On Survivability Testing and Computing the Node Connectivity of a Topological Structure of a Interconnection Network Graph

Nethravathi B[1], Kamalesh V N[2]
[1]Department of ISE, JSSATE, Bangalore, [2]Department of Science and Engg., TJIT, Bangalore,
[1]nethravathi.sai@gmail.com, [2]kamalesh.v.n@gmail.com

ABSTRACT- One of the major objectives in the design of topological structures of interconnection survivable computer communication networks is the construction of k-node connected networks. This is to achieve maximum survivability in the presence of link or node failure. Once a potential network structure is constructed by an existing algorithm or network is synthesized by merging other networks, and a claim is made that the generated network topology is k-node connected. There are few heuristics available in literature to verify the same. This research article reviews and analyzes the most popular methods and strategies for testing survivability of a topological structure of a interconnection network graph.

Keywords: physical network topology; topological design; fault tolerance; survivability; k-connected;

I INTRODUCTION

The topological design of an interconnection network structures specifies how the communication nodes of a network are interconnected by links. This is a significant stage in the design of survivable network structure, because the routing algorithm flow control, transmission delay etc., depends mainly on the designed network topological structures. [2][8]

The mathematical graph theory concepts and ideas are extensively used in computer science applications especially in the field of computer networks. [4]

The topological structure of network can be represented and mathematically modeled by using a simple graph where the vertices of the graph represent the processing components (nodes) of the network and the edge represents the communication links. On the other way, any simple graph represents a topological structure of some networks. Thus the mathematically graph models and the network topological structures are one and the same. [8][4]A linear graph is an order pair or sets G = (V,E), where set V = { v_1, v_2, …} whose elements are called vertices of the set E = {e_1,e_2 , … } whose elements are called the edges, such that each edge e_i is associated with a pair of vertices that is e_j = (v_i v_j).A simple graph is a graph with no parallel edges and loops. A graph with no loops is a multigraph.A path P in a graph G = (V,E) is an alternating sequence of distinct vertices and distinct edges starting and ending with vertices.

A graph G = (V,E) is said to be connected if there exists at least one path between every pair of vertices otherwise it is said to be disconnected. A component of a graph G is a maximal connected sub graph. A disconnected graph has more than one component, whereas a connected graph has exactly one component.

Two vertices v_i,v_j in a graph G (V,E) are said to be adjacent if they are connected by an edge. Two edges are said to be adjacent, if they share a vertex. Degree of a vertex v in an undirected graph G = (V,E) is the number of edges incident on it. Degree D (G) of a graph G = (V,E) is the smallest degree of all the vertices in G.

$D(G) = \min\{\deg(v) \mid v \text{ in graph } G\}$.

The vertex-connectivity (node connectivity) of a connected graph G is denoted by Κ (G). It is the minimum number of vertices whose removal can either disconnect G or reduce it to a 1-vertex graph.

The edge connectivity (link connectivity) of a connected graph G denoted by **λ**(G), is the minimum number of edges whose removal can disconnects G.

A vertex-cut in a graph G is a vertex set U such that G-U has more components than G. An edge-cut in a graph G is a set of edges D such that G-D has more components than G.

The relation between node connectivity, link connectivity and degree of a graph is given by Whitney‘s inequality and is as follows

$K(G) \leq \lambda(G) \leq D(G)$ . . .(1)

From (1) it is the node connectivity more crucial than the link connectivity because in equation (1) if K is more or higher, then definitely Z is always higher, that is, in particular, if survivable network can withstand the loss of p nodes, it can also definitely withstand the loss of p links.

If a graph is *k*-connected, there are k node disjoint paths between any two distinct nodes. The degree of survivability of a network improves ask is made larger and larger.

The survivability of a network is the fundamental issue in the design of interconnection network. The networks are survivable if their underlined network topological structure is fault tolerant. It is the ability of the network system to execute the specified algorithm correctly regardless of the hardware failure and the program errors. If the physical component of the network system fails then its function has to be performed by other components of the same network system.

It is essential to have proper yardstick or a mathematical abstraction to measure the effect of the fault tolerance. A deterministic graph theoretical based measure for fault tolerance is k- node connectivity of network.[1][2][3][8][18]

II EXPLORATION OF THE RESOURCE MATERIALS, EXISTING HEURISTICS AND METHODS

A network engineers and mathematicians have proposed few algorithms and heuristics for computation or testing of node connectivity of interconnection network graphs.

The earliest method in this field is due to kleitman [17]. This method is applicable for the large network graphs. It makes use of ${}^{r}c_2 + {}^{r}c_{(k-r)}$ verifications instead of ${}^{k}C_2$ verifications. The input for this method is the given network graph N with n nodes and the number k (node connectivity number). The algorithm gives the output whether the given network is k - connected or not. The method starts by choosing any node n_1 and checks whether there exist k node disjoint paths between n1 and every other nodes of the network graph N.

If there does not exist k-node disjoint path, then network is declared same that it is not k-connected else delete the node n_1 and all the links/branches incident to the node n1 and obtained the network graph (N – { n_1}). Now choose any node n_2 for the network graph (N – { n_1}) and check for the (k-1) node disjoint path between n_2 and every other nodes of (N – { n_1}). If we get the required number of node disjoint path, continue the process, if not stop and declare the same, that the network is not k –connected. By repeating the above process k – times, we end up a network graph (N – { $n_1, n_2 \ldots n_k$}). This graph should be 1 - connected graph if not, graph is not k-connected.Tarjan [16] in his article has presented a method for checking 2-connected networks using the technique of Depth First Search. It is a linear time algorithm and he has illustrated the same.Hopcraft and Tarjan [15] have presented a technique for checking 3-node connectedness of a given network graph by dividing the network graphs in to triconnected components. It is a linear time algorithm. He proves that the method is theoretically optimal to within a constant factor and efficient in practice.S. Even and Tarjan [14] has described a model for the maximal stationary flow using the algorithm of Dinic by assuming that the capacities of node and links as one. Further as an application of the above result, they calculate the node connectivity of the network graph.S. Even [13] has presented an algorithm for testing whether the node connectivity for large network graph N with n number of nodes is at least k. This method works for undirected graph. He has also shown the above method also works for the directed graph with variance in algorithm and it starts by identifying the nodes of the network graph by using the counting natural numbers starting from

This algorithm has two phases. In the first phase; the first k nodes are checked for k disjoint paths and in the second phase, the new node X is augmented to original network by connecting X to every node of specially considered set L and check for k disjoint path between X and j where j Є L. For any j if there are less than k disjoint path, the network is not k connected, else k connected. This is repeated for all j belongs to L.

Becker[12] adds a probabilistic variant to the Even & Tarjan method with the minimum error probability in expected time for sparse group with the condition

$$P_r(\mu > k) \leq \varepsilon,$$

Where $\mu = \min N(a,b)$ and $N(a,b) = \min (|S| \leq V-\{a,b\}$ is an (a,b) vertex separator).

Esfahanian and Hakimi [11] presented a technique for computing k-edge connectivity of a network graph or a digraph with n/2 calls to the maxflow algorithm. Using this they have described a method to determine the given network is at least k-node connected by using n-k+1/2(k-1)(k-2) calls.

Kanevsky and Ramachandran [10] have presented a sequential algorithm to check the network graph for 4-node connectivity based open ear decomposition. Each vertex of the network graph is deleted and the resultant network in turn tested for 3-connectivity.

Joseph cherrian *et.al* [7] has presented randomized approach based deterministic approach testing methods for K vertex connectivity for a directed graph using Monte Carlo algorithm.

M. R. Henzinger and Rao[9] have presented deterministic algorithm to compute the node connectivity and finding the corresponding separator of the network graph. The main algorithm is the generalization of the previous preflow-push algorithm for network flow.

Srivatsa et al [6] have presented a method to compute the node connectivity of a network graph by using the adjacency matrix, in their study of survivability of high speed topological network structures. It is a non iterative algorithm. The algorithm begins by numbering the node by appropriate numbering technique. Subsequently node disjoint paths between given pair of nodes is calculated by using the adjacency relationship between the pair of nodes by using the matrix data structures. In their methods the nodes of graph are partitioned in to 2 sets L_1 and L_2, and the corresponding counters are created. Using the adjacency relationship counters are updated. Once all the nodes corresponding counter values are checked for adjacency then the values of the counters are sorted. The lowest value of the sorted counter list gives the node connectivity number of the network graph.

III. ALGORITHMIC PROFILE ANALYSIS, RESULTS AND DISCUSSIONS

The heuristics, algorithms and methods presented by various authors on Survivability testing and computing the node connectivity of a topological structure of an interconnection network graph are studied in detail. A theoretical profiling of the well known methods is assessed by detailed examination.

Based on the results, in this section we portray the pros and cons of the above proposed heuristics and algorithms are discussed below in brief.

The Kleitman [17] method is the oldest method and is widely used by mathematicians and early network engineers. This method just checks whether the given network is k-connected or not.

The methods due to Tarjan [16], Hopcroft & Tarjan [15], Kanevsky & Ramachandran [10] are very specific methods

applicable only to check for 2-node connectivity, 3-node connectivity and 4-node connectivity respectively.
The methods due to Kleitman [17] and S.Even [13] are the methods which just tests or checks for the given k-connectivity, *i.e*. it takes the network graph N (V,E) the number of nodes n and the connectivity number k and the expected output is given network is k-connected or not. To calculate or compute connectivity of the network graph one has to execute the algorithm recursively for k=1, 2, 3…
However the method due to Henzinger & Rao [9], Esfahanian & Hakimi [11] and S K Srivatsa [6] are generalized methods for computing the node connectivity of a given network that is these algorithm takes the network graph N (V,E) as an input and output the corresponding node connectivity k. Esfahanian & Hakimi [11] is based on max flow algorithm and applicable for both undirected and directed network graph. The method due to Henzinger & Rao [9] is a randomized and deterministic algorithm.
The method due to joseph cherrian *et al* [7] is applicable for digraphs. In reality, these methods applicable only for simplex mode communication.
S K Srivatsa [6] method is a direct and generalized method which makes uses of one dimensional and two dimensional data structure for programming. The method takes the entire network details as input to the algorithm and output the connectivity number k. This method is applicable and suitable for the design of small and medium network with optimal number of closed circuits passing through a single node.

The detailed analysis of the above work is tabulated in table1.

IV. CONCLUSION

This research article has presented the explorative and comprehensive review of the existing methods and recent developments on survivability testing and computing the node connectivity of a topological structure of a interconnection network graph. In the analysis the methods are classified as iterative and non iterative approaches applicable for either testing or computing. It is observed from our analysis that the previous methods are restricted only to either checking the node connectivity or testing. However there is no single method which attempts to address both completely, hence there is a scope for further research.

REFERENCES

[1] Santanu Saha Ray, 2013, *Graph Theory with algorithms and its applications,* First Edition, Springer Publishers.

[2] Andrew S. Tanenbaum, 2011, *Computer networks,* 5th Edition, pearson education.

[3] Maarten van Steen, *An Introduction to Graph Theory and Complex Networks*, Copyrighted material - January 2010.

[4] S.G.Shirinivas, S.Vetrivel, Dr. N.M.Elango, *Applications of graph theory in computer science an overview*, International Journal of Engineering Science and Technology Vol. 2(9), 2010.

[5] Abdol-Hossein Esfahanian *On the Evolution of Graph Connectivity Algorithms*, Computer Science and Engineering Department, Michigan State University. East Lansing, Michigan. http://www.cse.msu.edu/eesfahani/book chapter/ Graph connectivity chapter.pdf

[6] Kamalesh V. N**.** and Srivatsa S. K. **(**2010**),** *Adjacency matrix based method to compute the node connectivity of a computer communication network*, Int. Journal of Computer Science Issues, vol. 7, pp. 62-67.

[7] Joseph Cheriyan and John H. Reift, 2005, Vol. 14, No. 4, pp. 435-451, *Directed s-t numberings, rubber bands, and testing digraph k-vertex connectivity*, Springer Berlin / Heidelberg.

[8] Junming Xu, 2004, *Topological structure and analysis of interconnection networks*, Second Edition, Kluwer Academic Publishers.

[9] M. R. Henzinger, S. Rao and H. N. Gabow, *Computing vertex connectivity: new bounds from old techniques*, Proc. 37th IEEE F. O. C. S. (1996), 462— 471.

[10] Kanevsky and V. Ramachandran, *Improved algorithms for graph fourconnectivity*, J. Comp. System Sci 42 (1991), 288—306.

[11] H. Esfahanian and S. L. Hakimi, *On computing the connectivities of graphs and digraphs*. Networks (1984), 355-366.

[12] M. Becker, W. Degenhardt, J. Doenhardt, S. Hertel, G. Kaninke, and W. Keber, *A probabilistic algorithm for vertex connectivity of graphs.* Inf. Proc. Letters 15 (1982), 135-136.

[13] S. Even, *An algorithm for determining whether the connectivity of a graph is at least k*, SIAM J. Computing 4 (1975), 393-396.

[14] S. Even and R. E. Tarjan, *Network flow and testing graph connectivity*, SIAM J. Computing 4 (1975), 507-518.

[15] J. Hopcroft, R.E. Tarjan, *Dividing a graph into triconnected components*. SIAM J. Comput 2 (1973), 135-158.

[16] R. E. Tarjan, *Depth first search and linear graph algorithms*. SIAM J. Comput 1 (1972), 146-160.

[17] D. J. Kleitman, *Methods for investigating connectivity of large graphs*, IEEE Trans. Circuit Theory CT16 (1969), 232-233.

[18] Kamalesh V.N, PhD thesis , *Topological Design of Networks using Graph Theoretical Algorithms*, 2010.

[19] Nethravathi B, Kamalesh V N, Manjunath Guru V.G, *Topological Design of Computer Communication Network Structures : A Comprehensive Review*, accepted for publication at Indian Journal of Science and Technology.