Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 70 (2015) 205 – 211

Procedia Computer Science

4th International Conference on Eco-friendly Computing and Communication Systems

Disjointing Technique for Reliability Evaluation of Computer-Communication Networks

Kriti Kaushik[a], G.L. Pahuja[b]

[a]Department of Electronics and Communication Engineering, Seth Jai Parkash Mukand Lal Institute of Engineering (JMIT), Radaur (Yamunanagar), 135133, India
[b]Department of Electrical Engineering, National Institute of Technology Kurukshetra, 136119, India

Abstract

Source to all terminal reliability of a network is the ability of the network to transmit the commodity from source node to all other terminal nodes. The paper proposes an algorithm for source to all terminal reliability evaluation of directed networks. The algorithm starts with a spanning tree which is termed as first spanning tree (FST). All other disjoint spanning trees are generated from the knowledge of first spanning tree. Source to all terminal reliability is evaluated by taking probability of union of disjoint spanning trees. The method puts no constraint on FST selection. The proposed method is fast, efficient and no failed spanning tree is generated. The algorithm doesn't generate duplicate spanning trees. The method is explained with the help of an example. The advantage in terms of computational complexity is also compared with the existing techniques.

Peer-review under responsibility of the Organizing Committee of ICECCS 2015

Keywords: Directed network; SAT Reliability; Spanning trees; Reliability Evaluation

1. Introduction

A network can be represented in the form of a graph G (v, b) with v vertices and b edges. More general measure of reliability, Source to terminal, ST, reliability of a network is defined as the ability to communicate from source node to terminal/sink node of the network. With technological advancement, network size has increased and focus has also been on source to all terminal reliability of network. Generally, source to all terminal (SAT) reliability of a network means that required amount of information originating from source should reach all the terminal nodes of a network. In early literature, one of the approaches of calculating source to all terminal reliability was in terms of

Peer-review under responsibility of the Organizing Committee of ICECCS 2015
doi:10.1016/j.procs.2015.10.074

source to terminal reliability computation for every pair of nodes present in the network [12]. This is a highly inefficient method for SAT reliability evaluation. Literature is enriched with a number of approaches of determining SAT reliability of undirected networks. One of the approaches is the spanning tree approach. Aggarwal [2] and Feng, Chan [3] used vertex cutset approach to generate spanning trees. Jain and Gopal generated all the spanning tress of the network by initially choosing a special spanning tree termed as Initial Spanning Tree (IST) [4]. The method needs to store a few spanning trees (depending upon the size of the network) in order to generate rest of the spanning trees. The method is inefficient. In 1989, Page and Perry, used factoring theorem to calculate network reliability of directed networks [13]. They applied factoring theorem along with certain reductions which they modified for directed networks. During 1990s work on evaluating SAT reliability using spanning trees was extended on complex networks like hypernets, hypercubes and star network [5, 6]. Since 1996 most of the work is focused on finding the number of spanning trees for different type of networks [7-11]. Since tree topology is the most adopted topology in wired and wireless networks, latest work is being done on maximizing tree reliability by Genya et.al [1]. In nutshell the existing methods of reliability evaluation are inefficient, lengthy, involved and some are network specific. The proposed method removes all these limitations. The proposed method computes SAT reliability of directed networks. The method initiates with the knowledge of any one directed spanning tree and generates all the disjoint spanning trees from it. The method puts no constraint on the selection of initial spanning tree and has been termed as First Spanning Tree (FST). To avoid the generation of failed spanning trees certain rules have been advanced which decrease the generation of failed spanning trees to zero. The proposed method generates all disjoint spanning trees and from the knowledge of these spanning trees, SAT reliability can easily be computed. This paper is organized in different sections. Section 2 describes the terminology used in the development of the method. Section 3 details the methodology used. Algorithm is explained in Section 4. Section 5 illustrates the algorithm with the help of an example. Section 6 concludes the paper.

2. Terminology

Graph is a way of representing any network. A graph is formed by vertices and edges connecting the vertices. G (v, b) is a graph with v vertices and b edges. A directed graph also called digraph is formed by vertices connected by directed edges. Fig.1 represents graph G(5,7) of a network. Sub graph is a subset of graph. Tree is a connected circuitless graph. Spanning tree of a graph is a subgraph that includes all of the vertices of G that is a tree. Fig.2 shows the directed spanning tree of G (5, 7). Indegree of a node is the number of incoming edges on that particular node in a graph. In Fig.1 two edges are incident on node v3 so indegree of node v3 is two. Outdegree of a node is the number of outgoing edges from that particular node in a graph. In Fig.1, one branch is directed away from v3 so outdegree of node v3 is one. Any spanning tree of a graph can be considered as the First spanning tree (FST). Fig.2 shows the FST for G (5, 7). Successful spanning tree (ST_i) is a spanning tree which is formed by appending appropriate edges with failed FST.

Nomenclature

FST	First spanning tree
S	Set of edges in FST
T	Set of Edges excluded from FST
C	Set of cyclic edges in graph
v	Number of vertices in G (v, b)
b	Number of edges in G (v, b)
l	Number of edges in FST
k	Number of edges in set T
$d \le k$	Number of edges to be failed in IST
m=l	
X_i	Logical variable
$\overline{X_j}$	Complimented logical variable
p_i	Reliability of links

q_j	Unreliability of links
R	Reliability of network
ST_i	Successful spanning trees
$\cup_i$	Union of spanning trees
SSF	System Success Function

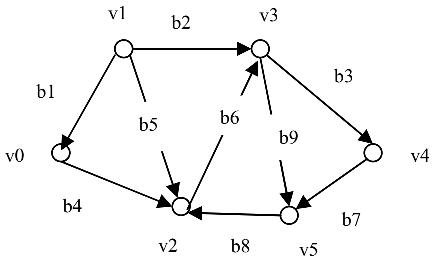

Fig.1 G (6, 9)

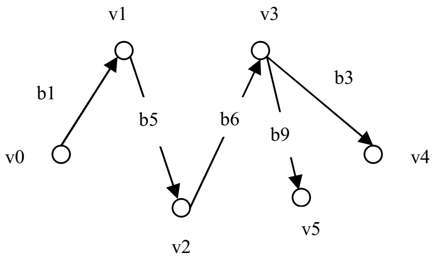

Fig.2 Directed Spanning tree of fig.1

3. Methodology

Given any directed network, the algorithm starts with any spanning tree of the network taken as First spanning tree (FST). Number of edges in FST is one less than total number of nodes in graph (l=v-1). Edges present in FST are moved to set S. Edges which are not present in FST are listed in set T. All cyclic combinations present in graph are moved to set C. Disjoint spanning trees from FST are generated by following the below mentioned procedure:

At first, edges in set S are failed by taking any one edge at a time and appropriate edges from set T are selected based on the indegree of last node (towards which the arrow was pointing) which was disconnected due to edge failure (one at a time) in FST. Those selected edges are tried with failed FST (one edge deleted) in order to see feasibility of formation of disjoint ST_i's. Also, before appending an edge from set T, set C is checked for the avoidance of cyclic combinations. Set C contains all the cyclic combinations of edges present in the network. After this any two edges at a time from set S are failed and all the combinations of two edges at a time from set T are selected based on the indegree of last two nodes in the graph which are disconnected due to edge failure (two at a time) in FST. Those selected edges are tried with failed FST (two edges deleted) in order to obtain disjoint ST_i's. Also, before appending an edge from set T, set C is checked for the avoidance of cyclic combinations. This process terminates when number of failed edges in FST reach the maximum value i.e. 'd'. Every time new edges are appended with failed FST, set C is checked so that generation of nonsuccessful/failed spanning trees is being stopped by not appending the cyclic combination of edges which are present in the graph. The algorithm thus stops the formation of failed spanning trees. Illustration of the methodology for Fig. 1 is shown in Fig. 3.

The system success function, SSF, is obtained by taking union of all disjoint spanning trees. The SAT reliability expression can be obtained by changing:

i) Union operator by summation operator in SSF.

ii) Logical variables of each spanning tree to corresponding probability variables i.e. $X_i \rightarrow p_i$ and $\overline{X_j} \rightarrow q_j$.

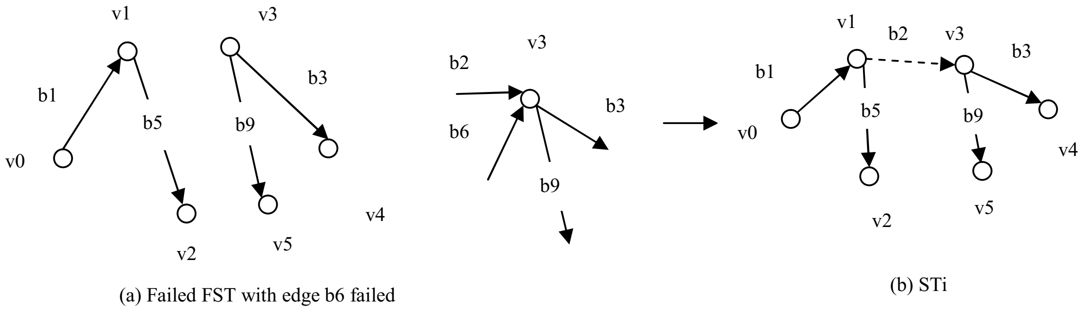

Fig. 3 Illustration of the process of generation of STi from failed FST

4. Algorithm

Step1. Pick up any spanning tree as First spanning tree (FST) which has minimum number of edges forming the link between source node and all other terminal nodes. Edges present in FST are moved to set S.

Step2. All the edges of the network not present in FST are moved to set T.

Step3. (i) There are ${}^{m}C_1$ possible ways a link can fail in FST. The way links from set T are appended with each failed FST depends upon the in degree of last node in a graph which was disconnected due to edge failure in FST. The procedure in (iv) is followed after edge selection from set T. All successful spanning trees (ST_i's) so generated are retained.

(ii) There are ${}^{m}C_2$ possible ways two links can fail at a time in FST. The way two links from set T are appended with each failed FST depends upon the in degree of last two nodes in a graph which were disconnected due to edge failure in FST. The procedure in (iv) is followed after edge selection from set T. All successful spanning trees so generated are retained.

(iii) The procedure in (ii) is repeated with three, more… d failures at a time in FST.

(iv)The generation of nonsuccessful/failed spanning trees is being avoided by checking set C for cyclic formation every time edges are appended with failed FST from set T. This minimizes the formation of failed spanning trees.

Step4. SAT reliability expression can be obtained by taking the union of disjoint spanning trees and changing the logical variables of each spanning tree to corresponding probability variables

5. Illustration

Consider the directed graph G (6, 12) of a network shown in fig. 4. Any spanning tree of the graph is picked up as the First spanning tree (FST) as shown in Fig. 5. In this example S = {2, 3, 8, 10, 12}, T = {1, 4, 5, 6, 7, 9, 11} and C = {9, 12; 8, 11; 7, 10; 6, 9, 11; 4, 7, 11}. When Edge 2 from set S is failed it disconnects node v3 from node v1 (=s) in FST. Indegree of node v3 is two. Edge 9 from set T is selected and set C is checked before appending the edge with failed FST. Appending of edge 9 gives cyclic combination with 12 hence it doesn't yield ST_i and hence dropped. Table 1 show the generation of ST_i's with appended edges from set T. Similarly, other edge failures in set S are considered (one edge at a time) independently and elements from set T are appended with failed FST to check the formation of ST_i's. When two edges at a time are failed in FST, while appending edges from set T in degree of last two nodes which were disconnected due to edge failure in FST, is considered. All cyclic combination of edges which are listed in set C are avoided while selecting edges from set T. This avoids the formation of failed spanning trees. Overall 34 successful spanning trees are obtained. Formation of failed spanning trees is stopped and represented by ($\times$) in Table I. No successful spanning tree repeats in consecutive iterations. The formation of failed spanning trees is nil as compared to our previous method [14]. System success function, SSF, is obtained by taking union of all disjoint ST_i's. Reliability expression is obtained by changing: i) Union operator by summation operator in SSF. ii) Logical variables of each spanning tree to corresponding probability variables. Every term in ST_i contains mixed variable i.e. complimented and uncomplimented. Successful disjoint spanning trees are given below in Table 2.

System success function, SSF, may be written as:

$SSF = \cup_i ST_i$

Reliability expression is obtained as:

$R = P(\cup_i ST_i) = \sum P(ST_i)$

R=p2p3p8p10p12+p2q3p8p10p12p5+p2p3q8p10p12p6+p2p3p8q10p12p4+p2p3p8p10q12p1+q2q12p3p8p10p1p9+p2q3q8p10p12p5p6+p2q3q10p8p12p5p4+p2q3q12p8p10p1p5+p2p3q8q10p12p4p6+p2p3q8q10p12p4p7+p2p3q8q12p10p1p6+p2p3q8q12p10p11p6+p2p3p8q10q12p1p4+q2q3q12p8p10p1p5p9+q2q8q12p3p10p1p6p9+q2q10q12p3p8p1p4p9+p2q3q8q10p12p4p5p6+p2q3q8q10p12p4p5p7+p2q3q8q12p10p1p5p6+p2q3q8q12p10p11p5p6+p2q3q10q12p1p8p4p5+p2p3q8q10q12p8p1p4p6+p2p3q8q10q12p1p4p7+p2p3q8q10q12p4p6p11+q3q2q8q12p10p1p5p6p9+q2q3q10q12p8p1p4p5p9+q2q8q10q12p3p1p4p6p9+q2q8q10q12p3p1p4p7p9+p2q3q8q10q12p1p4p5p6+p2q3q8q10q12p1p4p5p7+p2q3q8q10q12p4p5p6p11+q2q3q8q10q12p1p4p5p6p9+q2q3q8q10q12p1p4p5p7p9

Implementing A. Satyanarayan method [15] on this network generates 34 minimal spanning trees and by using inclusion-exclusion principle reliability expression contains 2^{34}-1 cancelling and noncancelling terms.

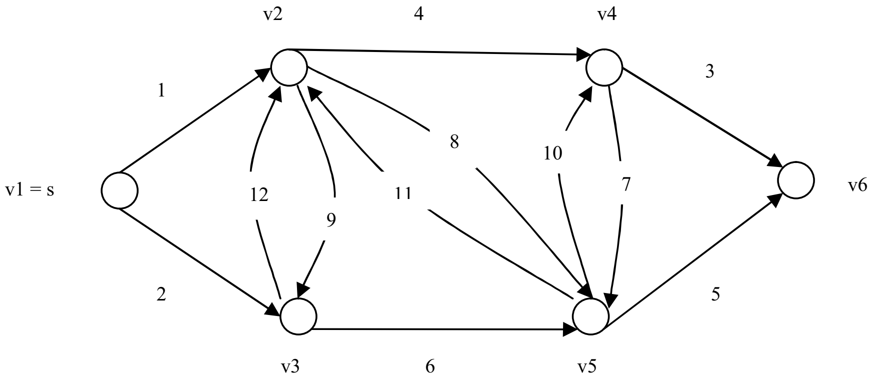

Fig. 4 ARPANET G (6, 12)

Table 1. Generation of successful spanning trees, ST_i

Sr. No.	IST with failed links	Appended edges
1.	$\overline{2}$ 381012	×
2.	2$\overline{3}$ 81012	(5)
3.	23 $\overline{8}$ 1012	(6)
4.	238$\overline{10}$ 12	(4)
5.	23810$\overline{12}$	(1)
6.	$\overline{23}$ 81012	×
7.	$\overline{28}$ 31012	×
8.	$\overline{210}$ 3812	×
9.	$\overline{212}$ 3810	(19)
10.	2$\overline{38}$ 1012	(56)
11.	2$\overline{310}$ 812	(54)
12.	2$\overline{312}$ 810	(15)
13.	2312$\overline{810}$	(46), (47)

14.	$23\overline{812}\ 10$	(16) , (116)
15.	$238\overline{1012}$	(14)
16.	$\overline{238}\ 1012$	×
17.	$\overline{2310}\ 812$	×
18.	$\overline{2312}\ 810$	(159)
19.	$\overline{2810}\ 312$	×
20.	$\overline{2812}\ 310$	(169)
21.	$\overline{21012}\ 38$	(149)
22.	$212\overline{3810}$	(456) , (457)
23.	$210\overline{3812}$	(156) , (1156)
24.	$28\overline{31012}$	(145)
25.	$23\overline{81012}$	(146) , (147) , (4611)
26.	$12\overline{23810}$	×
27.	$10\overline{23812}$	(1569)
28.	$8\overline{231012}$	(1459)
29.	$3\overline{281012}$	(1469) , (1479)
30.	$2\overline{381012}$	(1456) , (1457) , (45611)
31.	$\overline{2381012}$	(14569) , (14579)

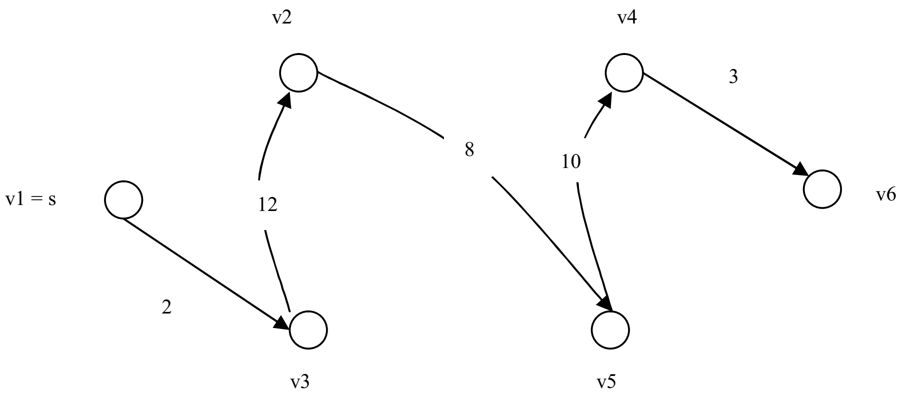

Fig. 5 FST of G (6, 12)

Table 2. Disjoint spanning trees, ST_i

Sr. No.	Spanning trees	Sr. No.	Spanning trees
1.	2381012	19.	$2\overline{3810}\ 12\ 457$
2.	$2\ \bar{3}\ 81012\ 5$	20.	$2\overline{3812}\ 10\ 156$
3.	$23\ \bar{8}\ 1012\ 6$	21.	$2\overline{3812}\ 10\ 1156$
4.	$238\ \overline{10}\ 12\ 4$	22.	$2\overline{31012}\ 8\ 145$
5.	$23810\ \overline{12}\ 1$	23.	$23\overline{81012}\ 146$
6.	$\overline{212}\ 3810\ 19$	24.	$23\overline{81012}\ 147$
7.	$2\overline{38}\ 1012\ 56$	25.	$23\overline{81012}\ 4611$
8.	$2\overline{310}\ 812\ 54$	26.	$\overline{23812}\ 10\ 1569$

9.	$2\overline{312}$ 810 15	27.	$\overline{231012}$ 8 1459
10.	$23\overline{810}$ 12 46	28.	$\overline{281012}$ 3 1469
11.	$23\overline{810}$ 12 47	29.	$\overline{281012}$ 3 1479
12.	$23\overline{812}$ 10 16	30.	$2\overline{381012}$ 1456
13.	$23\overline{812}$ 10 116	31.	$2\overline{381012}$ 1457
14.	$238\overline{1012}$ 14	32.	$2\overline{381012}$ 45611
15.	$\overline{2312}$ 810 159	33.	$\overline{2381012}$ 14569
16.	$\overline{2812}$ 310 169	34.	$\overline{2381012}$ 14579
17.	$\overline{21012}$ 38 149		
18.	$2\overline{3810}$ 12 456		

6. Conclusion

The proposed method follows one step approach. It is very easy to find the number of spanning trees using this method. The advantage of this method is that it requires only one spanning tree (FST) to begin with. Moreover, this method puts no constraint on FST selection. As compared to our previous method [14] this method results in the generation of no failed spanning tree. The proposed method is simple and requires less memory space. Hence it is fast computationally. Using A. Satyanarayan [15] method 34 minimal spanning trees are generated and 2^{34}-1 cancelling and noncancelling terms are obtained in reliability expression. The method is executed in two steps. The method of Aggarwal [2] is also a two step approach; at first it yields minimal spanning trees and then uses a disjointing technique to obtain reliability expression in terms of disjoint spanning trees. Method is computationally less attractive but reliability expression consists of same number of terms as with the proposed method.

References

1. Ishigaki, G.; Yoshida, M.; Shinomiya, N., "On maximizing tree reliability based on minimum diameter spanning tree," IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2014, vol., no., pp.603,606, 17-20 Nov. 2014
2. Aggarwal, K.K.; Rai, S., "Reliability Evaluation in Computer-Communication Networks," IEEE Transactions on Reliability, vol.R-30, no.1, pp.32,35, April 1981
3. Feng, H.; Chan, S.-P., "A method of reliability evaluation for computer-communication networks, " IEEE International Symposium on Circuits and Systems, 1990., vol., no., pp.2682,2684 vol.4, 1-3 May 1990
4. Jain, S.P.; Gopal, K., "An efficient algorithm for computing global reliability of a network," IEEE Transactions on Reliability, vol.37, no.5, pp.488,492, Dec 1988
5. Kaur, D.; Singh, H.; Kaushal, R.P., "Reliability evaluation of hypercube and hypernets using spanning tree approach," Proceedings of the 32nd Midwest Symposium on Circuits and Systems, 1989., , vol., no., pp.927,930 vol.2, 14-16 Aug 1989
6. Fragopoulou, P.; Akl, S.G., "Edge-disjoint spanning trees on the star network with applications to fault tolerance," IEEE Transactions on Computers, vol.45, no.2, pp.174,185, Feb 1996
7. Bapeswara Rao, V.V., "Most-vital edge of a graph with respect to spanning trees," IEEE Transactions on Reliability, vol.47, no.1, pp.6,7, Mar 1998
8. Jinn-Shyong Yang; Jou-Ming Chang; Shyue-Ming Tang; Yue-Li Wang, "Reducing the Height of Independent Spanning Trees in Chordal Rings," IEEE Transactions on Parallel and Distributed Systems, vol.18, no.5, pp.644,657, May 2007
9. Modabish, A.; Lotfi, D.; El Marraki, M., "The number of spanning trees of planar maps: Theory and applications," International Conference on Multimedia Computing and Systems (ICMCS), 2011, vol., no., pp.1,6, 7-9 April 2011
10. Yuzhi Xiao; Haixing Zhao, "Counting the number of spanning trees of generalization Farey graph," Ninth International Conference on Natural Computation (ICNC), 2013, vol., no., pp.1778,1782, 23-25 July 2013
11. Lotfi, D.; El Marraki, M.; Aboutajdine, D., "Spanning trees in a closed chain of planar networks," International Conference on Multimedia Computing and Systems (ICMCS), 2014, vol., no., pp.1229,1234,14-16 April2014
12. Wilkov,R "Analysis and Design of Reliable Computer Networks," IEEE Transactions on Communications, vol.20, no.3,pp.660,678,Jun1972
13. L.B. Page, J.E. Perry "Reliability of directed networks using the factoring theorem," IEEE Transactions on Reliability, vol.38, no.5, pp.556,562, Dec 1989
14. G.L. Pahuja, Kriti Kaushik, Ashish Sachdeva "SAT Reliability Evaluation of Directed Networks using Disjointing Technique" communicated.
15. A .Satyanarayana, Jane N. Hagstrom "A New Algorithm for the Reliability Analysis of Multi-Terminal Networks," IEEE Transactions on Reliability, vol.R-30, no.4, pp.325, 334, Oct. 1981