- \( t_c \) = cost induced by contention per message
- \( \text{overlap} \) = 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

From lecture 6
Network Representation & Characteristics

- A parallel machine interconnection network is a graph \( V = \{\text{switches or processing nodes}\} \) 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 channels per node or switch 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 \( h \) on route from source to destination.
- A network is generally characterized by:
  - Type of interconnection. Static (point-to-point) or Dynamic
  - Topology. Network node connectivity/interconnection structure of the network graph
  - Routing Algorithm. Deterministic (static) or Adaptive (dynamic)
  - Switching Strategy. Packet or Circuit Switching
  - Flow Control Mechanism. Store & Forward (SF) or Cut-Through (CT)
Network Characteristics

• **Type of interconnection:**
  1. **Static, Direct Dedicated (or point-to-point) Interconnects:**
     • Nodes connected directly using static point-to-point links.
     • Such networks include:
       - Fully connected networks, Rings, Meshes, Hypercubes etc.
  2. **Dynamic or Indirect Interconnects:**
     • Switches are usually used to realize dynamic links (paths or virtual circuits) between nodes instead of fixed point-to-point connections.
     • Each node is connected to specific subset of switches.
     • Dynamic connections are usually established by configuring switches based on communication demands.
     • Such networks include:
       - Shared-, broadcast-, or bus-based connections. (e.g. Ethernet-based).
       - Single-stage Crossbar switch networks. One large switch
       - Multi-stage Interconnection Networks (MINs) including:
         • Omega Network, Baseline Network, Butterfly Network, etc.
Network Characteristics

- **Network Topology:**
  - **Node connectivity:** Which nodes are directly connected
  - **Total number of links needed:** Impacts network cost/total bandwidth
  - **Node Degree:** Number of channels per node.
  - **Network diameter:** Minimum routing distance in links or hops between the farthest two nodes.
  - **Average Distance:** in hops between all pairs of nodes.
  - **Bisection width:** Minimum number of links whose removal disconnects the network graph and cuts it into approximately two equal halves.
    - Related: **Bisection Bandwidth** = Bisection width x link bandwidth
  - **Symmetry:** The property that the network looks the same from every node.
  - **Homogeneity:** Whether all the nodes and links are identical or not.

Hop = link = channel in route
Network Topology and Requirements for Parallel Processing

1. **For Cost/Complexity Scalability:** The total number of links, node degree and size/number of switches used should grow slowly as the size of the network is increased.

2. **For Low network latency:** Small network diameter, average distance are desirable (for a given network size).

3. **For Latency Scalability:** The network diameter, average distance should grow slowly as the size of the network is increased.

4. **For Bandwidth Scalability:** The total number of links should increase in proportion to network size.

5. **To support as many concurrent transfers as possible (High network throughput):** A high bisection width is desirable and should increase proportional to network size.
   - Needed to reduce network contention and hot spots.

More on this later in the lecture
Network Characteristics

- **Routing Algorithm and Functions:**
  - The set of paths that messages may follow.
  - **Deterministic (static) Routing:** The route taken by a message determined by source and destination regardless of other traffic in the network.
  - **Adaptive (dynamic) Routing:** One of multiple routes from source to destination selected to account for other traffic to reduce node/link contention.

- **Switching Strategy:**
  - Circuit switching vs. packet switching.

- **Flow Control Mechanism:**
  - When a message or portions of it moves along its route:
    1. **Store & Forward (SF) Routing.**
    2. **Cut-Through (CT) or Worm-Hole Routing.** (usually uses circuit switching)
  - What happens when traffic is encountered at a node:
    - Link/Node Contention handling.
    - Deadlock prevention.

- **Broadcast and multicast capabilities.**
- **Switch routing delay.**
- **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 bytes/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 (average latency, aggregate network bandwidth).
  - **Cost scalability**: Relationship between network size in terms of number of nodes/links and network cost/complexity.
Communication Network Performance: 
Network Latency

Time to transfer n bytes from source to destination:
\[ \text{Time}(n)_{s-d} = \text{overhead} + \text{routing delay} + \text{channel occupancy} + \text{contention delay} \]

Unloaded Network Latency = routing delay + channel occupancy

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

\( b = \) channel bandwidth, \( \text{bytes/sec} \)
\( n = \) payload size
\( n_e = \) packet envelope: header, trailer.

Effective link bandwidth = \( bn / (n + n_e) \)

The term for unloaded network latency is refined next by examining the impact of flow control mechanism used in the network.

channel occupancy = transmission time

\( i.e. \) no contention delay \( t_c \)

\( S = \) Source
\( D = \) Destination

\(~ i.e. \) transmission time

\( \text{Added to payload} \)

Next
Flow Control Mechanisms:
Store&Forward (SF) Vs. Cut-Through (CT) Routing

Unloaded network latency for n byte packet:

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

\[ h = \text{distance in hops (number of links in route)} \]
\[ \Delta = \text{switch delay} \]

\[ b = \text{link bandwidth} \quad \text{n = size of message in bytes} \]

AKA Worm-Hole or pipelined routing

Usually Done by Data Link Layer

CMPE655 - Shaaban

#11 lec # 8  Spring 2014  4-22-2014
Store & Forward (SF) Vs. Cut-Through (CT) Routing Example

Example:
For a route with \( h = 3 \) hops or links, unloaded

Store & Forward (SF)

\[ T_{sf}(n, h) = h \left( \frac{n}{b} + \Delta \right) = 3 \left( \frac{n}{b} + \Delta \right) \]

Cut-Through (CT)

\[ T_{ct}(n, h) = \frac{n}{b} + h \Delta = \frac{n}{b} + 3 \Delta \]

Channel occupancy
Routing delay

b = link bandwidth
n = size of message in bytes
h = distance in hops
\( \Delta \) = switch delay

AKA Worm-Hole or pipelined routing

i.e No contention delay \( t_c \)
Communication Network Performance:

Refined Unloaded Network Latency Accounting For Flow Control
(i.e no contention, $T_c = 0$)

- For an unloaded network (no contention delay) the network latency to transfer an $n$ byte packet (including packet envelope) across the network:

  $$\text{Unloaded Network Latency} = \text{channel occupancy} + \text{routing delay}$$

- For store-and-forward (sf) routing:

  $$\text{Unloaded Network Latency} = T_{sf}(n, h) = h\left(\frac{n}{b} + \Delta\right)$$

- For cut-through (ct) routing:

  $$\text{Unloaded Network Latency} = T_{ct}(n, h) = \frac{n}{b} + h\Delta$$

  \[\begin{align*}
b &= \text{channel bandwidth} & n &= \text{bytes transmitted} \\
h &= \text{distance in hops} & \Delta &= \text{switch delay} \\
\end{align*}\]

  (number of links in route)

channel occupancy = transmission time
Reducing Unloaded Network Latency

(i.e. no contention, $T_c = 0$)

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

2. **Reduce number of links or 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.

3. **Increase link bandwidth $b$.**
4. **Reduce switch routing delay $\Delta$.**

*Unloaded implies no contention delay $t_c*
**Mapping of Task Communication Patterns to Topology**

**Example**

**Task Graph:**

- T1
- T2
- T3
- T4
- T5

**Parallel System Topology:**

*3D Binary Hypercube*

**Poor Mapping:**

- T1 runs on P0
- T2 runs on P5
- T3 runs on P6
- T4 runs on P7
- T5 runs on P0

- Communication from T1 to T2 requires 2 hops
  - Route: P0-P1-P5
- Communication from T1 to T3 requires 2 hops
  - Route: P0-P2-P6
- Communication from T1 to T4 requires 3 hops
  - Route: P0-P1-P3-P7
- Communication from T2, T3, T4 to T5
  - similar routes to above reversed (2-3 hops)

**Better Mapping:**

- T1 runs on P0
- T2 runs on P1
- T3 runs on P2
- T4 runs on P4
- T5 runs on P0

- Communication between any two communicating (dependant) tasks requires just 1 hop

---

From lecture 6

h = number of hops in route from source to destination
Available Effective Bandwidth

Factors affecting effective local link bandwidth available to a single node:

1. Accounting for Packet density: \(b \times \frac{n}{n + n_e}\)
2. Also Accounting for Routing delay: \(b \times \frac{n}{n + n_e + \Delta}\)
3. Contention: \(t_c\)
   - At endpoints.
   - Within the network.

Factors affecting throughput or Aggregate bandwidth:

1. Network bisection bandwidth:
   - Sum of bandwidth of smallest set of links when removed partition the network into two unconnected networks of equal size.

2. Total bandwidth of all the C channels: \(Cb\) bytes/sec, \(Cw\) bits per cycle or \(C\) phits per cycle.
   - Suppose \(N\) hosts each issue a message every \(M\) cycles with average routing distance \(h\) and average distribution: i.e. uniform distribution over all channels
      - Each message occupies \(h\) channels for \(\frac{\ell}{n/w}\) cycles
      - Total network load = \(Nh\ell / M\) phits per cycle.

   - Average Link utilization = Total network load / Total bandwidth
   - Average Link utilization: \(\rho = \frac{Nh\ell}{MC} < 1\)

Note: equation 10.6 page 762 in the textbook is incorrect
Network Saturation

Indications of Network Saturation

High queuing Delays

Link utilization =1

Potential or Offered Bandwidth

Delivered Bandwidth

Large Contention Delay $t_c$
Contention: Several packets trying to use the same link/node at same time.
  - May be caused by limited available buffering.
  - Possible resolutions/prevention:
    - Drop one or more packets (once contention occurs).
    - Increased buffer space.
    - Use an alternative route (requires an adaptive routing algorithm or a better static routing to distribute load more evenly).
    - Use a network with better bisection width (more routes).

To Prevent:

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

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.

Caused by communication load imbalance creating a high level of contention at these few nodes/links.
Deterministic Routing vs. Adaptive Routing

**Example: Routing in 2D Mesh**

1. **Deterministic (static) Dimension Order Routing in 2D mesh:** Each packet carries signed distance to travel in each dimension \([\Delta x, \Delta y]\). First move message along \(x\) then along \(y\).

2. **Adaptive (dynamic) Routing in 2D mesh:** Choose route along \(x, y\) dimensions according to link/node traffic to reduce node/link contention.
   - More complex to implement.

---

**Reducing node/link contention:**

- AKA Static
- AKA Dynamic

---

**Deterministic Dimension Routing along \(x\) then along \(y\) (node/link contention)**

**Adaptive (dynamic) Routing (reduced node/link contention)**
Sample Static Network Topologies

(Static or point-to-point)

- Linear
- Ring
- 2D Mesh
- Hypercube
- Binary Tree
- Fat Binary Tree
- Fully Connected

Higher link bandwidth
Closer to root
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

\[ \text{N(N-1)/2 Links} \quad \text{--> O(N^2) complexity} \]
\[ \text{Node Degree: N -1} \]
\[ \text{Diameter = N -1} \]
\[ \text{Average Distance = 2/3N} \]
\[ \text{Bisection Width = (N/2)^2} \]

**Linear Array:**

- AKA 1D Mesh

\[ \text{N-1 Links} \quad \text{--> O(N) complexity} \]
\[ \text{Node Degree: 1-2} \]
\[ \text{Diameter = N -1} \]
\[ \text{Average Distance = 2/3N} \]
\[ \text{Bisection Width = 1} \]

**Ring:**

- AKA 1D Torus
- Or Cube

\[ \text{N Links} \quad \text{--> O(N) complexity} \]
\[ \text{Node Degree: 2} \]
\[ \text{Diameter = N/2} \]
\[ \text{Average Distance = 1/3N} \]
\[ \text{Bisection Width = 2} \]

Examples: Token-Ring, FDDI, SCI (Dolphin interconnects SAN), FiberChannel Arbitrated Loop, KSR1

N = Number of nodes

Match network graph (topology) to task graph

Route A -> B given by relative address R = B - A

CMPE655 - Shaaban

#21 lec # 8  Spring 2014  4-22-2014
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 $0 \leq i_j \leq k_j - 1$ for $0 \leq j \leq d-1$

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

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

N = Total number of nodes
Properties of d-dimensional k-ary Meshes and Tori (k-ary d-cubes)

Routing:  
- Deterministic or static
  - Dimension-order routing (both).
    - 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.

Diameter:
- \( d(k-1) \) for mesh
- \( d \lfloor k/2 \rfloor \) for cube or torus

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

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

Bisection width:
- \( k^{d-1} \) links for mesh.
- \( 2k^{d-1} \) links for cube or torus.

For \( k = 2 \) Diameter = \( d \) (for both)

Number of Nodes:
- \( N = k^d \) for all

Number of Links:
- \( dN - dk \) for mesh
- \( dN = d \cdot k^d \) for cube or torus
  (More links due to wrap-around links)

\( N = \) Number of nodes
Static (point-to-point) Connection

Networks Examples:

2D Mesh

(2-dimensional $k$-ary mesh)

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

- Number of nodes $N = k^2$
- Node Degree: 2-4
- Network diameter: $2(k-1)$
- No of links: $2N - 2k$
- Bisection Width: $k$
- Where $k = \sqrt{N}$

How to transform 2D mesh into a 2D torus?

K = 4 nodes in each dimension

$k = 4$

$N = 16$

Diameter = $2(4-1) = 6$

Number of links = $32 - 8 = 24$

Bisection width = 4
Static Connection Networks Examples

Hypercubes

- Also called binary \( d \)-cubes (2-ary \( d \)-cube)
- Dimension = \( d = \log_2 N \)
- Number of nodes = \( N = 2^d \)
- Diameter: \( O(\log_2 N) \) hops = \( d \) = Dimension
- Good bisection width: \( N/2 \)
- Complexity:
  - Number of links: \( N(\log_2 N)/2 \)
  - Node degree is \( d = \log_2 N \)

Or: Binary \( d \)-cube
2-ary \( d \)-torus
Binary \( d \)-torus
Binary \( d \)-mesh
2-ary \( d \)-mesh?

A node is directly connected to \( d \) nodes with addresses that differ from its address in only one bit
Message Routing Functions Example
Dimension-order (E-Cube) Routing

Network Topology:
3-dimensional static-link hypercube
Nodes denoted by $C_2C_1C_0$

Routing by least significant bit $C_0$

Routing by middle bit $C_1$

Routing by most significant bit $C_2$

For Hypercubes: Diameter = max hops = $d$ here $d = 3$
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 \(k\). (Not for leaves, for leaves degree = 1)
- Route up to common ancestor and down:
  - \(R = B \oplus A\)
  - 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 Width = 1

Good? Or Bad?
Static Connection Networks Examples: Fat-Trees

- “Fatter” higher bandwidth links (more connections in reality) as you go up, so bisection bandwidth scales with number of nodes N.

- Example: Network topology used in Thinking Machine CM-5
Embedding A Binary Tree Onto A 2D Mesh

Embedding:
In static networks refers to mapping nodes of one network (or task graph?) onto another network while attempting to minimize extra hops.

H-Tree Configuration to embed binary tree onto a 2D mesh

(A) = Additional nodes added to form the tree

(i.e. Extra hops)

(PP, Chapter 1.3.2)
Embedding A Ring Onto A 2D Torus

The 2D Torus has a richer topology/connectivity than a ring, thus it can embed it easily without any extra hops needed.

Ring:
Node Degree = 2
Diameter = \(\lceil N/2 \rceil\)
Links = N
Bisection = 2

Here
N = 16
Diameter = 8
Links = 16

2D Torus:
Node Degree = 4
Diameter = 2 \(\lceil k/2 \rceil\)
Links = 2N = 2 \(k^2\)
Bisection = 2k

Here \(k = 4\)
Diameter = 4
Links = 32
Bisection = 8

Also: Embedding a binary tree onto a Hypercube is done without any extra hops

Extra Hops Needed?
Dynamic Connection Networks

- **Switches** are usually used to **dynamically implement connection paths** or virtual circuits between nodes instead of fixed point-to-point connections.
- Dynamic connections are established by configuring switches based on communication demands.
- Such networks include:
  1. **Bus systems.**
  2. **Multi-stage Interconnection Networks (MINs):**
     - Omega Network.
     - Baseline Network.
     - Butterfly Network, etc.
  3. **Single-stage Crossbar switch networks.**
     (one N x N large switch)
     
     - **O(N^2) Complexity?**
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 NxN Switches

- **Input Buffer**
- **Switch Fabric**
  - Complexity $O(N^2)$
  - Cross-bar
- **Output Buffer**

- Or implement in stages then complexity $O(N \log N)$

- **Control**
  - Routing, Scheduling

$\Delta = \text{Total Switch Routing Delay}$

Implemented using one large $N \times N$ switch or by using multiple stages of smaller switches
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.
  - May support quality of service constraints/priority routing.
## Switch Size And Legitimate States

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

Switch Size = All Legitimate States (includes broadcasts)

Example: Four states for 2x2 switch

- 2 permutation connections
- 2 broadcast connections

For n x n switch: Complexity = O(n^2)  n= number of input or outputs
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 (one to one) mapping:

\[ a \rightarrow b, \quad b \rightarrow c, \quad c \rightarrow a, \quad d \rightarrow e, \quad e \rightarrow d \]

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 one position left.
- The perfect shuffle and its inverse for 8 objects are shown here:

Perfect Shuffle (circular shift left one position)

Inverse Perfect Shuffle: rotate binary address one position right

e.g. For N = 8
Generalized Structure of Multistage Interconnection Networks (MINS)

Fig 2.23 page 91

Kai Hwang ref.

See handout
In the Omega network, perfect shuffle is used as an inter-stage connection (ISC) 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 in one pass, so it is possible to have permutations that cannot be provided in one pass (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 in one pass.
- 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 (MINS) Example: The Omega Network $\Omega$

ISC patterns used define MIN topology/connectivity
Here, ISC used for Omega network is perfect shuffle

$N =$ size of network

2x2 switches used $\log_2 N$ stages
Multi-Stage Networks: The Omega Network

ISC = Perfect Shuffle
a = b = 2 (i.e. 2x2 switches used)

Node Degree = 1 bi-directional link or 2 uni-directional links
Diameter = \( \log_2 N \) (i.e. number of stages)
Bisection width = \( N/2 \)

\( N/2 \) switches per stage, \( \log_2 N \) stages, thus:
Complexity = \( O(N \log_2 N) \)

Fig 2.24 page 92

Kai Hwang ref.

See handout (for figure)
MINs Example: Baseline Network

Fig 2.25  page 93

Kai Hwang ref.

See handout
MINs Example: Butterfly Network

- Complexity: \(N/2 \times \log_2 N\) (\(\#\) of switches in each stage \(\times\) \(\#\) of stages)
- 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 width = \(N/2\)
- Diameter \(\log_2 N\) = Number of stages

\(N = \text{Number of nodes}\)
Relationship Between Butterfly Network & Hypercubes

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

**O(log₂ N)** Stage N-node MIN using 2x2 switches:

- **Max distance:** \( \log₂ N \) (good latency scaling)
- **Number of switches:** \( \frac{1}{2} N \log N \) (good complexity scaling)
- overhead = \( o = 1 \) us, BW = 64 MB/s, \( \Delta = 200 \) ns per hop
- Using pipelined or cut-through routing:
  - \( T_{64}(128) = 1.0 \) us + \( 2.0 \) us + 6 hops * 0.2 us/hop = 4.2 us
  - \( T_{1024}(128) = 1.0 \) us + \( 2.0 \) us + 10 hops * 0.2 us/hop = 5.0 us
- **Store and Forward**
  - \( T_{64}^{sf}(128) = 1.0 \) us + 6 hops * (2.0 + 0.2) us/hop = 14.2 us
  - \( T_{1024}^{sf}(128) = 1.0 \) us + 10 hops * (2.0 + 0.2) us/hop = 23 us

Only 20% increase in latency for 16x network size increase

~ 60% increase in latency for 16x network size increase

Latency when sending \( n = 128 \) bytes for \( N = 64 \) and \( N = 1024 \) nodes
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

Both networks used in T3D and T3E are: Point-to-point (static) using the 3D Torus topology

- **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 Network Examples

\[ \tau = \frac{1}{f} \quad \text{W or Phit} \quad \Delta \]

<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>