Chapter 2

Interconnection Networks and Hybrid Topologies

This chapter provides background knowledge about interconnection networks (INs) and hybrid topologies. Various properties associated with interconnection networks and metrics used to gauge the performance of various topologies are explained.

The major constituents of a multiprocessor system are multiple CPUs connected through an interconnection network and the software that enables the CPUs to work together. Communication in message passing systems is performed by send and receive commands, while in shared memory multiprocessor systems, communication is performed by writing to and reading from the global memory. In both cases, the interconnection network plays a vital role in determining the communication speed.

2.1 Criteria Used for Classification of INs

Several surveys related to interconnection networks have been reported in literature [164][197][106][206]. The needs of the communication industry particularly in the field of telephone switching, influenced the early work on interconnection net-
works. As the need for more computing power increased, applications for interconnection networks within computing machines began to become evident. Amongst the first of these were matrix manipulation, sorting etc., but as interest in parallel processing increased, numerous interconnection networks were proposed both for shared memory systems and for distributed memory systems [196]. With advances in fast packet switching, interest in interconnection networks which were originally proposed for parallel processing are now being considered for use in fast packet switch based systems. According to Bhuyan [42] multiprocessor interconnection networks (INs) can be classified based on the following criteria:

- Mode of operation (synchronous versus asynchronous)
- Control strategy (centralized versus decentralized)
- Switching techniques (circuit versus packet)
- Topology (static versus dynamic)

El-Rewini and Abd-El-Barr [101] and Bhuyan et al. [43] have also used the same criteria.

### 2.1.1 Mode of Operation

Based on the mode of operation INs are classified as synchronous or asynchronous. In synchronous mode of operation, a single global clock is used by all components within the system while the asynchronous mode of operation does not require a global clock; instead handshaking signals are used in order to coordinate system operation.

### 2.1.2 Control Strategy

Based on control strategy INs are classified as centralized or decentralized. In centralized control systems, a single central control unit is used to coordinate and control the operation of the components within the system. In decentralized
2.2. Topology Based Classification for INs

control, the control function is distributed among different components within the system.

2.1.3 Switching Techniques

This approach classifies the interconnection networks as packet switched or circuit switched. In the circuit switching mechanism, a complete path has to be established prior to the start of communication between a source and a destination while in a packet switching mechanism, communication between a source and destination takes place via messages that are divided into smaller entities, called packets.

2.1.4 Topology

An interconnection network topology is a mapping function from the set of processors and memories onto the same set of processors and memories.

In general, interconnection networks can be classified as static or dynamic. In static networks, direct fixed links are established among nodes to form a fixed network, while in dynamic networks, connections are established as needed. Switching elements are used to establish connections between inputs and outputs. Depending on the switch settings, different interconnections can be established. Nearly all multiprocessor systems can be distinguished by their interconnection network topology.

2.2 Topology Based Classification for INs

On the basis of topology, interconnection networks (INs) can be classified as static or dynamic. In static INs, fixed links are created at the time of fabrication while dynamic INs establish connections on the fly based upon connection requirements. In dynamic networks, switches are used to establish connections between inputs and outputs. By manipulating the switch settings, different interconnections can
2.3 Dynamic Interconnection Networks

Dynamic interconnection networks can be either be bus based or switch based. The following subsections discuss them:

2.3.1 Bus Based Dynamic Interconnection Networks

Single Bus Systems

A single bus is considered the simplest way to connect multiprocessor systems. Figure 2.2 shows an illustration of a single bus system.
In its general form, such a system consists of \( N \) processors, each having its own cache, connected by a shared bus.

The use of local caches reduces the processor-memory traffic. All processors communicate with a single shared memory.

The typical size of such a system varies between 2 and 50 processors.

System size is limited by the bandwidth of the bus and the fact that only one processor can access the bus, and in turn only one memory access can take place at any given time.

**Multiple Bus Systems**

The use of multiple buses to connect multiple processors is a natural extension to the single shared bus system.

- A multiple bus multiprocessor system uses several parallel buses to interconnect multiple processors and multiple memory modules.

- A number of connection schemes are possible in this case, namely multiple bus with full bus memory connection (MBFBMC), multiple bus with single bus memory connection (MBSBMC), multiple bus with partial bus memory connection (MBPBMC), and multiple bus with class-based memory connec-
2.3. Dynamic Interconnection Networks

The multiple bus with full busmemory connection has all memory modules connected to all buses. The multiple bus with single busmemory connection has each memory module connected to a specific bus. The multiple bus with partial busmemory connection has each memory module connected to a subset of buses. The multiple bus with class-based memory connection has memory modules grouped into classes whereby each class is connected to a specific subset of buses.

In general, multiple bus multiprocessor organization offers a number of desirable features such as high reliability and ease of incremental growth.

On the other hand, when the number of buses is less than the number of memory modules (or the number of processors), bus contention is expected to increase.

2.3.2 Switched Based Interconnection Networks

In this type of network, connections among processors and memory modules are made using simple switches. Three basic interconnection topologies exist: crossbar, single-stage, and multistage.

Crossbar

• Unlike the single bus which can provide only a single connection, a crossbar can provide simultaneous connections among all its inputs and outputs.

• The crossbar contains a switching element (SE) at the intersection of any two lines extended horizontally or vertically inside the switch.

• Figure 2.5 shows an $8 \times 8$ crossbar. A SE (also called a cross-point) is provided at each of the 64 intersection points.
Simultaneous connections between $P_i$ and $M_{8-i+1}$ for $1 \leq i \leq 8$ are made.

The two possible settings of an SE in the crossbar (straight and diagonal) are also shown in the figure.

As can be seen from the figure, the number of SEs (switching points) required is 64 and the message delay to traverse from the input to the output is constant, regardless of which input/output are communicating.

In general for an $N \times N$ crossbar, the network complexity, measured in terms of the number of switching points, is $O(N^2)$ while the time complexity, measured in terms of the input to output delay, is $O(1)$.

Crossbar is a nonblocking network that allows a multiple input/output connection pattern (permutation) to be achieved simultaneously.

For a large multiprocessor system the complexity of the crossbar can become...
2.3. Dynamic Interconnection Networks

Figure 2.4: (i) MBPBMC (ii) MBCBMC

Figure 2.5: (i) MBFBMC (ii) MBSBMC
a dominant financial factor.

**Single Stage Networks**

In this case, a single stage of switching elements (SEs) exists between the inputs and the outputs of the network. The simplest switching element that can be used is the $2 \times 2$ switching element (SE). Figure 2.6 shows the four possible settings that a SE can assume. These settings are called straight, exchange, upper-broadcast, and lower-broadcast [204].

![Figure 2.6: Different settings of the $2 \times 2$ SE](image)

- In the straight setting, the upper input is transferred to the upper output and the lower input is transferred to the lower output.
- In the exchange setting, the upper input is transferred to the lower output and the lower input is transferred to the upper output.
- In the upper-broadcast setting, the upper input is broadcast to both the upper and the lower outputs.
- In the lower-broadcast setting, the lower input is broadcast to both the upper and the lower outputs.

Following are some examples of Single Stage Networks:

**Shuffle Exchange Network**

- To establish communication between a given input (source) to a given output (destination), data has to be circulated a number of times around the network.
2.3. Dynamic Interconnection Networks

- A well-known connection pattern for interconnecting the inputs and the outputs of a single-stage network is the ShuffleExchange.

- Two operations are used. These can be defined using an m bit-wise address pattern of the inputs \(p_{m-1}p_{m-2}p_{m-3}...p_1p_0\) as follows:

\[
S(p_{m-1}p_{m-2}p_{m-3}...p_1p_0) = p_{m-2}p_{m-3}p_1p_0p_{m-1}
\]

\[
E(p_{m-1}p_{m-2}p_{m-3}...p_1p_0) = p_{m-1}p_{m-2}p_{m-3}...p_1\overline{p}_0
\]

- With shuffle \((S)\) and exchange \((E)\) operations, data is circulated from input to output until it reaches its destination.

- To complete the network every output is buffered and fed back to its corresponding input. Packets of data therefore circulate through the structure until they exit at the desired output [68].

- For example, if the number of inputs (processors), in a single-stage IN is \(N\) and the number of outputs (memories), is \(N\), the number of SEs in a stage is \(N/2\).

- The maximum length of a path from an input to an output in the network, measured by the number of SEs along the path, is \(\log_2 N\).

- Stone [205] introduced the perfect shuffle as a pattern of interconnection links of some interest. The goal was to solve a number of classes of computational problem via a tightly coupled parallel processor.

The Cube Network

- The interconnection pattern used in the cube network is defined as follows:

\[
C_i(p_{m-1}p_{m-2}p_{m-3}...p_{i+1}p_{i}p_{i-1}p_0) = p_{m-1}p_{m-2}p_{m-3}...p_{i+1}\overline{p}_ip_{i-1}...p_1p_0
\]

- Considering a 3-bit address \((N = 8)\), then \(C_2(6) = 2\), \(C_1(7) = 5\) and \(C_0(4) = 5\). Figure 2.7 shows the cube interconnection patterns for a network with \(N = 8\).
The network is called the cube network due to the fact that it resembles the interconnection among the corners of an n-dimensional cube \((n = \log_2 N)\).

**The Butterfly Network**

- The interconnection pattern of Butterfly Network uses the butterfly function, which is defined as:

\[
B(p_{m-1}p_{m-2}p_{m-3} \ldots p_1p_0) = p_0p_{m-2}p_{m-3} \ldots p_1p_{m-1}
\]

- For a 3-bit address \((N = 8)\), the following is the butterfly mapping:

\[
\begin{align*}
B(000) & = 000 \\
B(001) & = 100 \\
B(010) & = 010 \\
B(011) & = 110 \\
B(100) & = 001
\end{align*}
\]
2.3. Dynamic Interconnection Networks

\[
\begin{align*}
B(101) &= 101 \\
B(110) &= 011 \\
B(111) &= 111
\end{align*}
\]

Multistage Networks

- If multiple copies of the single stage shuffle exchange are cascaded, a multi-stage interconnection network (MIN) results and is called a multi-stage shuffle exchange. Data is no longer required to circulate through the network but passes through the structure from input to output.

- The most undesirable single bus limitation that MINs are set to improve is the availability of only one single path between the processors and the memory modules. Thus, MINs provide a number of simultaneous paths between the processors and the memory modules.

- As shown in Figure 2.8, a general MIN consists of a number of stages each consisting of a set of \(2 \times 2\) switching elements. Stages are connected to each other using Inter-stage Connection (ISC) Pattern.

Figure 2.8: Multistage interconnection network
2.3. Dynamic Interconnection Networks

Figure 2.9: An example of $8 \times 8$ ShuffleExchange network (SEN).
• Figure 2.9 shows an example of an $8 \times 8$ MIN that uses the $2 \times 2$ SEs described before. This network is known as the ShuffleExchange network (SEN).

• There exist $\log_2 N$ stages in a $N \times N$ MIN.

• Other examples of multi-stage interconnection networks include:

\textbf{The Banyan Network}

![Banyan Network Diagram]

Figure 2.10: An example of 8 x 8 Banyan Network.

• Figure 2.10 shows an example of an 8 x 8 Banyan network.

• The design principle of banyan network is that: if the number of inputs (processors) is $N$ and the number of outputs (memory), is also $N$, the number of MIN stages is $\log_2 N$ and the number of SEs per stage is $N/2$. Hence, the network complexity, measured in terms of the total number of SEs is $O(N \log_2 N)$. 

The Omega Network

Figure 2.11: An example of $8 \times 8$ Omega Network.

- Figure 2.11 shows an example of an $8 \times 8$ Omega network.
- A size $N$ omega network consists of $\log_2 N$ single-stage shuffle exchange networks \[143\].
- Each stage consists of a column of $N/2$, two-input switching elements whose input is a shuffle connection.

There is a long list of multistage interconnection networks reported in literature. Some better known examples include The Delta Network \[173\], The Clos \[78\], The Benes \[38\] and The Batcher Sorting Network \[36\].
2.3.3 Blocking and Non-Blocking Networks

- Multi-stage interconnection networks may be further classified according to the blocking characteristics they present which is reflected in the throughput they offer to traffic with a random distribution of packet destinations.

- Blocking networks possess the property that in the presence of a currently established interconnection between a pair of input/output, the arrival of a request for a new interconnection between two arbitrary unused input and output may or may not be possible. Examples of blocking networks include Omega, Banyan, ShuffleExchange, and Baseline.

- Nonblocking networks are characterized by the property that in the presence of a currently established connection between any pair of input/output, it will always be possible to establish a connection between any arbitrary unused pair of input/output.

- Re-arrangeable networks are characterized by the property that it is always possible to rearrange already established connections in order to make allowance for other connections to be established simultaneously. The Benes is a well-known example of a re-arrangeable network.

2.4 Static Interconnection Networks

There are two types of static interconnection networks:

- Completely Connected Networks (CCNs)

- Limited Connection Networks (LCNs)

Before proceeding further with discussing static interconnection networks, it is necessary to introduce some important topological definitions which play a major role in grading of these networks.
2.4. Static Interconnection Networks

**Degree** ($d$)

The Degree ($d$) of a node in a network is defined as the number of channels incident on the node. The number of channels into the node is termed as the in-degree, $d_{in}$. The number of channels out of a node is termed as the out-degree, $d_{out}$. The total degree, $d$, is the sum, $d = d_{in} + d_{out}$. For an undirected network, the number of edges incident on a node is called the degree of that node. The node degree reflects the number of I/O ports required per node, and thus the cost of the network. Therefore, the node degree should be kept constant and as small as possible in order to reduce cost. A constant node degree is very much desired to achieve modularity in building blocks for scalable systems.

**Diameter** ($D$)

The Diameter ($D$) of a network having $N$ nodes is defined as the longest path, $l$, of the shortest paths between any two nodes $D = \max \left( \min_{i,j} (\text{length}(l)) \right)$. In this equation, $l_{ij}$ is the length of the path between nodes $i$ and $j$ and $\text{length}(l)$ is a procedure that returns the length of the path, $l$. In simple words, diameter of a network is the maximum shortest path between any two nodes. The path length is measured by the number of links traversed. The network diameter indicates the maximum number of distinct hops between any two nodes, thus provides a figure of communication merit for the network [129]. It should be obvious that the diameter should be as small as possible from a communication point of view.

**Bisection Width** ($\omega$)

Bisection Width ($b$) of a network is the minimum number of edges that must be removed in order to divide the network into two halves. In case of a communication network each edge corresponds to a channel with $w$ bit wires. Then the wire bisection width is $B = bw$. The parameter $B$ reflects the wiring density of a network. When $B$ is fixed, the channel width (in bits) $w = B/b$. Thus the bisection width is a good indicator of the maximum communication bandwidth along the
bisection of a network.

**Symmetry**

A network is said to be symmetric if it is isomorphic to itself with any node labeled as the origin; that is, the network looks the same from any node. Rings and Tori networks are symmetric while linear arrays and mesh networks are not.

**Network Throughput**

Network throughput is an indicative measure of the message carrying capacity of a network. It is defined as the total number of messages the network can transfer per unit time. To estimate the throughput, the capacity of the network and the messages number of actually carried by the network are calculated. Practically the throughput is only a fraction of its capacity.

**Latency**

Latency is the delay in transferring the message between two nodes.

**Data Routing Functions**

These are the functions which when executed, establish the path between the source and the destination. In dynamic interconnection networks, there can be various interconnection patterns that can be generated from a single network. This is done by executing various data routing functions. Thus data routing operations are used for routing the data between various processors.

**Dimensionality of Interconnection Network**

Dimensionality indicates the arrangement of nodes or processing elements in an interconnection network. In a single dimensional or linear network, nodes are
connected in a linear fashion; in a two dimensional network the processing elements (PEs) are arranged in a grid while in a cube network they are arranged in a three dimensional network.

**Hardware Cost**

It refers to the cost involved in the implementation of an interconnection network. It includes the cost of switches, arbiter unit, connectors, arbitration unit, and interface logic.

### 2.4.1 Completely Connected Networks

- In a completely connected network (CCN), each node is connected to all other nodes in the network.
- Completely connected networks guarantee fast delivery of messages from any source node to any destination node (only one link has to be traversed).
- Since every node is connected to every other node in the network, routing of messages between nodes becomes a straightforward task.
2.5 Limited Connection Networks

- Completely connected networks are, however, expensive in terms of the number of links needed for their construction. This disadvantage becomes more and more apparent for higher values of \( N \).

- The number of links in a completely connected network is given by \( N(N - 1)/2 \), that is, \( O(N^2) \).

- The delay complexity of CCNs, measured in terms of the number of links traversed as messages are routed from any source to any destination is constant, that is, \( O(1) \).

- An example having \( N = 6 \) nodes is shown in Figure 2.12. A total of 15 links are required in order to satisfy the complete interconnectivity of the network.

2.5 Limited Connection Networks

Limited connection networks (LCNs) do not provide a direct link from every node to every other node in the network. Instead, communications between some nodes have to be routed through other nodes in the network. The length of the path between nodes, measured in terms of the number of links that have to be traversed, is expected to be longer compared to the case of CCNs. Two other conditions seem to have been imposed due to limited interconnectivity in LCNs. These are: the need for a pattern of interconnection among nodes and the need for a mechanism for routing messages around the network until they reach their destinations. A number of regular interconnection patterns have evolved over the years for LCNs. These patterns include:

Linear Array

- This is a one dimensional network in which \( N \) nodes are connected by \( N - 1 \) links in a line.
2.5. Limited Connection Networks

- Internal nodes have degree 2 and the terminal nodes have degree 1. The diameter is $N - 1$, which is rather long for large $N$. The Bisection Width is $b = 1$.

- Linear arrays are the simplest connection topology. However, the structure is asymmetric and communication is quite inefficient when $N$ becomes very large.

- As the diameter increases linearly with respect to $N$, linear arrays are not suited for large $N$. For very small $N$, it is rather economical to implement a linear array.

**Ring and Chordal Ring**

- A ring is obtained by connecting the two terminal nodes of a linear array with one extra link.

- A ring can be unidirectional or bidirectional. It is symmetric with a constant node degree of 2.

- The diameter is $\lfloor N/2 \rfloor$ for a bidirectional ring and $N$ for unidirectional ring.

- By increasing the node degree from 2 to 4 we obtain a chordal ring.

- Two or more links may be added to produce other chordal rings. In general, the more links added, the higher the node degree and the shorter the network diameter.

**Tree and Star**

- A binary tree network consists of nodes connected in a binary tree arrangement.

- In general a k-level, completely balanced binary tree should have $N = 2k - 1$ nodes. The maximum mode degree is 3 and the diameter is $2(k - 1)$. 
2.5. Limited Connection Networks

- With a constant node degree the binary tree is a scalable architecture. However, the diameter is rather long.

- The Star is a two level tree with a high node degree of \(d = N - 1\) and a small constant diameter of 2.

**Fat Trees**

- The channel width of a fat tree increases as we ascend from levels to root [148].

- A fat tree is more like a real tree in that branches get thicker toward the root.

- One of the major problems in using the conventional binary tree is the bottleneck problem toward the root, since traffic toward the root becomes heavier. The fat tree was proposed to alleviate the problem.

- The idea of fat tree has been applied in the Connection Machine CM-5. The idea of binary trees can also be extended to multiway fat trees.

**Mesh and Torus**

- A \(3 \times 3\) mesh network consists of 9 nodes.

- In general, a \(k\)-dimensional mesh with \(N = nk\) nodes has an interior node degree of \(2k\) and the network diameter is \(k(n - 1)\). The node degrees at the boundary and corner nodes are 3 and 2 respectively.

- Pure mesh is not symmetric. However, some of its variants may be symmetric.

- The torus can be viewed as another variant of the mesh with an even shorter diameter.

- This topology combines the ring and mesh and extends to higher dimensions. The torus has ring connections across each row and along each column of the array.
In general an \( n \times n \) torus has a node degree of 4 and a diameter of \( \lfloor n \rfloor / 2 \).

The torus is a symmetric topology. All added wraparound connections help reduce the diameter by one-half from that of mesh.

**Systolic Arrays**

- This is a class of multidimensional pipelined array architectures designed for implementing fixed algorithms.

- A systolic array has been specially designed for performing matrix-matrix multiplication.

- The systolic array has become a popular research area ever since its introduction. With fixed interconnection and synchronous operation, a systolic array matches the communication structure of the algorithm.

- For special applications like image/ signal processing, systolic arrays may offer a better performance/ cost ratio.

**Hypercube**

- This is a binary \( n \)-cube architecture which has been implemented in the iPSC, nCUBE and CM-2 systems.

- In general, an \( n \)-cube consists of \( N = 2^n \) nodes spanning along \( n \) dimensions, with two nodes per dimension.

- The node degree of an \( n \)-cube equals \( n \) and so does the network diameter.

- In fact, the node degree increases linearly with respect to the dimension, making it difficult to consider the hypercube a scalable architecture.

- Binary hypercube has been a very popular architecture for research and development. Both Intel iPSC/1, iPSC/2 and nCUBE machines were built based upon the hypercube architecture. The architecture has a dense connection.
2.6 Hybrid Interconnection Networks

- With poor scalability and difficulty in packaging higher-dimensional hypercubes, the architecture has been replaced by other architecture. For example the CM-5 chooses the fat tree over the hypercube implemented in CM-2. The Intel Paragon chooses a two dimensional mesh over its hypercube predecessors.

Besides above popular networks, numerous limited connection networks like Cube Connected Cycles (CCC) [176], Barrel Shifter etc. have also been reported in literature. Table 2.5 lists topological properties of some important limited connection networks for network size of \( N \):

<table>
<thead>
<tr>
<th>Network</th>
<th>Degree</th>
<th>Diameter</th>
<th>Bisection Width</th>
<th>Symmetry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Array</td>
<td>2</td>
<td>( N - 1 )</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>Ring</td>
<td>2</td>
<td>( \lceil N \rceil /2 )</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>Completely</td>
<td>2</td>
<td>1</td>
<td>((N/2)^2)</td>
<td>Yes</td>
</tr>
<tr>
<td>Connected Binary Tree</td>
<td>3</td>
<td>( 2(\log_2 N - 1) )</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>Star</td>
<td>( N - 1 )</td>
<td>2</td>
<td>( \lceil N \rceil /2 )</td>
<td>No</td>
</tr>
<tr>
<td>2D-Mesh</td>
<td>4</td>
<td>( 2(\sqrt{N} - 1) )</td>
<td>( \sqrt{N} )</td>
<td>No</td>
</tr>
<tr>
<td>2D-Torus</td>
<td>4</td>
<td>( 2\lceil \sqrt{N}/2 \rceil )</td>
<td>( 2\sqrt{N} )</td>
<td>Yes</td>
</tr>
<tr>
<td>k-Hypercube</td>
<td>( \log_k N )</td>
<td>( \log_k N )</td>
<td>( N/2 )</td>
<td>Yes</td>
</tr>
<tr>
<td>( \text{CCC}(k), \ k = 3 )</td>
<td>( (2k - 1 + \lceil k/2 \rceil) )</td>
<td>( N/(2k) )</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

---

Table 2.1: Topological properties of limited connection networks

2.6 Hybrid Interconnection Networks

Hybrid topology, as the name suggests, is a combination of two or more topologies. Two types of hybrid interconnection networks have been proposed in literature:
1. Hierarchical Interconnection Networks: Hierarchical interconnection networks (HINs) provide a framework for designing networks with reduced link cost by taking advantage of the locality of communication that exists in parallel applications [2]. Some well known networks in this category are Hierarchical $n$-Hypercube [161], Extended Hypercube [142], dBCube [67], HFCube [195] etc. Abd-El-Barr and Al-Somani [2] present a good review of various hierarchical networks and their topological properties.

2. Product Networks: There are several graph-theoretic operators which enable us to derive large networks from small networks. Cross product or Cartesian product is one of them.

Cross product of undirected graphs as a means of combining two topologies with known properties has been extensively exploited in literature. The resultant graph retains the properties of both the graphs. Example includes multidimensional meshes, tori, $\omega$-graphs [190], Banyan-Hypercubes [235], mesh connected trees [98] etc.

The next chapter presents a highly scalable and economical topology for parallel and distributed systems. The proposed topology is based on the cross product of two topology graphs. Hence, to facilitate ease of understanding of the next chapter, the next section is devoted to Product Networks.

### 2.7 Cross Product as Net. Synthesizing Operator

Let $X_1 = (V_1, E_1)$ and $X_2 = (V_2, E_2)$ be two undirected graphs representing two interconnection topologies, where $V_1$ and $V_2$ are sets of vertices (processors) and $E_1$ and $E_2$ are sets of edges (interconnections between processors). Then the cross product of the two topologies is defined as below:

**Definition 2.7.1** Cross Product: The cross product $X = X_1 \otimes X_2$ of two undirected connected graphs $X_1 = (V_1, E_1)$ and $X_2 = (V_2, E_2)$ is the undirected graph $X = (V, E)$, where $V$ and $E$ are given by:
1. \( V = \langle x_1, x_2 \rangle \mid x_1 \in V_1 \quad \text{and} \quad x_2 \in V_2, \) and

2. for any \( u = \langle x_1, x_2 \rangle \) and \( v = \langle y_1, y_2 \rangle \) in \( V, (u, v) \) is an edge in \( E, \) iff either 
\((x_1, y_1)\) is an edge in \( E_1 \) and \( x_2 = y_2, \) or \((x_2, y_2)\) is an edge in \( E_2 \) and \( x_1 = y_1. \)

An Example

![Diagram of Cross Product]

Figure 2.13: An example of Cross Product

Consider \( X_1 = (V_1, E_1), \) where, \( V_1 = \{1, 2\} \) and \( E_1 = \{(1, 2)\} \)

\( X_2 = (V_2, E_2), \) where, \( V_2 = \{1, 2, 3\} \) and \( E_2 = \{(1, 2), (2, 3), (3, 1)\} \)

The product graph \( X = X_1 \otimes X_2 \) is shown in Figure 2.13.

As can be seen from the above example \( X = X_1 \otimes X_2 \) can be constructed by replacing each vertices of \( X_1 \) by \( X_2 \) (or by repeating \( X_2, |V_1| \) times) and setting up the appropriate links as per Definition 2.7.1. Here \( |V_1| \) means the cardinality of set \( V_1. \) Clearly, the Cartesian Product Operator is a network synthesizer. It takes two networks \( X_1 \) and \( X_2 \) and synthesizes a third network \( X = X_1 \otimes X_2. \)

2.7.1 Topological Properties of Product Networks

Topological properties of product networks have been described in [234]. From the point of view of this thesis the following properties are important, hence are being briefly discussed below. Throughout the discussion the following notion have been adopted:

- \( \text{deg}_X(x) = \) the degree of node \( x \) in graph \( X \)
\( d_X(x, y) \) = the distance from \( x \) to \( y \) in \( X \)

- \( D_X \) = the diameter of graph \( X \)
- \( \overline{d}_X \) the average distance of \( X \)

**Definition 2.7.2** The Degree Formula: If \( X_1 \) and \( X_2 \) are two graphs of degrees \( \text{deg}_{X_1} \) and \( \text{deg}_{X_2} \) respectively then the degree of \( X_1 \otimes X_2 \) i.e. \( \text{deg}_{(X_1 \otimes X_2)} \) is given by:

\[
\text{deg}_{(X_1 \otimes X_2)} = \text{deg}_{X_1} + \text{deg}_{X_2}
\]

**Definition 2.7.3** Size and Diameter Formula: if \( X_1 \) and \( X_2 \) are two undirected connected graphs with diameters \( D_{X_1} \) and \( D_{X_2} \) and of sizes \( N_1 \) and \( N_2 \) respectively then:

- \( X_1 \otimes X_2 \) is connected
- the size \( N \) and the diameter \( D_{(X_1 \otimes X_2)} \) of \( X_1 \otimes X_2 \) are given by \( N = N_1 \times N_2 \) and \( D_{(X_1 \otimes X_2)} = D_{X_1} + D_{X_2} \)

**Corollary 2.7.4** It is obvious that \( D_X(n) = nD_X \)

**Definition 2.7.5** Shortest Path Formula: If \( x, x_1, x_2, \ldots, x_l, y \) is a shortest path from \( x \) to \( y \) in \( X_1 \) and if \( x', x_1', x_2', \ldots, x_l', y' \) a shortest path from \( x \) to \( y \) in \( X_2 \), then \( xx', x_1x', x_2x', \ldots, x_lx', yy' \) is a shortest path from \( xx' \) to \( yy' \) in \( X_1 \otimes X_2 \).

**Definition 2.7.6** Average Distance Formula: if \( X_1 \) and \( X_2 \) are two undirected connected graphs then

\[
\overline{d}_{(X_1 \otimes X_2)} = \overline{d}_{X_1} + \overline{d}_{X_2}
\]

**Corollary 2.7.7** It is obvious that \( \overline{d}_X(n) = n\overline{d}_X \)

### 2.7.2 Significance of Cross Product

The following example illustrates the significance of Product Networks.
Example

Let $SC(n, h)$ be the cross product of the $n$-Star Graph $S(n)$ and the $h$-Hypercube $C(h)$. $SC(n, h)$ offers a higher flexibility in choosing the network size and adjusting the degree and diameter parameters. Table 2.2 on page 50 illustrates this feature by showing the size, degree and diameter of $S(n)$, $C(h)$ and $SC(n, h)$. From this table, it can be observed that in all the cases considered, $SC(n, h)$ has a better fit to the desired size than $S(n)$ and a lower degree and diameter than $C(h)$.

2.7.3 Routing on Product Networks:

A distributed routing algorithm for a network $X = (V, E)$ is a function $R$ from $V \times V$ to $V$ that associates for each pair of nodes (current, destination) a node next, where current is the node at which a given message is currently stored, destination is the destination node for that message, and next is the next node to be visited by the message in its way to the destination node. Let $R_1$ and $R_2$ be two distributed routing algorithms for $X_1$ and $X_2$, respectively then, a distributed routing algorithm $R$ for $X_1 \otimes X_2$ is given by:

$$R[(cur_1, cur_2), (dest_1, dest_2)] = \begin{cases} 
[R_1(cur_1, dest_1), cur_2] & \text{if } cur_1 \neq cur_2 \\
[cur_1, R_2(cur_2, dest_2)] & \text{if } cur_1 = cur_2
\end{cases}$$

The above routing rules route a message along $X_1$-edges until the $X_2$-component of the current node is equated to the $X_1$-component of the destination node. Once that is achieved, the routing continues along $X_2$ edges. The relative order of these two routing stages may be reversed or interleaved.

2.7.4 Broadcasting on Product Networks

Many parallel and distributed applications require an efficient broadcast algorithm. Given a broadcast algorithm for each $X_1$ and $X_2$, a broadcast algorithm for $X_1 \otimes X_2$, can be obtained as follows. Assume that a message $M$, originally at
### Table 2.2: Significance of product networks

<table>
<thead>
<tr>
<th>Desired Size</th>
<th>Star Graph, $S(n)$</th>
<th>Hypercube, $C(h)$</th>
<th>Star Cube, $SC(n,h)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n$</td>
<td>Size</td>
<td>Degree</td>
</tr>
<tr>
<td>50</td>
<td>5</td>
<td>120</td>
<td>4</td>
</tr>
<tr>
<td>100</td>
<td>5</td>
<td>120</td>
<td>4</td>
</tr>
<tr>
<td>500</td>
<td>6</td>
<td>720</td>
<td>5</td>
</tr>
<tr>
<td>1000</td>
<td>6</td>
<td>720</td>
<td>5</td>
</tr>
<tr>
<td>10,000</td>
<td>7</td>
<td>50,400</td>
<td>6</td>
</tr>
<tr>
<td>100,000</td>
<td>9</td>
<td>362880</td>
<td>8</td>
</tr>
</tbody>
</table>
source node \((a, b)\) of \(X_1 \otimes X_2\), is to be broadcast to all other nodes. In the first stage, a known broadcast algorithm of \(X_1\) can be used to broadcast \(M\) in \(X_1\) subgraph of \(X_1 \otimes X_2\) that contains all the nodes of the form \((c, b)\) for any node \(c\) in \(X_1\). In other words in the first stage, a copy of \(M\) is delivered to each node that can be reached from the source using only \(X_1\) edges. In the second stage, all nodes of the form \((c, b)\) have received a copy of \(M\) during the first stage, apply broadcast algorithm of \(X_2\) to initiate parallel broadcasts in different \(X_2\) subgraphs of \(X_1 \otimes X_2\). During the second stage, communication is performed along \(X_2\) edges only.

### 2.7.5 Performance Issues of Product Networks

As explained earlier, the cross product of two interconnection networks yields a third interconnection network, so ultimately a product network is also a multiprocessor interconnection network. It is therefore obvious that the factors which are used to determine the performance of any multiprocessor also apply to a product network. They have already been described in Section 2.4.

### 2.8 Network Topology and Load Balancing

The effect of topology on the performance of load balancing schemes have been studied by few researchers including Loh et al. [155]. The following paragraphs outline their observations.

- Loh et al. [155] studied four topologies namely \(4 \times 4\) mesh, 4d-Hypercube, Fibonacci Cube and Linear array under five load balancing schemes. Their performance parameters were normalized performance and stabilization time. The researchers concluded that during load balancing process, the topologies which possessed larger average processor distances and lower average node connectivity introduced significant communication overheads.

- All load balancing schemes performed best with hypercube and Fibonacci cube topologies (because of better node connectivity and better internode
They further showed that varying physical parameters in an interconnection network topology can significantly effect the performance of a load balancing scheme irrespective of the load conditions of the system.

Zhang [239] showed that in the absence of wormhole routing, communication latencies can be a serious problem in parallel systems as not all parallel systems support wormhole routing.

Above findings justify the research undertaken within this thesis for a better topology for parallel and distributed systems.

**Chapter Summary**

In this chapter we have discussed different interconnection networks used for interconnecting multiprocessors. The design issues of interconnection networks, types of interconnection network, permutation network and performance metrics of the interconnection networks are discussed. Classification of interconnection networks based on topology has been reviewed and a new addition namely 'Hybrid Interconnection Networks' has been introduced to the conventional topology based classification to accommodate hierarchical and product networks. Dynamic and static interconnection networks have also been reviewed. In the dynamic interconnection scheme, three main mechanisms have been covered. These are the bus topology, the crossbar topology, and the multistage topology. In static interconnection networks several limited connection interconnection network topologies including hypercube, mesh, ring etc. have been reviewed. Principles of designing product networks have been briefed. The discussion on the contemporary interconnection network is related to the design of limited connection hybrid interconnection networks. The focus of this research is to design a cost-effective and highly scalable topology for massively parallel systems.
The next chapter presents the design and analysis of a new interconnection network called “Scalable Twisted Hypercube (STH)”. 