An Efficient Method based on Self-Generating Disjoint Minimal Cut-Sets for Evaluating Reliability Measures of Interconnection Networks

P. K. TRIPATHY*1, R. K. DASH2, C. R. TRIPATHY3

1Department of CSE, Silicon Institute of Technology, Bhubaneswar, Odisha, INDIA.
2Department of CSA, College of Engineering & Technology, Bhubaneswar, Odisha INDIA.
3Department of CSE, VSS University of Technology, Burla, Odisha, INDIA.

(Received on October 29, 2013, revised on March 14 and April 02, 2014)

Abstract: The reliability evaluation of interconnection networks is an important issue in their quality management. Some important reliability measures are network reliability or g-terminal reliability, terminal reliability or 2-terminal reliability and k-terminal reliability. This paper concentrates on evaluating these three different reliability measures of interconnection networks. In this context a new method based on self generating disjoint minimal cut-set is proposed for evaluating the reliability of the interconnection networks. As, the minimal cut-sets thus generated are self-disjoint and non-redundant; it saves the overhead due to disjointing process and eliminates the duplicates. The method is well supported by an efficient algorithm for calculating the different reliability measures of interconnection networks. The proposed method as well as the algorithm is illustrated with a suitable example. The proposed method is compared against some existing methods in order to ensure its simplicity and efficiency in computing the reliability of interconnection networks. Further to show the generality of the proposed method the reliability of some regular interconnection networks are evaluated by using the proposed method.

Keywords: Reliability, Interconnection networks, Minimal cut-set/ path-set, probabilistic graph

1. Introduction

Interconnection network plays an important role in many real world application systems such as, distributed access networks, computer networks, communication networks etc., since it is the only mean for inter-processor communication. So there is always a need to design a highly reliable system which is quite economical and operational. Hence in designing of such systems the reliability is an important performance criteria [1]. The different reliability measures are g-terminal, k-terminal and two-terminal reliability [1-2].

Review of the literature records some existing techniques based on spanning trees with a disjoint grouping approach for the evaluation of g-terminal reliability of networks e.g., [3]. However, most of these methods, though very simple, become impractical even for moderate size networks due to the exponential growth in the number of spanning trees with an increase in the network size. Recently, Kansal and Devi [4] suggested a minimal cut-set based algorithm; however, their algorithm generates many redundant cut-sets, and so it does not provide correct results [5].

The methods for evaluating two-terminal/k-terminal reliabilities use one of the following techniques: State enumeration, Boolean algebra and Sum-of-Disjoint Products (SDP), Factoring Theorem and Binary Decision Diagram (BDD). The methods in [6-10] are based on state enumeration and sum-of-disjoint product and therefore require minimal
paths/cuts to be enumerated in advance. Then, the minimal paths/cuts are manipulated to get the counterparts in sum-of-disjoint product form. However such methods become quite intractable for large size networks, as it requires much computational efforts for disjointing of the minimal path/cut sets. However, methods [11-15] supplemented some improvement on this method by generating minimal path and minimal cut sets for general networks.

The demerits of different methods discussed so far motivate us to propose a new method which finds self-generating disjoint minimal cut-sets without duplication and the reliability is evaluated by considering corresponding links of the generated minimal cut sets. Thus, the proposed method eliminates the computational task needed for the disjointing process. Rest of the paper is organized as follow: A new method based on self-generating disjoint cut-sets is proposed in Section 2, Section 3 presents the algorithm and Section 4 illustrates the method with suitable examples. Comparison of proposed method with existing methods is been presented in Section 5. Section 6 concludes the paper.

2. Proposed Method For Evaluating The Reliability of Interconnection Networks

The following notation and assumptions are used in this paper.

**Notation:**

\[ G(V,E) \]  
Probabilistic graph with \( V \) number of nodes and \( E \) number of edges

\( s, t \)  
a pair of Source node and sink node

\( U \)  
The set containing the \( k \) no. of nodes for which \( k \)-terminal reliability is to be calculated

\( X, W \)  
Vertex set

\( L \)  
Link Set

\( |X| \)  
Number of elements in \( X \)

\( \bigcup_k \)  
Union operation up to \( k \) times

\( L \)  
Link Set

\( |X| \)  
Number of elements in \( X \)

\( \bigcup_k \)  
Union operation up to \( k \) times

\( \oplus \)  
Boolean XOR operation

\( \text{link}(W) \)  
Links associated with node set \( W \)

\( N \)  
Number of nodes in the network

\( R \)  
Reliability

\( n \)  
Dimension of a network

**Assumptions:**

i) The nodes of the network are perfectly reliable. (ii) The network and its links have only two state; working or failed (iii) The link failures are statistically independent

The following theorem and lemma are proposed to ensure the correctness of the proposed method.

**Theorem 1** (For \( g \)-terminal Reliability): The node sets \( W_i \), for \( i=1,2,... \) are called as minimal cut sets if the following conditions hold.

(i) \( \{s\} \in W_i \)  
(ii) \( \exists \) at least one node \( v \) in a path of length \( K \) from source node \( \{s\} \)

(iii) \( V \sim W_i \) must be connected  

**Condition-(1)**
**Proof:** The node sets \( W_i \), for \( i = 1, 2, 3 \ldots \) are said to be minimal cut-sets if they satisfy the following two properties.

a. Deleting all arcs associated with \( W_i \) reduces the probabilistic graph \( G \) into two connected components.

b. There exists no proper subset of \( W_i \) satisfying the condition (i).

So, in evaluating the \( g \)-terminal reliability of the Inter Connection Network the node set containing source node \( s \) and rest of the nodes must belong to different connected components of the reduced graph which implicitly proves the condition- (iii) of the proposed theorem. As \( s \) is the source node, there must exist a path starting with \( s \) which explicitly requires \( s \) being a member of cut-set which proves the condition- (i) of the theorem. As \( W_i \) is a minimal cut-set, the connected components containing the source node \( s \) of graph \( G \) may be considered as sub-graph of \( G_i \) which is connected, which follows the condition- (ii) of the proposed theorem.

From the above theorem, the following two lemmas are proposed:

**Lemma 1** (For \( k \)-terminal Reliability): The node sets \( W_i \), for \( i = 1, 2 \ldots \) are called as minimal cut sets if the following conditions hold.

(i) \( \exists \) at least one node \( u \in U \) that must be a member of \( W_i \) (ii) \( \exists \) at least one node \( v \) in a path of length \( K \) from node \( \{ u \} \) (iii) \( V \sim W_k \) must be connected and should contain at least one element from the node set \( U \).

**Condition-(2)**

**Lemma 2** (For 2-terminal Reliability): The node sets \( W_i \), for \( i = 1, 2 \ldots \) are called as minimal cut sets if the following conditions hold.

(i) \( \{ t \} \in W_k \) (ii) \( \exists \) at least one node \( v \) in a path of length \( K \) from sink node \( \{ t \} \).

(iii) \( V \sim W_k \) must be connected.

**Condition-(3)**

The following lemma ensures that the minimal cut-sets generated by using Theorem-1 are non-redundant in nature.

**Lemma 3:** The minimal cut-sets satisfying the Theorem-1 are non-redundant in nature.

**Proof:** In addition to the minimal cut-sets \( W_i \), let it be assumed that \( W_i' \) be also a series of minimal cut-sets of the probabilistic graph \( G \). Since \( W_i' \) are minimal cut-sets, \( W_i' \) must satisfy the condition (i)-(iii) of the proposed Theorem-1. So, each member of \( W_i' \) be same as the members of \( W_i \) or their proper subsets, this implies \( W_i \) not to be the minimal cut-sets, which contradicts our assumption that \( W_i \) is the minimal cut-sets. So, \( W_i = W_i' \), which proves the lemma.

3. **Proposed Algorithm for Enumerating of self-generating Disjoint Minimal Cut-sets of Interconnection Networks**

The following algorithm is proposed to evaluate the reliability of the Interconnection network:

**Reliability_Evaluation (IN, U, s, t)**

\[
\{ 
\text{Generate } X \text{ such that each element, } x \in X \text{ is a node } v \in V \sim s \}
\]
Initialize $W_0 = \{ s \}$, $L = \emptyset$

For $k = 1$ to $|X| - 1$ /*Finding self generating disjoint Minimal Cut-Set*/

Find $W_k = \bigcup_i X_i, i = 1, 2, 3 \cdots$ using Condition-1 to 3)

for $k = 0$ to $|X| - 1$

$\bigcup_i \bigoplus (\text{link}(W_k)) \bigcup \bar{I}$

$R = (L)_{\text{each } l \in L \rightarrow p \text{ and each } \bar{I} \in L \rightarrow q, \cup \rightarrow + \text{ and } \cap \rightarrow \times}$

return (R)

**Theorem 2:** The running time of the proposed algorithm is found to be linear in nature.

**Proof:** The first for loop of the proposed algorithm in step 5 will be iterated to a maximum number of the nodes ($V$) in the graph. The second for loop in step 7 of the proposed algorithm also takes a worst case running time of $O(V)$. So, the overall running time of the proposed algorithm is found to be $\max\{O(V), O(V)\} = O(V)$, which is linear in nature.

4. Illustration

The proposed method is illustrated by taking the following example:

![Figure 1: Network 1 (Network with 6 Nodes and 9 Links)](image)

$|X| = \{ \{1\}, \{2\}, \{3\}, \{4\}, \{5\} \}$ i.e., 5. Rest of the results are shown in Table 1.

So the link set $L = \{ (a,b), (b,c,d,\overline{a}), (a,c,e,f,\overline{b}), (d,e,f,\overline{a},\overline{b},\overline{c}), (b,c,e,g,h,\overline{a},\overline{d}), (a,c,e,g,i,\overline{b},\overline{f}), (f,g,h,\overline{a},\overline{b},\overline{c},\overline{d},\overline{e}), (b,c,e,g,i,\overline{a},\overline{d},\overline{h}), (d,e,g,i,\overline{a},\overline{b},\overline{c},\overline{f}), (a,c,e,g,h), (b,\overline{f},\overline{i}), (b,c,e,f,\overline{a},\overline{d},\overline{g},\overline{i},\overline{h}), (b,\overline{h},\overline{a},\overline{b},\overline{c},\overline{d},\overline{e},\overline{f},\overline{g}), (d,e,g,h,\overline{a},\overline{b},\overline{c},\overline{f},\overline{i}), (a,c,d), (b,\overline{f},\overline{e},\overline{g},\overline{i},\overline{h}) \}.$

The g-terminal reliability expression for the example network is found to be:

$p \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q \cdot p \cdot q . \cdots \ (1)$

Considering the reliability of links to be $0.9$, i.e $p=0.9$, the g-terminal reliability of the example network (Fig. 1) by using the proposed method is computed to be $0.968317341$. 
For g-Terminal Reliability:

Table 1: Stepwise Illustration of the Algorithm for g-Terminal Reliability:

<table>
<thead>
<tr>
<th>k</th>
<th>$W_k$</th>
<th>Node Set</th>
<th>Links Associated with the Node Set</th>
<th>$\overline{I}$</th>
<th>$L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>{s}</td>
<td>{s}</td>
<td>{$\overline{{a,b}}$}</td>
<td>-</td>
<td>{$\overline{{a,b}}$}</td>
</tr>
<tr>
<td>1</td>
<td>{s,1}</td>
<td>{s,1}</td>
<td>{$\overline{{a,b,{a,c,d}}}$}</td>
<td>{$\overline{{a}}$}</td>
<td>$L \cup {\overline{{{b,c,d,\overline{{a}}}}}}$</td>
</tr>
<tr>
<td>2</td>
<td>{s,2}</td>
<td>{s,2}</td>
<td>{$\overline{{a,b,{b,c,e,f}}}$}</td>
<td>{$\overline{{b}}$}</td>
<td>$L \cup {\overline{{{a,c,e,f,\overline{{b}}}}}}$</td>
</tr>
<tr>
<td>3</td>
<td>{s,1,2}</td>
<td>{s,1,2}</td>
<td>{$\overline{{a,b,{a,c,d,\overline{{e,f}}}}}$}</td>
<td>{$\overline{{a,b,c}}$}</td>
<td>$L \cup {\overline{{{d,e,f,\overline{{a,b,c}}}}}}$</td>
</tr>
<tr>
<td>4</td>
<td>{s,1,3,4}</td>
<td>{s,1,3,4}</td>
<td>{$\overline{{a,b,{a,c,d,\overline{{e,f}}}}}$}</td>
<td>{$\overline{{a,d}}$}</td>
<td>$L \cup {\overline{{{b,c,e,g,h,\overline{{a,d}}}}}}$</td>
</tr>
</tbody>
</table>

For k-Terminal Reliability:

Considering the $U = \{s,1,3,4\}$, the output of the algorithm are shown in Table-2.

The k-terminal reliability expression for the example network found to be:

Considering the reliability of links to be 0.9, i.e., $p = 0.9$, the k-terminal reliability of the example network (Fig. 1) by using the proposed method is computed to be 0.9629071020.
For 2-Terminal Reliability:

Considering the source node as 1 and destination node as 6, the output of the algorithm are presented in Table-3.

Therefore \( L = \{[h,i], [d,e,i,g,\overline{h}, i], [f,g,h,\overline{i}], [a,\overline{b}, c,\overline{d}, e, g, i], [b, c, e, f, g, h, i], [d, e, f, g, h, i], [a, b, c, \overline{d}, e, f, g, h, i] \} \).

The two terminal reliability expression for the source node \( s \) to the sink node \( t \) turns out to be:

\[
P^s_t = p_1^s p_2^s q^s p_3^s q^s p_4^s q^s p_5^s q^s p_6^s q^s p_7^s
\]

(3)

Considering the reliability of links to be 0.9, i.e. \( p = 0.9 \), the two-terminal reliability of the example network (Fig. 1) by using the proposed method is found to be 0.961121781.

Table 2: Stepwise Illustration of the Algorithm for k-Terminal Reliability:

<table>
<thead>
<tr>
<th>k</th>
<th>( W_k )</th>
<th>Node Set</th>
<th>Links associated with the node set</th>
<th>( I )</th>
<th>( L )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>{s}</td>
<td>{s}</td>
<td>{[a, b]}</td>
<td>{[a, b]}</td>
<td>{}</td>
</tr>
<tr>
<td>1</td>
<td>{{s,1}, {s,2}}</td>
<td>{s,1}</td>
<td>{[a, b], [a, c, d]}</td>
<td>{a}</td>
<td>{{a, b, d}}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{s,2}</td>
<td>{[a, b], [b, c, e, f]}</td>
<td>{b}</td>
<td>{{a, c, e, f}}</td>
</tr>
<tr>
<td>2</td>
<td>{{s,1,2}, {s,1,3}, {s, 2,4}}</td>
<td>{s,1,2}</td>
<td>{[a, b], [a, c, d, f], [b, c, e, f]}</td>
<td>{a, b, c}</td>
<td>{{d, e, g, h}}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{s,1,3}</td>
<td>{[a, b], [a, c, d, f], [b, c, e, f]}</td>
<td>{a, d}</td>
<td>{{d, e, g, h}}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{s,2,4}</td>
<td>{[a, b], [b, c, e, f], [f, g, i]}</td>
<td>{b, f}</td>
<td>{{a, c, e, g, i}}</td>
</tr>
<tr>
<td>3</td>
<td>{{s,1,2,3}, {s,1,3, t} }, {s,1,4}, {s,2,4}, {s,3,4}, {s,2,3, 4}}</td>
<td>{s,1,2,3}</td>
<td>{[a, b], [a, c, d, f], [d, e, g, h], [b, c, e, f]}</td>
<td>{a, b, c, d, e}</td>
<td>{{f, g, h, a, b, c, d, e}}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{s,1,3, t}</td>
<td>{[a, b], [a, c, d, f], [d, e, g, h], [b, i]}</td>
<td>{a, d, b}</td>
<td>{{d, e, g, h, a, b, c}}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{s,1,2,4}</td>
<td>{[a, b], [a, c, d, f], [f, g, i]}</td>
<td>{a, b, c, f}</td>
<td>{{d, e, g, h, a, b, c, f}}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{s,2,3,4}</td>
<td>{[a, b], [b, c, e, f], [f, g, i], [d, e, g, h]}</td>
<td>{b, e, f, g}</td>
<td>{{a, c, d, i, h, b, e, f, g}}</td>
</tr>
<tr>
<td>4</td>
<td>{{s,1,3,4, t}, {s,1,3, 2,4}, {s,1,2,4, t}, {s, 2,3,4}}</td>
<td>{s,1,3,4}</td>
<td>{[a, b], [a, c, d, f], [g, i], [h, i]}</td>
<td>{a, d, g, h}</td>
<td>{{b, c, e, f, a, d, g, h}}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{s,1,3,2,4}</td>
<td>{[a, b], [a, c, d, f], [g, i], [h, i]}</td>
<td>{a, b, c, d, e, f, g}</td>
<td>{{h, i, a, b, c, d, e, f, g}}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{s,1,2,4, t}</td>
<td>{[a, b], [a, c, d, f], [f, g, i], [h, i]}</td>
<td>{a, b, c, f, i}</td>
<td>{{d, e, g, h, a, b, c, f, i}}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{s,2,3,4, t}</td>
<td>{[a, b], [b, c, e, f], [f, g, i], [h, i]}</td>
<td>{b, e, g, h}</td>
<td>{{a, c, d, b, e, f, g, i, h, i}}</td>
</tr>
</tbody>
</table>
4. Results and Discussion

The following benchmark networks (Fig-2) along with the example network are taken into consideration for comparing the proposed method against some existing methods of similar interest. For estimating the $g$-terminal reliability of the said networks the number of spanning trees, the number of minimal cut sets (in disjoint form) along with the no. of self-generating disjoint minimal cut-sets are tabulated in Table 4.

Table 3: Stepwise Illustration of the Algorithm for 2-Terminal Reliability:

<table>
<thead>
<tr>
<th>k</th>
<th>$W_k$</th>
<th>Node Set</th>
<th>Links associated with the node set</th>
<th>$I_1$</th>
<th>$L_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>${1}$</td>
<td>${1}$</td>
<td>${h,i}$</td>
<td>$\emptyset$</td>
<td>${h,i}$</td>
</tr>
<tr>
<td>1</td>
<td>${{3,4}}$</td>
<td>${3,4}$</td>
<td>${h,i,{d,e,g,h}}$</td>
<td>${h}$</td>
<td>$\cup{d,e,i,g,h}$</td>
</tr>
<tr>
<td></td>
<td>${4}$</td>
<td>${4}$</td>
<td>${f,g,i}$</td>
<td>$\emptyset$</td>
<td>$\cup{f,g,h,i}$</td>
</tr>
<tr>
<td>2</td>
<td>${{1,3}, {2,4}}$</td>
<td>${1,3}$</td>
<td>${h,i,{a,c,d},{d,e,g,h}}$</td>
<td>${d,h}$</td>
<td>$\cup{a,c,d,e,g,h}$</td>
</tr>
<tr>
<td></td>
<td>${2,4}$</td>
<td>${2,4}$</td>
<td>${h,i,{b,c,e,f},{f,g,i}}$</td>
<td>${f,i}$</td>
<td>$\cup{b,c,e,f,g,h,i}$</td>
</tr>
<tr>
<td></td>
<td>${3,4}$</td>
<td>${3,4}$</td>
<td>${h,i,{d,e,g,h},{f,g,i}}$</td>
<td>${g,h,i}$</td>
<td>$\cup{d,e,f,g,h,i}$</td>
</tr>
<tr>
<td>3</td>
<td>${{1,3,4}, {1,2,3,4}}$</td>
<td>${1,3,4}$</td>
<td>${h,i,{a,c,d},{d,e,g,h},{g,i}}$</td>
<td>${d,g,h,i}$</td>
<td>$\cup{a,c,d,e,f,g,h,i}$</td>
</tr>
<tr>
<td></td>
<td>${2,3,4}$</td>
<td>${2,3,4}$</td>
<td>${h,i,{b,c,e,f},{d,e,g,h}}$</td>
<td>${e,f,g,h,i}$</td>
<td>$\cup{b,c,d,e,f,g,h,i}$</td>
</tr>
<tr>
<td></td>
<td>${1,2,3,4}$</td>
<td>${1,2,3,4}$</td>
<td>${h,i,{a,c,d},{b,c,e,f},{d,e,g,h}}$</td>
<td>${c,d,e,f,g,h,i}$</td>
<td>$\cup{a,b,c,d,e,f,g,h,i}$</td>
</tr>
</tbody>
</table>

From Table 4, it is quite clear that the proposed method generates very less number of cut-sets in their disjoint form than its counterparts e.g., for network with size $(6N9L)$, the corresponding number of spanning trees and the number of cut-sets (in disjoint form) are
81 and 24(52) respectively, in contrast to which the proposed method generates only 15 cut-sets in their disjoint form. Similarly, for the network having 9 number of nodes and 14 number of links, a remarkable difference in the above values can be observed, which is like 647, 46(242) and 21. From all these observations, the proposed method can be considered to be more efficient than its counterparts in estimating the g-terminal reliability of the said networks. For the evaluation of two terminal reliability of the said networks, the number of minimal path sets (in disjoint form), the number of minimal cut-sets (in disjoint form) and the number of self-generating minimal cut-sets are tabulated in Table-5. From Table-5, it can be observed that, the proposed method provides much less number of disjoint terms in the reliability expression in comparison to the methods [11] and [8].

Table 4: Comparison of the Proposed Method for estimating the g-Terminal Reliability with the methods described in [11] and [8].

<table>
<thead>
<tr>
<th>Network #</th>
<th>Network (N,L)</th>
<th>Number of panning edges</th>
<th>Number of spanning Trees disjoint form</th>
<th>Number of minimal cut sets [8]</th>
<th>Number of Minimal cut sets in disjoint form [8]</th>
<th>Number of Self-generating disjoint minimal cut sets</th>
<th>Reliability (p=0.9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6N9L</td>
<td>81</td>
<td>81</td>
<td>24</td>
<td>52</td>
<td>15</td>
<td>0.9683173</td>
</tr>
<tr>
<td>2</td>
<td>7N10L</td>
<td>96</td>
<td>96</td>
<td>27</td>
<td>56</td>
<td>17</td>
<td>0.9632425</td>
</tr>
<tr>
<td>3</td>
<td>8N11L</td>
<td>168</td>
<td>168</td>
<td>53</td>
<td>133</td>
<td>18</td>
<td>0.9683688</td>
</tr>
<tr>
<td>4</td>
<td>8N12L</td>
<td>247</td>
<td>247</td>
<td>36</td>
<td>122</td>
<td>22</td>
<td>0.9732406</td>
</tr>
<tr>
<td>5</td>
<td>8N13L</td>
<td>576</td>
<td>576</td>
<td>52</td>
<td>222</td>
<td>23</td>
<td>0.9826993</td>
</tr>
<tr>
<td>6</td>
<td>9N14L</td>
<td>647</td>
<td>647</td>
<td>46</td>
<td>242</td>
<td>21</td>
<td>0.9538703</td>
</tr>
</tbody>
</table>

Table 5: Comparison of the Proposed Method for estimating the Two-Terminal Reliability with the methods described in [11] and [8].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6N9L</td>
<td>9</td>
<td>12</td>
<td>11</td>
<td>14</td>
<td>9</td>
<td>0.96232</td>
</tr>
<tr>
<td>2</td>
<td>7N10L</td>
<td>8</td>
<td>11</td>
<td>13</td>
<td>17</td>
<td>10</td>
<td>0.96112</td>
</tr>
<tr>
<td>3</td>
<td>8N11L</td>
<td>9</td>
<td>11</td>
<td>17</td>
<td>23</td>
<td>12</td>
<td>0.96782</td>
</tr>
<tr>
<td>4</td>
<td>8N12L</td>
<td>20</td>
<td>30</td>
<td>20</td>
<td>53</td>
<td>15</td>
<td>0.97728</td>
</tr>
<tr>
<td>5</td>
<td>8N13L</td>
<td>29</td>
<td>77</td>
<td>29</td>
<td>72</td>
<td>15</td>
<td>0.97893</td>
</tr>
<tr>
<td>6</td>
<td>9N14L</td>
<td>44</td>
<td>80</td>
<td>25</td>
<td>80</td>
<td>22</td>
<td>0.95387</td>
</tr>
</tbody>
</table>

Further, the path sets or cut sets generated using the methods [11] and [8] requires an overhead of O(N3) for disjointing. Even through this disjointing process may not be come into count for small networks but for large networks it takes much more computational time before estimating the reliability.

The proposed method is applied to the following regular topologies viz. Hypercube [6], twisted hypercube [16], mesh [17], torus [18] and crossed cube [6] for evaluating their different reliability measures. The numbers of disjoint minimal cut sets in the reliability expression along with the computed values of different reliability measures of the said network are presented in Table 6.
5. Conclusion

In this paper, a new method based on self-generating disjoint minimal cut sets for evaluating the different measures of reliability viz., 2-terminal reliability, k- terminal reliability and terminal reliability of interconnection networks is proposed where the minimal cut sets are generated in their disjoint form and are non-redundant in nature. This eliminates the overhead due to disjointing process.

Table-6: The Proposed Method applied for estimating different Reliability Measures of some Regular Interconnection Networks (p = 0.9)

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Network</th>
<th>g-Terminal</th>
<th>k-Terminal</th>
<th>Two-Terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Num. of Self generating disjoint minimal cut sets</td>
<td>Reliability</td>
<td>Num. of Self generating disjoint minimal cut sets</td>
<td>Reliability</td>
</tr>
<tr>
<td>1</td>
<td>Hyper Cube</td>
<td>41</td>
<td>0.964699</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>Twisted Hyper Cube</td>
<td>41</td>
<td>0.964698</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>3-D Mesh</td>
<td>32</td>
<td>0.983738</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>3-D Torus</td>
<td>121</td>
<td>0.9895567</td>
<td>39</td>
</tr>
<tr>
<td>7</td>
<td>Crossed Cube</td>
<td>45</td>
<td>0.965707</td>
<td>20</td>
</tr>
</tbody>
</table>

References


Pradyumna Kumar Tripathy has completed his M.Tech. Degree in Computer Science from Utkal University, India in 2007. He is now a Ph.D. student in Utkal University, India. Mr. Tripathy is currently working as Asst. Prof. in the Department of Computer Science and Engineering, Silicon Institute of Technology, Bhubaneswar, Odisha, India. His research interests include reliability analysis of interconnection networks, parallel distributed systems and topological optimization of interconnection networks.

Ranjan Kumar Dash received the MCA degree from F.M. University in 2001. He got his Ph.D. in Computer Science from Sambalpur University in the year 2008. He is now working as Reader and Head, CSA Department, College of Engineering & Technology, Bhubaneswar, Odisha, India. His research areas are Reliability of parallel & Distributed system, Analysis of & Design of algorithm. E-mail: drranjandash@gmail.com

Chitta Ranjan Tripathy received the B.Sc. (Engg.) in Electrical Engineering from Sambalpur University and M.Tech. degree in Instrumentation Engineering from Indian Institute of Technology, Kharagpur. He got his Ph.D. in Computer Science & Engineering from Indian Institute of Technology, Kharagpur. He is currently working as a Professor and Head, Department of Computer Sc. & Engg., UCE, Burla, Odisha, India. He has more than 60 publications. in different national and international journals and conferences. His research interest includes dependability, reliability and fault-tolerance of parallel & distributed systems. Email: cr.tripathy1@gmail.com