Chapter

3. Design of Dynamically Reconfigurable Logical Topologies

In this chapter, we study the application of local perturbations paradigm for dynamic reconfiguration of well-known topologies such as Torus and also design a new set of topologies that treat reconfiguration as a logical topology design issue. Section 3.1 gives introduction to the design of dynamically reconfigurable logical topologies and presents local perturbations as a reconfiguration paradigm. In section 3.2, we investigate the application of local perturbations for well-known topologies such as Torus. Section 3.3, first, discusses the need for a Hamiltonian circuit for maintaining connectivity in logical topology and then discusses the design of dynamically reconfigurable logical topologies based on Circulant graphs with edge disjoint Hamiltonian circuits. Finally, section 3.4 compares the dynamically reconfigurable logical topologies designed in this chapter with the well-known logical topologies studied in the previous chapter.

3.1 Introduction

Many regular topologies have been studied in literature as logical topologies for multi-hop fiber optic LAN/MANs. This is mainly because regular topologies facilitate simple routing and use identical hardware. Simple routing is achieved because the interconnection among the nodes is governed by well-defined mathematical functions.

Most of the regular topologies were initially proposed as interconnection networks for parallel computers. When these regular topologies were adopted as logical topologies for fiber optic LANs/MANs, reconfiguration of logical topology for tolerating node faults and additions was not considered as an issue. However, we believe that dynamic reconfiguration is essential for tolerating network changes. Access nodes in fiber optic LANs/MANs that use regular topologies as their logical topology, route the packets to the destination nodes and failure of nodes may disconnects the topology. As the LAN/MANs
use broadcast (physical) topologies as physical interconnection, reconfiguration of logical topology can be done independent of physical topology by (re)tuning wavelengths of nodes' transmitters/ receivers. Further, we assume that reconfiguration should be an integral part of logical topology design. This is because the reconfiguration affects the network performance parameters such as reliability and expandability similar to the way the properties of logical topology such as diameter and average internode distance affects network throughput and packet delays. It is also observed that reconfiguration and properties of topology such as node degree are interdependent.

To facilitate the integration of dynamic reconfiguration into logical topology design, this work proposes Local Perturbations paradigm in the [Reddy and Reddy 2001]. Local Perturbations paradigm assumes a base topology of maximum size, which gives the most affordable performance. Each node has a unique position in the base topology. In this work, local perturbations refer to the changes made to immediate neighbors in the network to accommodate dynamic changes within the network. In other words, whenever a node gets added or deleted from the network, immediate neighbors change their logical links in order to retain the connectivity.

For example, consider node X as connected to \( n \) nodes via incoming links and to \( n \) nodes via outgoing links, as shown in Figure 3.1(a). Failure of node X changes the logical links of immediate neighbors as per a reconfiguration function that retains connectivity by mapping nodes of incoming links to nodes of outgoing links, as shown in Figure 3.1(b). An example of the mapping where an incoming link \( i \) to outgoing link \( j \) is shown in Figure 3.1(c).
Figure 3. Perturbations of Node X

It may be observed from the Figure 3.1(b), that there are $n!$ possible mappings between neighboring nodes of incoming and outgoing links of a failed node. The selection of mapping between neighboring nodes not only needs to maintain connectivity, but also needs to maintain the structural properties of the network such as routing and diameter.

A node can get added according to its position in the base topology and establishes links accordingly. Change of logical link is particularly easy for multichannel lightwave networks as it can be done by tuning transmitters and or receivers of the nodes representing the logical link to a different wavelength.

Local perturbations have the following advantageous.

- The number of link changes is minimal. The number of link changes due to a reconfiguration based on local perturbations is equivalent to the degree of a node.
- Routing tables of immediate neighbors only need to be changed.
- Reconfiguration process takes very little time. Since the links in a regular topology are usually defined as a mathematical function of node ids, identification of neighbor during reconfiguration can also be computed mathematically.
> *Effect of reconfiguration on network traffic is minimum.* The preservation of connectivity by this paradigm minimizes packet loss, delay during the reconfiguration phase.

The following sections discuss the applicability of local perturbations paradigm for dynamic reconfiguration in existing topologies and proposes a set of new topologies that consider dynamic reconfiguration as an integral part of topology design.

### 3.2 Local Perturbations in Existing Topologies

In this section, we investigate the possibility of applying local perturbations as reconfiguration methodology for some of the existing topologies such as Ring, Torus, and Hypercube.

*It is observed that Ring topology is able to bypass faulty components (nodes or links) and it can tolerate unlimited number of faults.* By viewing Ring topology as a Hamiltonian circuit (a circuit which visits every node precisely once), reconfiguration process for bypassing failed nodes can be defined as connecting the previous node with the subsequent one in the network.

*Hypothesis 3.1.* Connectivity in any topology is maintained if reconfiguration is done along a Hamiltonian circuit.

Application of local perturbations to the topologies proposed in the literature, such as Torus, Hypercube, de Bruijn graph and Star graph, is considered as defining reconfiguration functions, from incoming nodes to outgoing nodes, in such a way that routing properties are preserved. Such a reconfiguration function may not guarantee the connectivity in the topology. For example, in Torus topology, each node is connected to four nodes in North, South, East and West directions. Routing between two arbitrary nodes X and Y can be done by moving in row of X to the column of Y and then moving row-wise to reach node Y. Of the different possible mappings among the incoming and outgoing nodes, as shown in Figure 3.2, connecting North-South neighbors and East-West neighbors preserves the routing properties.
However, such a reconfiguration leads to disconnection of the topology. For example, consider the Torus topology with 16 nodes shown in Figure 3.3. By assuming that nodes 2, 3, 5, 6, 7, 8, 9, 12, and 13 are failed and by applying reconfiguration, the resultant structure becomes a disconnected topology as shown in Figure 3.4.
Though reconfiguration along Hamiltonian circuit guarantees connectivity, defining local perturbations such that a Hamiltonian circuit is maintained, becomes a complex task. Moreover, such reconfiguration cannot retain routing properties of the base topology. Hence, additional methods are required for maintaining connectivity in existing topologies. In the following section, we will study Perturbed Torus derived from the base structure, Torus.

3.2.1 Perturbed Torus

Perturbed Torus is a dynamically reconfigurable topology derived from the base Torus structure in which whenever a change in topology occurs, perturbations are made by connecting North-South neighbors and East-West neighbors [Mohan Reddy and Reddy, 1996a]. However, to retain the connectivity of the nodes in the Perturbed Torus network, we propose to maintain a row (column) so that a node shall be present in every column (row) whenever there exists another node in that column (row). This row (column) is called as base row (column).

To understand how connectivity is maintained in Perturbed Torus using base row (column), consider that the network maintains the first row as the base row. Theorem 3.1 proves the preservation of connectivity in Perturbed Torus, with first row as base row, by showing the existence of a path between two arbitrary nodes. The proof can be generalized for a Perturbed Torus that maintains a base column.

Theorem 3.1. Maintenance of base row (column) and perturbations made by connecting North-South neighbors and East-West neighbors preserves the connectivity in the Perturbed Torus.

Proof: Whenever a node exists in a column, according to the concept of base row, there exists a node in the same column of the base row.

Nodes in a row or column of Torus topology are connected in the form of ring. Local Perturbations maintain these rings in rows and columns. It is because perturbations are made by connecting North-South neighbors and East-West neighbors of a deleted (failed) node.

To prove the existence of path between any two nodes, consider two nodes X and Y shown in Figure 3.5. Node Y can be reached from node X by following the steps below.
a) Move in the column ring of node X to the base row,
b) Then move in the base row to the column of the node Y, and finally
c) Move in the column ring of node Y to reach node Y.

Thus there exists a path between X and Y, and hence they are said to be connected. Hence, the connectivity of the network is maintained.

![Diagram of path between two arbitrary nodes X and Y in a Perturbed Torus](image)

*Figure 3.5 Path between two arbitrary nodes X and Y in a Perturbed Torus*

**Reconfiguration.**

Reconfiguration in Perturbed Torus is done when a node is either gets deleted from or added to the base topology. To simplify the discussion on reconfiguration, without loss of generality, we assume row 0 is used as base row. To explain the reconfiguration process, we distinguish the node's id from the node's position in base topology. Node's position is identified with the row and column in the base Torus structure. Usually, in a base Torus structure with $n$ rows and $n$ columns, node $i$ takes the position $(i/n, i%n)$, that is at $i/n$ row and $i%n$ column of the base topology.

Algorithm 3.1 describes the reconfiguration when node with id $i$ is deleted. According to this algorithm, whenever a node in the base row gets deleted, the South neighbor, if present, is replaced into the base row and perturbations are made at the South neighbor of the deleted node. If the deleted node is not in the base row, perturbations are made by connecting its East neighbor with West neighbor and North neighbor with South neighbor.
Algorithm 3.1. Reconfigure the topology when node $i$ is deleted from the Perturbed Torus derived from a base Torus structure with $n$ rows and $n$ columns.

if position of the deleted node $i$ is $(0,i)$ then
  if South neighbor with id $j$ at position $(j \mod n, j \mod n)$ is present
    Move node $j$ to position $(0,i)$ and establishes links with East, West, North and South neighbors at position $(0,i)$.
    Establish links between East and West neighbors and North and South neighbors of position $(j \mod n, j \mod n)$.
  else
    Establish links between East and West neighbors at position $(0,i)$.
  endif
else if the position of deleted node $i$ is $(i,0)$ then
  Establish links between East and West neighbors and North and South neighbors of position $(i,0)$.
endif

To explain the process of node deletion, consider a base Torus topology shown in Figure 3.6(a). Assume that node 5 has been deleted. By connecting neighbors of node 5, we get the resultant topology as shown in Figure 3.6(b). Now, assuming node 1 is deleted, the resultant topology after moving node 9 into the position $(0,1)$ becomes as shown in Figure 3.6(c).

Continuing the perturbations with the loss of nodes 4, 6 and 7 in the perturbed topology shown in Figure 3.6(b), we get the resultant structure equivalent to a regular Torus reduced by a row. Similarly, deletion of all nodes of a column will reduce the structure by one column. The reduced structure behaves as a regular Torus of lower dimension.

Addition of a node in Perturbed Torus is explained in Algorithm 3.2. According to this algorithm, when a node is added, it occupies its default position in the base Torus structure, if there exists a node in the base row whose default position is either in the base row or closer to the base row than the node being added. Otherwise, the node takes the position in the base row, by moving the node, if exists, in the base row to its default position.

To explain the process of node addition, consider Perturbed Torus with 14 active nodes and two deleted nodes (node 1 and 5) shown in Figure 3.6(c). If node 5 wishes to enter the network, it informs the network and occupies position in the base row by moving node 9 to its default position. The resultant structure is shown in Figure 3.6(d). Now
assuming the addition of node 1, the resultant topology becomes as shown in Figure 3.6(a).

Figure 3.6 Reconfiguration in Perturbed Torus with base Torus structure of 16 nodes.
Algorithm 3.2. Reconfigure the topology when node \( i \) is added to the Perturbed Torus derived from a base Torus structure with \( n \) rows and \( n \) columns.

\[
\text{if } i/n = 0 \text{ then}
\]

\[
\text{if there exists a node } j \text{ at position } (0, i) \text{ then}
\]

\[
\text{Move node } j \text{ to position } (jn, j\%n) \text{ where } j\%n = i \text{ and establish links for node } j \text{ with East, West, North and South neighbors at position } (jn, j\%n).
\]

\[
\text{Establish links for node } i \text{ with East, West, North and South neighbors at position } (0, i).
\]

\[
\text{else}
\]

\[
\text{Establish links for node } i \text{ with East, West, North and South neighbors at position } (0, i).
\]

\[\text{endif.}\]

\[
\text{else if } i/n > 0 \text{ then}
\]

\[
\text{if there exists a node } j \text{ at position } (0, i\%n) \text{ then}
\]

\[
\text{if } j/n = 0 \text{ then}
\]

\[
\text{Establish links for node } i \text{ with East, West, North and South neighbors at position } (i/n, i\%n).
\]

\[
\text{else if } (j/n > i/n) \text{ then}
\]

\[
\text{Move node } j \text{ to position } (jn, j\%n) \text{ and establish links for node } j \text{ with East, West, North and South neighbors at position } (jn, j\%n).
\]

\[
\text{Establish links for node } i \text{ with East, West, North and South neighbors at position } (i/n, i\%n).
\]

\[
\text{else then}
\]

\[
\text{Establish links for node } i \text{ with East, West, North and South neighbors at position } (i/n, i\%n).
\]

\[\text{endif.}\]

\[
\text{else}
\]

\[
\text{Establish links for node } i \text{ with East, West, North and South neighbors at position } (0, i\%n).
\]

\[\text{endif.}\]

Routing.

Perturbations are selected in order to preserve the routing properties of Torus topology. Regular Torus uses a simple routing technique - visit the column of the destination node, first, and then visit the destination node by moving in that column. This simple routing is achieved by making use of the rings in rows and columns of a regular Torus. For example, consider the regular Torus shown in Figure 3.6(a). Assume that node 5 wants to send a packet to node 15. To reach the column of node 15, the packet will be routed
through nodes 6 and 7, by moving in the row ring of node 5. Finally, the packet is moved to node 15 through node 11 by moving in the column of node 15.

A simple variation of the above routing technique, explained in Algorithm 3.3, is proposed for Perturbed Torus. According to this algorithm, whenever a source node wants to send a packet to a destination node, the packet first moves in the row of source node to reach the column of the destination node. If the column of the destination node cannot be reached, the packet continues to move in the column nearest to the column of the destination node towards the base row, each time trying to reach the column of the destination node. Finally, the packet reaches the destination node by moving in the column of the destination node.

Algorithm 3.3. Send a message from the current node C to the destination node D which is originated at source node S in the Perturbed Torus derived from a base Torus structure with n rows and n columns.

if (C = -D) then
   Send message to local node C.
else if (C%n = D%n)
   if the neighbor in North-South ring close to D knows that D is not present
      Send negative acknowledgement to the Source indicating that destination node D is failed.
   else
      Move the packet in North-South ring to the neighbor close to D than C.
   endif
else if neighbor at position (C’n, C%n) in East-West ring which is close to I) is present then
   Move the packet in East-West ring to the neighbor close to D than C.
else if the neighbor in East-West ring close to D knows that D is not present then
   Send negative acknowledgement to the Source indicating that destination node D is failed.
else
   Move towards the base row.
endif

For example, in the Perturbed Torus of 15 nodes with node 5 as a deleted node, shown in Figure 3.6(b), let us assume node 7 wants to send a packet to node 13. The packet first moves in the row of node 7 to node 6. At node 6, since node 5 is failed, the packet will be routed to the base row to node 2. From node 2, the packet will be sent to node 1 which is in the column of the destination 13, from there the packet reaches the destination node.
node 13. As another example, consider a packet addressed to node 5 from node 15 will be routed to node 13 through node 14. Then from node 13, the packet will reach node 9. At node 9, the packet will be discarded as node 9 finds its north direction link connected to node 1 instead of node 5. An acknowledgement packet will be sent by node 9 to node 15 about the status of node 5.

Thus whenever a source node wishes to communicate to a destination node, it just sends the packet assuming that the destination node is present. If the destination node is not present, the neighboring nodes send acknowledgement packet indicating the status of the destination node. Hence, each node need not keep the status of all nodes in the network. It is sufficient to keep the status of neighboring nodes.

Diameter.

The following theorem proves the maximum diameter of a Perturbed Torus derived from a base Torus structure of size N is $2 \times \lceil \sqrt{N} / 2 \rceil + 1$.

**Theorem 3.2.** The maximum diameter of a Perturbed Torus derived from a Torus of N nodes is $2 \times \lceil \sqrt{N} / 2 \rceil + 1$.

**Proof:** Consider any two arbitrary nodes X and Y in the Perturbed Torus. For simplicity, we assume that base row is maintained. The same can be easily proved for maintaining base column.

**Case I.** Let us assume both of the nodes X and Y belong to either the same row or the same column. Then as the nodes in the same column or row of Perturbed Torus are connected in the form of bi-directional ring of maximum size $\lceil \sqrt{N} \rceil$, the distance between X and Y is less than or equal to $\lceil \sqrt{N} / 2 \rceil$.

**Case II.** Let the nodes X and Y belong to different rows and columns. If there exists a node Z such that either row(Z) = row(X) and column(Z) = column(Y) or row(Z) = row(Y) and column(Z) = column(X). Then we have a path from node X to node Y through node Z. The distance between the nodes X and Z is less than or equal to $\lceil \sqrt{N} / 2 \rceil$. This is because they belong to the same row or column of size $\lceil \sqrt{N} \rceil$. Similarly, the distance between the nodes Z and Y is less than or equal to $\lceil \sqrt{N} / 2 \rceil$ as both of them belong to the same column or row of size $\lceil \sqrt{N} \rceil$. Hence the distance between the nodes X and Y, which
is the sum of the distance between the nodes X and Z and the distance between the nodes 
Z and Y, is less than or equal to \(2^*\lceil \sqrt{N}/2 \rceil\).

Case III. Let the nodes X and Y belong to different rows and columns. Also assume that 
there exists no node Z such that either it is in the column of node X and row of node Y, or 
it is in the row of node X and column of node Y.

Let X' and Y' are the nodes in the base row corresponding to the columns of nodes X and 
Y, respectively. Assume no other pair of nodes in columns of nodes X and Y belongs to 
the same row. Then the total number of nodes in the columns of X and Y is less than or 
equal to \(\lceil \sqrt{N} \rceil +1\). Let m and n be the number of nodes in the columns of nodes X and Y, 
respectively. Then \(m+n \leq \lceil \sqrt{N} \rceil +1\). The distance between the nodes X and X'
is less than or equal to \(m/2\) and the distance between the nodes Y and Y' is less than or equal to \(n/2\).

Hence the sum of the distance between nodes X and X', and the distance between nodes 
Y and Y' is less than or equal to \((m+n)/2\) which is less than or equal to \(\lceil \sqrt{N}/2 \rceil +1\). The distance between X' and Y' is less than or equal to \(\lceil \sqrt{N}/2 \rceil \). Hence the distance 
between the nodes X and Y, which is the sum of the distance between nodes X and X', 
the distance between nodes X' and Y', and the distance between nodes Y' and Y, is less 
than or equal to \(2^*\lceil \sqrt{N}/2 \rceil +1\).

Suppose that X'' and Y'' are the nodes in the columns of nodes X and Y that belong to the 
same row and are nearer than the nodes X' and Y' from nodes X and Y, respectively. 
Also assume that no other pair of nodes are nearer than the nodes X'' and Y'' from nodes 
X and Y, respectively. Then the sum of the distances between nodes X and X'', and the 
distance between nodes Y and Y'' is less than the sum of the distance between nodes X 
and X', and the distance between nodes Y and Y', i.e., \(\lceil \sqrt{N}/2 \rceil +1\). Also the distance 
between the nodes X'' and Y'' is less than or equal to \(\lceil \sqrt{N}/2 \rceil \).

Hence the distance between X and Y, which is the sum of the distance between nodes X 
and X'', the distance between nodes X'' and Y'', and the distance between nodes Y'' and 
Y, is less than or equal to \(2^*\lceil \sqrt{N}/2 \rceil +1\).

Therefore the maximum diameter of Perturbed Torus is \(2^*\lceil \sqrt{N}/2 \rceil +1\).
It may be noted that Perturbed Torus retains simple routing properties of Torus structure and also has the maximum diameter equivalent to that of base Torus topology.

It may be observed that defining reconfiguration using local perturbations in traditional topologies may lead to disconnected graph upon the application of reconfiguration for successive node failures.

To understand further, consider Binary Hypercube topology as a base topology. As the neighbor nodes in Binary Hypercube differ by one bit, packets are routed along the neighbors that reduce the bit distance to the destination. To preserve this routing property, reconfiguration due to node failure should change the links of neighbors so that the bit distance between neighbors is always limited to one. This leaves with the choice of reconfiguration where neighbor nodes develop self-loops along the direction of failed node. Application of reconfiguration due to the failure of all neighbors in a Binary Hypercube would cause isolating the node from rest of the nodes in the network.

Defining special methods similar to the one used in Perturbed Torus (maintaining base row), so that the reconfiguration always maintain connectivity and preserves structural properties in traditional topologies, is found to be infeasible. This is because such special methods complicate the reconfiguration process, which is against the spirit of local perturbations.

Hence, we investigated novel base topologies that allow local perturbations for maintaining connectivity while preserving other structural properties. In the following section, we discuss a subset of Circulant graphs that are constructed with edge-disjoint Hamiltonian circuits and design reconfigurable topologies using local perturbations paradigm.
3.3 Local Perturbations in Circulant Graphs with Edge-disjoint Hamiltonian Circuits

As studied in the previous section, reconfiguration along a Hamiltonian circuit in a logical topology guarantees network connectivity. This fact motivated us to consider Circulant graph as a base topology since it is a set of rings with at least one Hamiltonian circuit. In this section, we study a subclass of Circulant graphs constructed with a set of edge-disjoint Hamiltonian circuits. These Circulant graphs are used as base topologies for the dynamically reconfigurable topologies proposed in this thesis. We start our discussion by first defining Circulant graph.

Circulant Graph. A \( p \)-node graph is Circulant with respect to a given set \( S \) of integers, if each node has a unique label in the range 0 through \( p-1 \) and each node \( x \) is connected to all nodes with labels \( (x + s) \mod p \) where \( s \in S \). We call the set \( S \) as connection set and the members of the set as jumps or offsets. The \( p \)-node Circulant graph with connection set \( S \) is denoted by \( C_{p,S} \).

An example of Circulant graph with 8 nodes and a connection set \( \{+1,-1,-3,+3\} \) is shown below.

![Circulant graph with 8 nodes and connection set \{-3,-1,1,3\}](image)

In the figure above, we can identify the paths associated with a jump as the path obtained with edges representing that particular jump. For example, the path associated with jump +1 is 0,1,2,3,4,5,6,7,0. Similarly, the path associated with jump -3 is 0,5,2,7,4,1,6,3,0. Interestingly, these paths represent Hamiltonian circuits in the graph. The following
Theorem 3.3. Let \( V_0, V_1, \ldots, V_p \) be the sequence of numbers satisfying the following conditions.

1. \( V_i = (V_{i-1} + s) \pmod{p} \)
2. \( V_0 < p; s < p \)
3. \( s \) is relatively prime with respect to \( p \).

Then all the elements in the sequence are distinct except for \( V_0 = V_p \).

**Proof:** By definition of \( V_i = (V_{i-1} + s) \pmod{p} \), we can write \( V_i = V_{i-1} + s - k_i * p \) for some \( k_i \geq 0 \). By applying this definition recursively, we get

\[
V_i = V_0 + i * s - K_i * p = V_0 + j * s - K_j * p
\]

where \( K_i = \sum_{m=1}^{i} k_m \). Let us assume \( V_i = V_j \) for some \( i \) and \( j \). Using Equation (3.1), we get

\[
V_0 + i * s - K_i * p = V_0 + j * s - K_j * p
\]

This implies

\[
(i - j) * s = (K_i - K_j) * p
\]

Since \( s \) is relatively prime with respect to \( p \), the above Equation (3.2) is valid, i.e., two numbers of the sequence \( V_i \) and \( V_j \) are equal, only if \( (i-j) \) is a multiple of \( p \). Since \( i \) and \( j \) can take values in the range 0 to \( p \), \( (i-j) \) is a multiple of \( p \) only for \( i = 0 \) and \( j = p \). Hence all the numbers of the sequence are distinct except for \( V_0 = V_p \).

\[ \bullet \]

Theorem 3.4. In a Circulant graph \( C_{p,s} \), if a jump, \( s \in S \), is relatively prime with respect to \( p \), then the path associated with jump \( s \) forms a Hamiltonian circuit.

**Proof:** Consider any node \( V_0 \) and the sequence of nodes \( V_0, V_1, \ldots, V_p \), which satisfies the condition: \( V_i = (V_{i-1} + s) \pmod{p} \), where \( s \in S \) and \( s \) is relatively prime with respect to \( p \). The sequence is a Hamiltonian circuit for the following two reasons.

Firstly, the sequence is a path. This is because any two successive nodes of the sequence are adjacent (By the definition of Circulant graph).
Secondly, because $s$ is relatively prime with respect to $p$, all the nodes in the sequence are distinct except for $V_0=V_p$ (according to Theorem 3.3).

In the Circulant graph of Figure 3.7, all the four jumps of the connection set $S$ are relatively prime with respect to $p$. Hence, according to the Theorem 3.4, we have four Hamiltonian circuits, each one being associated with a jump. Moreover, these Hamiltonian circuits are edge disjoint as the jumps of the connection set $S$ are distinct. This means that the Hamiltonian circuits associated with different jumps, in the connection set $S$ of the circulant graph $C_{p,S}$, are edge disjoint.

Hypothesis 3.2. The Hamiltonian circuits associated with different jumps in the connection set $S$ of the circulant graph $C_{p,S}$ are edge disjoint.

In the following subsections, we study dynamically reconfigurable topologies that use three different Circulant graphs with edge-disjoint Hamiltonian circuits as the base topologies. These base topologies are akin to 2-D Torus, n-D Torus and Binary Hypercube respectively, in their performance.

3.3.1 Reconfigurable Circulant Graph - T

In this section, we design a dynamically reconfigurable topology derived from the base Circulant graph $C_{p,S}$ with $p=n^2-1$ and $S = \{-n,-1,1,n\}$ which reconfigures as per the local perturbations paradigm [Mohan Reddy and Reddy, 1996b]. The base Circulant graph $C_{p,S}$ with $p=n^2-1$ and $S = \{-n,-1,1,n\}$ is referred as Circulant Graph-I or CG-I, in the rest of the thesis. Before designing this reconfigurable topology, we first study the properties of the base Circulant graph in order to prove that the structural properties of base topology are retained.

An example of the base CG-I topology, $C_{p,S}$ with $p=8$ and connection set $S = \{-3,-1,1,3\}$ is shown in Figure 3.7. A mesh-like representation of the same is shown in Figure 3.8.
Before studying some more properties of this base topology, first we will define an image of anode.

**Image of a Node.** X and Y are said to be images of each other, iff $X \equiv_{n} Y$.

In the CG-I topology shown in Figure 3.8, node 0, 3, and 6 are said to be images of each other. Similarly, nodes 2 and 5 are said to be images of each other. It is observed from Figure 3.8 where the nodes are represented in 2-D Euclidean space that an image of node can be reached in maximum of $n$ hops along the jumps +1 or -1.

![Figure 3.8 Mesh-like representation of CG-I topology with p 8 and S {-3,-}

**Hypothesis 3.3.** In the CG-I topology, $C_{p,s}$, an image of a node can be reached in maximum of $n$ hops along the jumps +1 and -1.

It may be observed from the Figure 3.7 and Figure 3.8 that there exist four edge-disjoint Hamiltonian circuits, each being associated with a jump in the connection set S. Corollary 3.1 proves the existence of such Hamiltonian circuits in the base Circulant graph.

**Corollary 3.1.** There are four edge-disjoint Hamiltonian circuits in CG-I topology, $C_{p,s}$ with $p \equiv_{n}^{-1}$ and $S = \{-n,-1,1,n\}$

**Proof:** According to Hypothesis 3.2, since all the jumps in the connection set S are distinct, it is sufficient to show that the paths associated with jumps are Hamiltonian.
circuits. Further, according to Theorem 3.4, it is sufficient to prove that all these jumps are relatively prime with respect to $p$.

$+1$ and $-1$ are relatively prime with respect to $p$. Suppose $n$ is not relatively prime with respect to $p$. Then $p/n$ is an integer, say, $k$. But $p/n = (n^2-1)/n$ is not an integer. Hence, there is a contradiction in our assumption. Therefore, $n$ has to be relatively prime with respect to $p$. By similar argument, we can show that $-n$ is relatively prime with respect to $p$. Hence, the Hamiltonian circuits associated with four different jumps are edge disjoint.

**Routing in base Circulant Graph-I.**

A distributed shortest path routing method in the base Circulant Graph-I topology is described in the following Algorithm 3.4.

**Algorithm 3.4.** Send a message from the current node $C$ to the destination node $D$ which is originated at source node $S$ in a CG-I topology, $C$ with $p\ n\ n - 1$ and connection set $S = \{-n, -1, +1, +n\}$.

if ($C = D$) then
Send message to local node $C$
else
Compute the distance and jump to move on.
Distance $= (D-C+p) \mod p$
if(Distance $\mod n \ n/2$)
Move along jump $-1$.
else if(Distance $\mod n = 0$)
if(Distance $\geq p/2$)
Move along jump $-n$.
else
Move along jump $+n$.
endif
else
Move along jump $+1$.
endif
endif

Algorithm 3.4 describes routing a packet by making use of the rings present in the mesh-like representation of Circulant graph. It may be observed that Hamiltonian circuits associated with $+1$ and $-1$ jumps form a bi-directional ring. Similarly, Hamiltonian
circuits associated with \(+n\) and \(-n\) jumps form another bi-directional ring. To reach the destination node, a packet first moves along the bi-directional ring of jumps \(-1\) and \(+1\), and then along the bi-directional ring of jumps \(+n\) and \(-n\).

**Diameter of base Circulant Graph-I.**

The diameter of Circulant Graph-I is found to be \(n\). Theorem 3.5 proves this.

*Theorem 3.5. The diameter of Circulant graph \(C_{p,s}\) with \(p=n^2-1\) and connection set \(S=\{-n,-1,+1,+n\}\).*

**Proof:** According to Corollary 3.1, there exist four Hamiltonian circuits, along the jumps \(-n\), \(-1\), \(+1\) and \(n\). The Hamiltonian circuits along the jumps \(+1\) and \(-1\) together form a bi-directional ring while Hamiltonian circuits along the jumps \(+n\) and \(-n\) together form another bi-directional ring. The maximum hop-distance between any two arbitrary nodes along the bi-directional ring of jumps \(+1\) and \(-1\) is \((n^2-1)/2\). This hop-distance is further reduced by making use of \(+n\) and \(-n\) jumps.

Let the hop-distance, \(d\), between two arbitrary nodes \(X\) and \(Y\) on bi-directional ring of jumps \(+1\) and \(-1\) be expressed as \(d=i+n+j\), where \(j<n\). This means that the distance \(d\) can be reached by making \(i\) hops along the bi-directional ring of jumps \(+n\) and \(-n\) and \(j\) hops along the bi-directional ring of jumps \(+1\) and \(-1\).

It may be observed from Figure 3.8 that there exists bi-directional rings of size \((n+1)\) formed with \(n\) links of jumps \(+1\) (or \(-1\)) and one link of jump \(-n\) (or \(+n\)). Hence, the hop-distance between \(X\) and \(Y\) can be expressed as

\[
h_{(XY)} = \begin{cases} 
  i + j, & \text{if } j \leq n/2 \\
  (i+1) + \left(j - n/2\right), & \text{otherwise}
\end{cases}
\]  

(3.3)

As the maximum value of \(d\) is \((n^2-1)/2\), the maximum value of \(i\) is \(\left\lfloor (n^2-1)/(2*n)\right\rfloor < n/2\). Hence, the maximum value of \(h_{(X,Y)}\) is \((n/2)+(n/2) = n\). (Since, the number of hops with jumps of magnitude 1 is \#/2.)

Thus the diameter of \(CG-I\) topology, \(C_{p,s}\) with \(p=n^2-1\) and \(S=\{-n,-1,+1,+n\}\), the maximum of hop-distance between all pair of nodes is \(n\). 

64
[91x758]may be observed that the Circulant Graph-I, \( C_{p,s} \) with \( p = n^2 - 1 \) and \( S = \{-n, -1, +1, +n\} \) can be considered equivalent to 2-D Torus. This is because the CG-I topology also has a fixed node degree four and a diameter of \( \sqrt{p + 1} \), where \( p \) is the number of nodes in the network.

Having discussed the properties of base Circulant graph, we shall now define the Reconfigurable Circulant Graph-I (RCG-I). This reconfigurable topology is designed to preserve the properties of the base Circulant Graph-I, \( C_{p,s} \) with \( p = n^2 - 1 \) and

**Reconfigurable Circulant Graph-I.** Let the Circulant Graph-I, \( C_{p,s} \) with \( p = n^2 - 1 \) and \( S = \{-n, -1, +1, +n\} \) be the base topology and \( q \) be any positive integer such that \( q < p \).

The q-node Reconfigurable Circulant Graph-I (RCG-I) with connection set \( S = \{-n, -1, +1, +n\} \), denoted by \( R_{q,s} \), consists of \( q \) nodes. Each node in \( R_{q,s} \) has unique label in the range 0 through \( p \) [Mohan Reddy and Reddy, 1996b]. Each node \( i \) is connected to \( (i + j * s) \mod p \) where \( s \in S \) and \( / \) be a positive integer such that for any positive integer \( k < j \), nodes with labels \( (i + k * s) \mod p \) are failed.

Figure 3.9 shows an example of a 7-node RCG-I derived from a base CG-I topology, \( C_{p,s} \) with \( p = 8 \) and \( S = \{-3, -1, +1, +3\} \) and with a failed node 4.
Reconfiguration in RCG-I.

The reconfiguration of RCG-I is explained as follows. Whenever a node fails, the neighboring nodes connect to the next node on the corresponding Hamiltonian circuit of jumps through which the failed node is connected. Similarly, when a node recovers from failure, it occupies its position in the base circulant graph. Algorithm 3.5, shown below, explains the computation of new neighbor nodes.

Algorithm 3.5. Finding a new neighbor node for node \( i \) in the direction of jump \( s \in S \). \( p \) is the size of the base circulant graph.

\[
\text{for}(m=1; m<p; m++)
\]

\[
\text{if node with id \((i\times m \times s \mod p)\) is active}
\]

\[
\text{Set node \((i \times m \times s \mod p)\) as neighbor}
\]

\[
\text{break}
\]

\[
\text{endif.}
\]

\[
\text{endfor.}
\]

It may be observed that the above reconfiguration algorithm can be executed in a distributed manner to simultaneously result the new logical topology. Also the number of link changes is minimum that can be achievable by any reconfiguration algorithm.

To explain the process of reconfiguration in case of a node failure, consider a base topology shown in Figure 3.7. Assume that node 4 is failed. By connecting the neighbors of failed node to next nodes on the corresponding Hamiltonian circuits, we get the resultant topology as shown in Figure 3.9. Continuing the reconfiguration with failure of node 3 results in the structure shown in Figure 3.10.

![Figure 3.10 RCG-I with base topology C'_{p,s} with p 8 and S={-3,-1, + 1, + 3} and failed nodes 3,4.](image)
To explain the process of node addition, consider RCG-I with six active nodes and two failed nodes 3 and 4 as shown in Figure 3.10. If node 3 wishes to enter the network, it informs the network and occupies its position in the base circulant graph by forming links with its neighbors, i.e., with nodes 2, 5, 6, and 0. This results in the structure shown in Figure 3.9.

**Routing in RCG-I.**

A distributed routing method in RCG-I, $R_{q,s}$ with $q \leq p = n^2 - 1$ and $S = \{-n, -1, 1, +n\}$ is given in Algorithm 3.6. According to this algorithm, a packet is routed by making use of the rings present in the mesh-like representation of Circulant graph. To reach the destination node, a packet first tries to move along the Hamiltonian circuit of jump +1 when the reminder of distance on Hamiltonian circuit of jump +1 divided by $n$ is not zero. Otherwise, the packet is routed along the bi-directional ring of jumps $+n$ and $-n$.

To understand routing, consider the RCG-I, $R_s$ with six active nodes and two failed nodes with ids 3 and 4, shown in Figure 3.10. For example, assume node 0 wants to send a packet to node 4. The packet is first routed to node 1 from here it learned that node 4 is not present. Node 1 then informs node 0 about the absence of node 4. As another example, assume that node 1 has a packet destined to node 7. Since the distance to node 7 is 6, the packet is routed on jump +3. Incidentally, the destination node 7 is the neighbor on this jump.

Let us see another example where node 0 has a packet destined to node 5. Though node 5 is a neighbor to node 0, the packet takes the route 0 to 7 to 2 to 5. This shows that the routing algorithm doesn’t guarantee shortest path. By incorporating the knowledge about the neighbors, a better algorithm can be designed.

However, it is not possible to design a shortest path routing algorithm for a reconfigurable topology that reconfigure using local perturbations because nodes doesn’t have the knowledge of the structure of the topology at a given point of time.

**But, packets are guaranteed to reach the destination node, if it is active.** This is because any node can be reached on the Hamiltonian circuit of jump +1, provided it is active, and
this routing algorithm always tries, first, to move along the Hamiltonian circuit of jump +1.

**Algorithm 3.6.** Send a message from the current node $C$ to the destination node $D$ which is originated at source node $S$ in a RCG $R_{q,S}$ with $q \leq p - n^2$ and $S = \{-n, -1, 1, +n\}$.

```plaintext
if (C - D) then
    Send message to local node C
else
    // Compute the distance and jump to move on.
    Distance = (D - C + p) % p
if (Distance % n = 0)
    if (Distance > p/2)
        if ((p - Distance) % n = 0)
            Send negative acknowledgement to the source $S$ indicating that the destination node $D$ is failed
        else
            Move along jump -n.
    else
        Move along jump -l.
endif
else
    if ((Distance % n = 0)
        Send negative acknowledgement to the source $S$ indicating that the destination node $D$ is failed
    else
        Move along jump -l.
endif
else
    if ((Distance % n = 0)
        Send negative acknowledgement to the source $S$ indicating that the destination node $D$ is failed
    else
        Move along jump -l.
endif
endif
endif
```

**Diameter of RCG-I.**

The maximum diameter of RCG-I is found to be $3n/2$. Theorem 3.6 proves this.

**Theorem 3.6.** The maximum diameter of RCG, $R_{q,S}$ with $q \leq n - 1$ and $S = \{-n, -1, 1, +n\}$, is less than $3n/2$.

**Proof:** According to theorem 3.5, the maximum hop-distance between any two arbitrary nodes $X$ and $Y$, $h(X,Y) \leq n$.  

68
As the nodes between X and Y keep failing, some of the nodes along jumps of magnitude $n$ may not be accessible. This leads to movement in the bi-directional ring of jumps +1 and -1. Let us assume that Y is reached from X using Algorithm 3.6. Let $k$ be the number of hops required to reach Y from X. Then, Y can be expressed in terms of the hops from X as

$$Y = X + h_1 + h_2 + \cdots + h_k (n^2 - 1)$$

(3.4)

where $h_i$ is an integer multiple of +1, +n or -n.

Without loss of generality, we assume $X < Y$ and $(Y-X) < (n^2-1)/2$. Then Y is reached from X using jumps +n and +1 and Y is expressed as sum of hops from X as,

$$Y = (X + h_1 + h_2 + \cdots + h_k)$$

(3.5)

Out of the $k$ hops, let us assume that $m$ hops are made along the bi-directional ring of jumps +n and -n. Then $m < n/2$. The remaining hops ($k-m$) are made along the Hamiltonian circuit of jump +1. Let us assume that ($k-m$) > $n$ Then, out of ($k-m$) hops, there exists $a$, $b$ and $c$ such that

$$\sum_{i=a}^{b} h_i = c \times n$$

(3.6)

and

$$(b-a) > c > l$$

(3.7)

This is because, according to Hypothesis 3.3, there exists an image of a node in every $n$ hops along jump +1 or -1. Thus hops $h_a$ to $h_b$ can be reduced to at most $c$ hops along bi-directional ring of jumps +n and -n. By applying Hypothesis 3.3 repeatedly, the number of hops along the Hamiltonian circuit of jump +1, $k-m$, can be reduced to less than $n$. However, the hops $m$ along the bi-directional ring of jumps +n and -// remain less than $n/2$ due to the fact that $(Y-X) < (n^2-1)/2$.

Therefore, the number of hops, $k$, required to reach an arbitrary node Y from an arbitrary node X is less than $3n/2$.

Thus the maximum diameter of RCG-I, $R_{q,s}$ with $q < n^2/2$ and $S = \{-n,-l,1,+n\}$, i.e., the maximum of hop-distance between all pair of nodes is less than $3n/2$. 

•
Though we could prove the maximum diameter of RCG-I, $R_{q,s}$ with $q < n^2 - 1$ and $S = \{-n, -1, 1, +n\}$ is less than $3\sqrt{n}/2$, it is observed empirically in Chapter 4 that the maximum diameter is $n$. It is probably because $(k-m) > n/2$ jumps along Hamiltonian circuit of jump +1 can be replaced with one jump along the bi-directional ring $+n$ and $-n$ and $n-k+m$ jumps along the bi-directional ring +1 and -1. This is evident from the Figure 3.8 that $n$ hops along the bi-directional ring of jumps -1 and +1 (or $-n$ and $+n$) can be replaced with one hop along the bi-directional ring of jumps $-n$ and $+n$ (or $-1$ and $+1$).

The values of $n, p$ and $S$ in RCG-I are selected not only to keep $p$ as relatively prime with respect to the connection set $S$, but also to achieve the performance similar to a 2-D Torus structure. The facts that 2-D Torus structure gives maximum performance when the total number of nodes are $n^2$ and the links of a node are identified by jumps of length +1, -1, +$n$ and -$n$, assuming that the nodes in a 2-D Torus are represented as integers instead of their position in the grid (row and column), motivated the design of RCG-I.

The successful design of this RCG-I provided an insight into construction of a base Circulant graph that is on par with n-D Torus. In the following section, we design a dynamically reconfigurable topology derived from a base Circulant graph that is equivalent to n-D Torus.

### 3.3.2 Reconfigurable Circulant Graph - II

This section presents another dynamically reconfigurable topology derived from the base Circulant graph $C_{p,S}$ with $p = k^n - 1$ and $S = \{-k^{n-1}, -k^{n-2}, \ldots, -1, 1, \ldots, +k^{n-2}, +k^{n-1}\}$. We refer this base Circulant graph as Circulant Graph-II or CG-II and study its properties in order to prove the reconfigurable topology designed retains the structural properties. Circulant Graph-II is different from Circulant Graph-I in terms of connection set, $S$ and the number of nodes, $p$.

An example of the base Circulant graph $C_{p,S}$ with $p = 26$ and connection set $S = \{-9, -3, -1, 1, +3, +9\}$ is shown in Figure 3.11.
By visualizing this graph in n-dimensional Euclidean space where each node is identified with an n-digit k-ary number, links of magnitude k are treated as moving in /-th dimension. First, let us define an image of a node.

**Image of a Node.** X and Y are said to be images of each other in /-th dimension, if and only if $X \% k' = Y \% k'$.

In the CG-II topology shown in Figure 3.11, node 0, 3, and 6 are said to be images of each other. Similarly, nodes 10 and 19 are said to be images of each other. It may be observed from Figure 3.11 that an image of node along the jumps $+k$ or $-k$ can be reached in a maximum of $k$ hops.

**Hypothesis 3.4.** In the Circulant Graph-II, $C_{p,S}$, an image of a node along the jumps $k^1$ and $-k^1$ can be reached in a maximum of $k$ hops.

It may be observed from the Figure 3.11 that there exist six edge-disjoint Hamiltonian circuits, each one being associated with a jump in the connection set $S$. Corollary 3.2 proves the existence of such Hamiltonian circuits in the base Circulant graph.
Corollary 3.2. There are $2^n$ edge-disjoint Hamiltonian circuits in $CG-II$, $C_{p,S}$ with $p=k^n-1$ and $S=\{-k^{n-1}, -k^{n-2}, \ldots, -1, 1, \ldots, k^{n-2}, k^{n-1}\}$. 

Proof: According to Hypothesis 3.2, since all the jumps in the connection set $S$ are distinct, it is sufficient to show that the paths associated with jumps are Hamiltonian circuits. Further, according to Theorem 3.4, it is sufficient to prove that all these jumps are relatively prime with respect to $p$.

$+1$ and $-1$ are relatively prime with respect to $p$. Suppose $k', 0 \leq i < n-1$, is not relatively prime with respect to $p$. Then $p/k'$ is an integer. But $p/k'=(k^n-1)/k'$ is not an integer. This contradicts our earlier assumption. Therefore, $k'$ has to be relatively prime with respect to $p$. By similar argument, we can show that $-k'$ is relatively prime with respect to $p$. Hence, the Hamiltonian circuits associated with $2^n$ different jumps are edge disjoint.

Routing in base Circulant Graph-II.

A distributed shortest path routing method in the Circulant Graph-II $C_{p,S}$ with $p=k^n-1$ and $S=\{-k^{n-1}, -k^{n-2}, \ldots, -1, 1, \ldots, k^{n-2}, k^{n-1}\}$ is given in Algorithm 3.7.

Algorithm 3.7. Send a message from the current node $C$ to the destination node $D$ which is originated at source node $S$ in a Circulant Graph-II, $C_{p,S}$ with $p=k^n-1$ and connection set $S=\{-k^{n-1}, -k^{n-2}, \ldots, -1, 1, \ldots, k^{n-2}, k^{n-1}\}$.

if $(C=D)$ then
  Send message to local node $C$
else
  // Compute the distance and jump to move on.
  Distance = $(D-C+p) \mod p$
  jump = 1
  while (Distance $\mod (jump*k) = 0$)
    jump = jump $\times k$
  endwhile.
  if (Distance $\mod (jump*k) > (jump*k/2))$
    jump = $(-1) \times jump$
  endif.
endif.

According to this algorithm, a packet is routed by making use of the bi-directional rings present in the Circulant Graph-II. It may be observed that Hamiltonian circuits
associated with +1 and -1 jumps form a bi-directional ring. Similarly, Hamiltonian circuits associated with +k' and -k' jumps, 0<i<n-l, also form bi-directional rings. To reach the destination node, a packet first moves along the bi-directional ring of jumps -1 and +1, then along the bi-directional ring of jumps +k and -k, then along the bi-directional ring of jumps +k^2 and -k^2, and so on.

**Diameter of base Circulant Graph-II.**

The diameter of Circulant Graph-II is found to be n*k/2. Theorem 3.7 proves this.

*Theorem 3.7.* The diameter of Circulant Graph-II C_{p,S} with p = k^n-1 and connection set S={-k^{n-1},-k^{n-2},...,-l,..,+l,..,+k^{n^2},+k^{n-1}} is n*k/2.

*Proof:* The maximum hop-distance between any two arbitrary nodes along the bi-directional ring of jumps +1 and -1 is (k^n-1)/2. The hop-distance is further reduced by making use of jumps of higher magnitude, i.e., +k,-k,+k^2,-k^2,..,+k^{n-1} and -k^{n-1}.

Let h be the hop-distance between two arbitrary nodes X and Y, on bi-directional ring of jumps +1 and -1. Let h be expressed in k-ary number system as h=h_{n-1}h_{n-2}...h_0, where 0<h_i<k, \forall i<n. Each bit h_i≠0, i<n in h can be reduced to zero by moving along the bi-directional ring of +k' and -k', in a maximum of k/2 hops. Thus, all the bits of h can be reduced to zero, i.e., the distance h can be reached in a maximum of n*k/2 hops.

Hence, the diameter of CG-II, C_{p,S}, with p = k^n-1 and S={-k^{n-1},-k^{n-2},...,-l,..,+l,..,+k^{n^2},+k^{n-1}}, the maximum of hop-distance between any pair of nodes is n*k/2.

It may be observed that the Circulant Graph-II, C_{p,S} with p = k^n-1 and S={-k^{n-1},-k^{n-2},...,-l,..,+l,..,+k^{n^2},+k^{n-1}} can be considered at par with n-D Torus. This is because the Circulant graph C_{p,S} also has a node degree of 2*n and a diameter of n*k/2, where p = k^n-1 is the number of nodes in the network.

Having been discussed the properties of base Circulant Graph-II, we shall now define the reconfigurable topology, Reconfigurable Circulant Graph-II (RCG-II). This topology is designed to preserve the properties of the base Circulant Graph-II.
Reconfigurable Circulant Graph-II. Let the Circulant Graph-II, \( C_{p,S} \) with \( p=k^n-1 \) and \( S=\{-k^n, -k^{n-1}, \ldots, -1, 1, \ldots, +k^n, +k^{n-1}\} \) be the base topology and \( q \) be any positive integer such that \( q < p \). The \( q \)-node Reconfigurable Circulant Graph (RCG-II) with connection set \( S=\{-k^n, -k^{n-1}, \ldots, -1, 1, \ldots, +k^n, +k^{n-1}\} \), denoted by \( R_{q,S} \), consists of \( q \) nodes. Each node in \( R_{q,S} \) has unique label in the range 0 through \( p-1 \). Each node \( i \) is connected to \((i+j*s)\mod p\) where \( s \in S \) and \( j \) be a positive integer such that for any positive integer \( k < j \), nodes with labels \((i+k*s)\mod p\) are failed.

Figure 3.12 shows an example of a 25-node RCG-II derived from a base Circulant Graph-II, \( C_{p,S} \) with \( p=26 \) and \( S=\{-9, -3, -1, 1, +3, +9\} \) and with a failed node 4.

![Figure 3.12 RCG-II with base topology \( C_{p,S}, p=26 \) and \( S=\{-9, -3, -1, 1, +3, +9\} \) and failed node 4.](image)

Reconfiguration in RCG-II.

The reconfiguration of RCG-II is similar to that of RCG-I, as described in Algorithm 3.5. Whenever a node fails, the neighboring nodes connect to the next node on the corresponding Hamiltonian circuit of jump \( s \) through which the failed node is connected.
Similarly, when a node recovers from failure, it occupies its position in the base circulant graph.

To explain the process of reconfiguration in case of a node failure, consider a base topology shown in Figure 3.11. Assume that node 4 is failed. By connecting the neighbors of failed node to next nodes on the corresponding Hamiltonian circuits, we get the resultant topology as shown in Figure 3.12. Continuing the reconfiguration with failure of node 3 results in the structure shown in Figure 3.13.

To explain the process of node addition, consider RCG-II with six active nodes and two failed nodes 3 and 4 as shown in Figure 3.13. If node 3 wishes to enter the network, it informs the network and occupies its position in the base topology by forming links with its neighbors, i.e., with nodes 2, 5, 6, 0, 12 and 23. This results in the structure shown in Figure 3.12.

Routing in RCG-II.

A distributed routing method in RCG-II $R_{q,s}$ with $q < p = k^n - 1$ and $S = \{-k^{n-1}, -k^{n-2}, \ldots, -1, +1, \ldots, +k^{n-2}, +k^{n-1}\}$ is given in Algorithm 3.8. According to this...
algorithm, a packet is routed along a bi-directional ring of jumps \( +k \) and \(-k\) for which the reminder of distance divided by \( k \) is not zero and there exists no other bi-directional ring of jumps \( +k \) and \(-k', j < i\) for which the reminder is not zero.

Algorithm 3.8. Send a message from the current node \( C \) to the destination node 1) which is originated at source node \( S \) in a RCG-II, \( R_{q,S} \) with \( q \leq p' - 1 \) and \( S = \{-k^{n-1},-k^{n-2},...,l\}, l+1l...+k^{n-2},+k^{n-1}\}.

if \( (C = -D) \) then
  Send message to local node \( C \)
else
  // Compute the distance and jump to move on.
  Distance = (D-C+p) \%p
  jump = 1
  while (Distance \% (jump*\(k\)) \~ 0)
    jump = jump * \(k\)
  endwhile.
  if((Distance \% (jump*\(k\)) (jump*\(k\)/2)) \&\& (Distance\% (jump*\(k\))/0))
    jump = (-1) * jump
    if((p-Distance) magnitude of length of jump)
      Send negative acknowledgement to the source \( S \) indicating that the destination node \( D \) is failed
    else
      Move along jump.
    endif.
  else
    if(Distance magnitude of length of jump)
      Send negative acknowledgement to the source \( S \) indicating that the destination node \( D \) is failed
    else
      Move along jump.
    endif.
  endif.
endif.

Routing is explained by considering the RCG-II with 25 active nodes and one failed node with id 4, shown in Figure 3.12. For example, assume a packet is originated at node 3 to node 7. Since the distance is 4, the packet is routed along the Hamiltonian circuit of jump +1, to node 5. From there it continues on the Hamiltonian circuit of jump +1 to node 6 and then finally to the destination node 7. It may be observed that there exists a shorter route 3 to 6 to 7.
The routing algorithm doesn't guarantee to route a packet through shortest path. However, packets are guaranteed to reach the destination node, provided it is active. This is because the packet continues to move along the Hamiltonian circuit of jump \( k' \) as long as the remainder of the distance on the Hamiltonian circuit of jump +1 divided by \( k^{i+1} \) is not zero.

**Diameter of RCG-II.**

The maximum diameter of RCG-II is less than \( (n-1)(k-1) \). Theorem 3.8 gives the proof.

**Theorem 3.8.** The maximum diameter of RCG-II, \( R_{q,s} \) with \( q \leq k^n - 1 \) and connection set \( S = \{-k^{n-1}, -k^{n-2}, \ldots, -1, 1, \ldots, +k^{n-2}, +k^{n-1}\} \) is less than \( (n-1)(k-1) \).

**Proof:** Let \( d = (Y = X + k^n - 1) \) be the distance between two arbitrary nodes \( X \) and \( Y \) on the Hamiltonian circuit of jump +1. This distance can be expressed in \( k \)-ary number system as

\[
d = d_n d_{n-1} \ldots d_0
d\]

where \( d_i < k \). We can, without loss of generality, presume \( d_i < k/2 \) because if \( d_i > k/2 \), the distance \( d_i k' \) can be reached using one hop along jump \( k^{i+1} \) and \( (k-d_i) \) hops using jump \( -k' \).

Ideally this distance can be reached, according to theorem 3.7, in a maximum of \( n^q k/2 \) hops. However, as the nodes between \( X \) and \( Y \) keep failing, some of the nodes along jumps of magnitude \( k_i \), \( i > 0 \), may not be accessible. This leads to movement in the bi-directional ring of jumps of magnitude \( k_i \), \( j < i \).

Let us assume that \( Y \) is reached from \( X \), by making hops according to Algorithm 3.8, in \( m \) hops. Then \( Y \) can be expressed as sum of hops from \( X \) as

\[
Y = (X + h_1 + h_2 + \ldots + h_m) \% (k^n - 1)
\]

where \( h_j \) is an integer multiple of \( k^j \), \( 0 \leq j \leq (k-1) \).
Without loss of generality, we assume $X < Y$. Then,

$$Y - (X + h_1 + h_2 + \ldots + h_m) \quad (3.10)$$

Let $h_j$ for $j = 1, \ldots, 1$ are the hops along jumps $k^a$ and $k^b$ in the sequence of $m$ hops. Then $a < b$. This is because, according to Algorithm 3.8, a packet is routed along jump $k^i$, where $i$ is the lowest dimension in the binary representation of distance such that $d_i$.

Out of these $m$ hops, let the hops along jump $k^i$, $r$, is greater than or equal to $k$. Then, there exists $c$, $d$ and $e$ such that,

$$\sum_{j=r}^d h_j = e k^{r+1} \quad (3.11)$$

According to Hypothesis 3.4, there exists an image of a node along jumps of magnitude $k^1$ in every $k$ hops. Thus the hops along jump $k^i$, $r$, can be reduced to $(r-k)$ hops. By applying Hypothesis 3.4 repeatedly, to all jumps $k^i$ for which the hops are greater than $k$, it reduces the number of hops to less than $k$.

However, the maximum hops along the bi-directional ring of jumps $k^{n-1}$ and $-k^{n-1}$ is less than or equal to $k/2$, because Algorithm 3.8 makes use of the bi-directional ring of jumps $k^{n-1}$ and $-k^{n-1}$.

Therefore, an arbitrary node $Y$ can be reached from an arbitrary node $X$ in a maximum of $L = (n-1)(k-1)$ hops, that is, the sum of maximum of hops along bi-directional ring of jumps $k^{n-1}$ and $-k^{n-1}$ and the maximum hops along the Hamiltonian circuits of jumps $+k$, $V/0 \leq i < n-1$.

Thus the maximum diameter of RCG-II, $R_{q,s}$ with $q < k^n - 1$ and $S = \{-k^{n-1}, -k^{n-2}, \ldots, -1, +1, \ldots, +k^{n-2}, +k^{n-1}\}$ which is the maximum of hop-distance between all pair of nodes is less than $L = (n-1)(k-1)$.

Though it is proved that the maximum diameter of RCG-II is less than $L = (n-1)(k-1)$, it is observed empirically that the maximum diameter is $kn/2$ (refer Chapter 4). It is
probably because \( r > kl2 \) hops along bi-directional ring \(+k\) and \(-k\) can be replaced with one jump along the bi-directional ring \( +k^{l+1} \) and \(-k^{l+1} \) and \((k-r)\) jumps along the bi-directional ring \(+k\) and \(-k\).

The value \( k \) is selected such that it is \( n^{th}\)-root of the number of nodes in the base Circulant Graph-II, \( p=k^n-1 \).

This way of selecting \( k \), \( n \) and \( p \) is motivated by the fact that the links of a node in a n-D Torus are identified by jumps of length \(+1\), \(-1\), \(+k\), \(-k\), \( +k^{l+1} \) and \(-k^{l+1} \), assuming that the nodes are represented as integers instead of their position in the \( n\)-dimensional Euclidean space.

The successful design of RCG-II provided an insight to construct a base Circulant graph whose performance is equivalent to that of Binary Hypercube. Following section presents the design of a dynamically reconfigurable topology derived from this base topology.

3.3.3 Reconfigurable Circulant Graph - III

This section presents another reconfigurable topology derived from the base Circulant graph \( C_{p,s} \) with \( p=2^n-1 \) and \( S=\{1,2,\ldots,2^{n-2},2^{n-1}\} \). We refer this base Circulant graph as Circulant Graph-III or CG-III. Let us first see the properties of the base CG-I topology in order to prove that the resultant reconfigurable topology retains the structural properties of the base topology. CG-III is different from the other two Circulant graphs, CG-I and CG-II, in terms of connection set, \( S \) and the number of nodes, \( p \).

An example of the base Circulant Graph-III, \( C_{p,s} \) with \( p=7 \) and connection set \( S=\{1,2,4\} \) is shown in Figure 3.14.

It may be observed from the Figure 3.14 that there exists three edge-disjoint Hamiltonian circuits, each one being associated with a jump in the connection set \( S \). Corollary 3.3 proves the existence of such Hamiltonian circuits in the base Circulant graph.
Corollary 3.3. There are \( n \) edge-disjoint Hamiltonian circuits in \( C_{p,S} \) with \( p = 2^n - 1 \) and \( S = \{1,2,\ldots,2^n-2,2^n-1\} \).

Proof: According to Hypothesis 3.2, since all the jumps in the connection set \( S \) are distinct, it is sufficient to show that the paths associated with jumps are Hamiltonian circuits. Further, according to Theorem 3.4, it is sufficient to prove that all these jumps are relatively prime with respect to \( p \).

1 is relatively prime with respect to \( p \). Suppose \( 2^i, 0 \leq i < n-1, \) is not relatively prime with respect to \( p \). Then \( p/2^i \) should be an integer. But \( p/2^i = (2^n - 1)/2^i \) is not an integer. This contradicts our assumption. Therefore, \( 2^i \) has to be relatively prime with respect to \( p \).

Thus all the jumps of connection set \( S \) are relatively prime with respect to \( p \). Hence, the Hamiltonian circuits associated with \( n \) different jumps are edge disjoint.

**Routing in base Circulant graph-III.**

A distributed shortest path routing method in the CG-III topology is described in Algorithm 3.9. According to this algorithm, packets are routed by making use of the Hamiltonian circuits present in the Circulant graph. To reach the destination node, a packet first moves along the Hamiltonian circuit of jump 1, then along the Hamiltonian circuit of jump 2, then along the Hamiltonian circuit of jump \( 2^2 \), and so on.
Algorithm 3.9. Send a message from the current node $C$ to the destination node $D$ which is originated at source node $S$ in a Circulant Graph-II, $C_{p,S}$ with $p = 2^n - 1$ and connection set $S = \{1, 2, ..., 2^{n-2}, 2^{n-1}\}$.

if $(C = D)$ then
    Send message to local node $C$
else
    // Compute the distance and jump to move on.
    Distance $(D-C) \mod p$
    jump 1
    while $(\text{Distance} \mod (\text{jump} \times 2) = 0)$
        jump = jump $\times 2$
    endwhile.
endif.

Diameter of base Circulant Graph-III.

The diameter of Circulant Graph-III is found to be $n$. The following theorem proves this.

Theorem 3.9. The diameter of Circulant graph $C_{p,S}$ with $p = 2^n - 1$ and connection set $S = \{1, 2, ..., 2^{n-2}, 2^{n-1}\}$ is $n$.

Proof: The maximum hop-distance between any two arbitrary nodes along the Hamiltonian circuit associated with jump 1 is $(2^n - 1)/2$. This hop-distance is further reduced by making use of jumps of higher magnitude, i.e., $2, 2^2, ..., 2^{n-1}$.

Let $h$ be the hop-distance between two arbitrary nodes $X$ and $Y$ along the Hamiltonian circuit associated with jump 1. Let $h$ be expressed in binary number system as

$$ h = h_{n-1}h_{n-2}...h_0 $$

where $h_i = 0$ or 1, $i < n$. Each bit $h_i \neq 0$, $i < n$ in $h$ can be reduced to zero by moving along the jump $2^i$. Thus, all the bits of $h$ can be reduced to zero, i.e., the distance $h$ can be reached in a maximum of $h$ hops.

Hence, the diameter of $C_{p,S}$ with $p = 2^n - 1$ and $S = \{1, 2, ..., 2^{n-2}, 2^{n-1}\}$, the maximum hop-distance between any pair of nodes is $n$.

It may be observed that the Circulant graph $C_{p,S}$ with $p = 2^n - 1$ and $S = \{1, 2, ..., 2^{n-2}, 2^{n-1}\}$ can be considered equivalent to binary Hypercube. This is because the Circulant graph $C_{p,S}$
also has a node degree of \( n \) and a diameter of \( \frac{p}{2} \), where \( p = 2^n - 1 \) is the number of nodes in the network.

Having seen the properties of base Circulant Graph-III, we will now define the **Reconfigurable Circulant Graph-III (RCG-III)**. This reconfigurable topology is designed to preserve the properties of the base Circulant Graph-III, \( C_{p,S} \) with \( p = 2^n - 1 \) and \( S = \{1, 2, \ldots, 2^{n-2}, 2^{n-1}\} \).

**Reconfigurable Circulant Graph-III.** Let the Circulant Graph-III, \( C_{p,S} \) with \( p = 2^n - 1 \) and \( S = \{1, 2, \ldots, 2^{n-2}, 2^{n-1}\} \) be the base topology and \( q \) be any positive integer such that \( q < p \). The \( q \)-node **Reconfigurable Circulant Graph-III (RCG-III)** with connection set \( S = \{1, 2, \ldots, 2^{n-2}, 2^{n-1}\} \), denoted by \( R_{q,S} \), consists of \( q \) nodes. Each node in \( R_{q,S} \) has unique label in the range 0 through \( p-1 \). Each node \( i \) is connected to \((i + js) \mod p\) where \( s \in S \) and \( j \) be a positive integer such that for any positive integer \( k < j \), nodes with labels \((i + k*s) \mod p\) are failed.

Figure 3.15 shows an example of a 6-node RCG-III derived from a base CG-III topology, \( C_{p,S} \) with \( p = 7 \) and \( S = \{1, 2, 4\} \) and with a failed node 4.

![Figure 3.15 RCG-III with base topology \( C_{p,S} \) with \( p = 7 \) and \( S = \{1, 2, 4\} \) and failed node 4.](image)

**Reconfiguration in RCG.**

The reconfiguration of RCG-III is similar to that of the RCGs described earlier using the Algorithm 3.5. Whenever a node fails, the neighboring nodes connect to the next node on the corresponding **Hamiltonian** circuit of jump \( s \) through which the failed node is
connected. Similarly, when a node recovers from failure, it occupies its position in the base circulant graph.

To explain the process of reconfiguration in case of node failure, consider a base topology shown in Figure 3.14. Assume that node 4 is failed. By connecting the neighbors of failed node to next nodes on the corresponding Hamiltonian circuits, we get the resultant topology as shown in Figure 3.15. Continuing the reconfiguration with failure of node 3 results in the structure shown in Figure 3.16.

![Figure 3.16 RCG-III derived from base topology $G_p S$ with $p 7$ and $S=\{1,2,4\}$ and failed nodes 3,4.](image)

To explain the process of node addition, consider RCG-III with six active nodes and two failed nodes 3 and 4 as shown in Figure 3.16. If node 3 wishes to enter the network, it informs the network and occupies its position in the base circulant graph by forming links with its neighbors, i.e., with nodes 5 and 7. This results in the structure shown in Figure 3.15.

**Routing in RCG-III.**

A distributed routing method in RCG-III $R_{q,s}$ with $q \leq p=2^n-1$ and $S=\{1,2,\ldots,2^{n-2},2^{n-1}\}$ is described in the Algorithm 3.10. According to this algorithm, a packet is routed by making use of the Hamiltonian circuits present in the base Circulant Graph-III, each time reducing the distance of the destination along the Hamiltonian circuit of jump 1. To reach the destination node, a packet moves, starting from jump 1, along the jump $2^1$ where the $i$-bit in binary representation of distance is 1.
Algorithm 3.10. Send a message from the current node C to the destination node D which is originated at source node S in a RCG-III, \( R_{q,s} \) with \( q \leq 2^n-1 \) and \( S = \{1,2,\ldots,2^n-2^n+1\} \).

```plaintext
if (C=D) then
  Send message to local node C
else
  // Compute the distance and jump to move on.
  Distance = (D-C+p) %p
  jump = 1
  while(Distance % (jump*2) = 0)
    jump = jump * 2
  endwhile.
  if (Distance Length of jump)
    Send negative acknowledgement to the source S indicating that the destination node D is failed
  else
    Move along jump.
  endif.
endif.
```

For better understanding of routing in RCG-III, consider the topology shown in Figure 3.15. The topology has 6 active nodes and one failed node with id 4. Assume that node 3 wants to send a packet to node 1. The distance \( 5 = (1-3+7)%7 \) is reached by making hop along jump 1 to node 5. From node 5, where the distance is calculated as 3, the packet is moved along jump 1 to node 6. At node 6, the distance is calculated as 2 and the packet is routed along jump 2 to the destination node 1. Though the packet is reached the destination, this is not the shortest path from node 3 to node 1. However, this routing algorithm guarantees that the destination node can be reached, provided it is active. This is because the algorithm reduces the distance along the Hamiltonian circuit of jump 1, with each hop.

**Diameter of RCG-III.**

The maximum diameter of RCG-III is found to be \( n \). Theorem 3.10 gives the proof.

**Theorem 3.10.** The maximum diameter of RCG-III, \( R_{q,s} \) with \( q \leq 2^n-1 \) and connection set \( S = \{1,2,\ldots,2^n-2^n+1\} \) is \( n \).

**Proof:** According to theorem 3.9, the maximum hop-distance between any two arbitrary nodes X and Y, \( h(X,Y) \) is \( n \).
As the nodes between X and Y keep failing, some of the nodes along the Hamiltonian circuit of jump $2^i$, $i > 0$, may not be accessible. This leads to movement along the Hamiltonian circuit of jump $2^{i+1}$. Let us assume that Y is reached from X in $k$ hops, each hop reducing the distance on the Hamiltonian circuit using Algorithm 3.10. Then Y can be expressed as sum of hops from X as

$$Y = (X + h_1 + h_2 + \ldots + h_j) \mod (2^n - 1) \quad (3.12)$$

Without loss of generality, we assume $X < Y$. Then,

$$Y = X + h_1 + h_2 + \ldots + h_k \quad (3.13)$$

where $h_i$ is an integer multiple of $1, 2, \ldots, 2^{n-2}$, or $2^{n-1}$. Let $h$, and $h,+$ be two adjacent hops in the above sequence along the jumps $2^a$ and $2^b$. Then, according to Algorithm 3.10, $a < b$.

If $k$ is greater than $n$, then there exist two or more hops along jump $2^i$, for some $0 < i < n$. Let $h_c$ and $h_d$ be the starting and ending hop along the jump $2^i$, in the above sequence of $k$ hops. Then, hops between $h_c$ and $h_d$ can be represented as

$$h_j = e_j 2^i \quad (3.14)$$

where $e_j$ is called as the coefficient of hop $h_i$.

The hops from $h_c$ to $h_d$ can be replaced with/hops of jump $2^{(i+1)\mod n}$ such that

$$f < (d - c)$$

This is because the distance $2^{(i+1)\mod n}$ can be reached in at most two hops of jump $2^i$ and there exists at least $(d-c)$ hops whose sum of the coefficients is an even number.

Thus, by replacing the hops from $h_c$ to $h_d$ with $f$ hops of, there may exist at most one hop of jump $2^i$.

By repeating this procedure for all jumps, starting with jump 1, we can find a path with a maximum of $//h$ hops.
Hence, the number of hops required to reach an arbitrary node $Y$ from an arbitrary node $X$, $k < n$. In other words, the maximum diameter of $RCG-11I$, $R_{q,s}$ with $q\leq2^n-1$, the maximum of hop-distance between all pair of nodes is $n$.

So far we have studied the design of dynamically reconfigurable topologies whose base topologies are constructed with edge-disjoint Hamiltonian circuits. It is observed that these topologies have comparable performance with 2-D Torus, n-D Torus and Binary Hypercube.

Though the base Circulant Graphs are constructed using a set of edge-disjoint Hamiltonian circuits, it is enough if the base topology has one Hamiltonian circuit for maintaining connectivity.

### 3.4 Discussion

The reconfigurable logical topologies proposed in this chapter ~ Perturbed Torus, Reconfigurable Circulant Graphs that are equivalent to 2-D Torus, n-D Torus and Binary Hypercube — consider reconfiguration as part of the topology design issue. These topologies reconfigure using Local Perturbation paradigm proposed as part of this thesis. For accommodating changes (failure and addition of nodes) in network, these topologies assume an initial well defined regular structure called base topology. Addition or deletion of node is considered as assigning or un-assigning a position in the base topology. Hence, a reconfigurable topology moves closer or away from the base topology depending upon whether a node is added or deleted from the network.

Applicability of the reconfiguration method based on Local Perturbation to the traditional topologies such as Torus and Hypercube is investigated in Section 3.2. From the results obtained, it is shown that successive application of reconfiguration based on Local Perturbations for node failures disconnects the topology. To retain connectivity, additional steps must be devised into the reconfiguration process. Perturbed Torus is designed by maintaining a base row (column) in the base topology. However, it is not always possible to define such special methods to retain connectivity.
By adopting the concept of Hamiltonian circuit, wherein every node in the network is visited exactly once, a set of reconfigurable topologies is proposed. The idea is to define reconfiguration along Hamiltonian circuits so that the network is always connected. In Section 3.3, we designed a set of topologies, called Reconfigurable Circulant Graphs, derived from a subset of Circulant graphs. Incidentally, all these topologies have edge disjoint Hamiltonian circuits. However, it is enough if the base topology has one Hamiltonian circuit for maintaining connectivity.

As per the framework proposed earlier, for evaluation of topologies, here, we will evaluate the reconfigurable topologies proposed in this thesis. Table 3.1 summarizes the node degree and diameter properties of the dynamically reconfigurable topologies. It is observed that Perturbed Torus and Reconfigurable Circulant Graph - 1 (RCG-I) have same node degree and number of nodes as that of 2-D Torus and Circulant Graph - 1. Similarly, RCG-T1 has the same node degree and number of nodes as that of Circulant Graph - II and Multidimensional Torus. RCG-III has the same node degree and number of nodes as that of Circulant Graph - III and Binary Hypercube.

As the reconfigurable topologies change their structure with every reconfiguration, diameter is measured as the maximum of diameter for all different structures that can be derived from the base topology. Since the reconfiguration is done to retain routing properties, usually the maximum diameter of a reconfigurable topology is limited to the diameter of the base topology. We proved the maximum diameter of Perturbed Torus is almost same as 2-D Torus. Though the maximum diameter obtained for RCG-I and RCG-II is more than that of corresponding base topologies, the empirical results obtained in the next chapter supports our argument.

It may not be possible to formulate the computation of average internode distance for these topologies as the structure of reconfigurable topologies change with every reconfiguration. In chapter 4, we evaluate average internode distance empirically by assuming different structures and compare them with corresponding base structures and their counterparts in traditional topologies.
Designing shortest path routing algorithms is also not possible for reconfigurable topologies because the nodes in these networks don't have the idea about the structure of the topology at any given point of time. In other words, because the nodes wouldn't have the knowledge of which positions in the base topological structure are vacant, it is not possible to design a shortest path routing algorithm. It is also observed that a little overhead will be introduced into the routing algorithms of reconfigurable topologies when compared to the routing in corresponding base topologies.

<table>
<thead>
<tr>
<th>Topology</th>
<th>No. of Nodes</th>
<th>Node Degree</th>
<th>Diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Hypercube</td>
<td>$N=2^n$</td>
<td>$n$</td>
<td>$n$</td>
</tr>
<tr>
<td>Multidimensional Torus</td>
<td>$N=k^n$</td>
<td>$2^n$</td>
<td>$k^n/2$</td>
</tr>
<tr>
<td>Perturbed Torus</td>
<td>$N=n^2$</td>
<td>$4$</td>
<td>$2^\lceil \sqrt{N}/2 \rceil + 1$</td>
</tr>
<tr>
<td>2-D Torus</td>
<td>$N=n^2$</td>
<td>$4$</td>
<td>$n$</td>
</tr>
<tr>
<td>RCG-I</td>
<td>$N=n^2-1$</td>
<td>$4$</td>
<td>$3n/2$</td>
</tr>
<tr>
<td>Circulant Graph - I</td>
<td>$N=n^2-1$</td>
<td>$4$</td>
<td>$n$</td>
</tr>
<tr>
<td>RCG-II</td>
<td>$N=k^n-1$</td>
<td>$2^n$</td>
<td>$k/2 + (n-1)(k-1)$</td>
</tr>
<tr>
<td>Circulant Graph - II</td>
<td>$N=k^n-1$</td>
<td>$2^n$</td>
<td>$k^n/2$</td>
</tr>
<tr>
<td>RCG-III</td>
<td>$N=2^n-1$</td>
<td>$n$</td>
<td>$n$</td>
</tr>
<tr>
<td>Circulant Graph - III</td>
<td>$N=2^n-1$</td>
<td>$n$</td>
<td>$n$</td>
</tr>
</tbody>
</table>

*Table 3.1* Comparison of Reconfigurable topologies with their counterparts.
The main advantage of the proposed reconfigurable topologies when compared with traditional topologies such as deBruijn graph, MS Net and Binary Hypercube is the dynamic reconfiguration. *Dynamic reconfiguration is achieved with minimum disturbance to the network.*