ISSN No. 0976-5697

Volume 4, No. 4, March-April 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Approach to Merging of connected network topologies for Improving Ultimate Connectivity in Grid Backdrop

P.Priyanga*, R.G.Gopeeka and S.Lakshmi Prabha
B.E Computer Science and Engineering
SNS College of Technology Coimbatore
priyapalani.bharad@gmail.com*, gopeerg@gmail.com andsruthiprabha480@gmail.com

Abstract: In this paper, we are merging the network topologies for maximum connectivity. Here, we are introducing a method for merging an m-connected network topology and an n-connected network topology, where, m is greater than n, into a single network topology of connectivity m, without disturbing the dwelling links and geographical locations of the nodes. The proposed method is a first of its kind in the area of research concerned with merging networks for ultimate connectivity. The suggested method is made clear in section 2 and illustrated in section 3.

Keywords: Network topology, linking number, interconnection network, ultimate connectivity

I. INTRODUCTION

Applications of computer communication networks are increasing in every field of human activity. Computer communication networks have many applications in the field of education, business, media, multi-player computer gaming, etc. All these applications demand the design of efficient fault tolerant survivable computer communication network topology with minimum transmission delay, response time and maximum throughput [1] [2]. The energy, memory, transmitting and computing power of the nodes is to be limited in the adhoc design network.

The topological structure of interconnection network can be modeled by a uncomplicated graph, whose vertices represent components of the network and whose edges epitomize physical communication links, where directed edges represents one way communication links and undirected edges represent two way communication links. The incidence function specifies a way that the components of the network are interconnected by links. Such a graph is called the topological structure of the interconnection network, or, in short, network topology. Conversely, any graph can also be considered as a topological structure of some interconnection network. Topologically, graphs and interconnection networks are identical. [3].

II. PROPOSED METHOD

This section presents the proposed method for merging an *m*-connected network topology and an *n*-connected network topology where *m* is greater than *n*, into a single network topology of connectivity *m*, without disturbing the existing links and geographical locations of the nodes. To start with, a network graph G_m (V_m, E_m) of connectivity *m* and another network graph G_n (V_n, E_n) of connectivity *n* are considered. The nodes of both the topologies are numbered by using decimal numbers [4].

Tables T_m and T_n are constructed for both the network graphs G_m and G_n with the fields- node, degree of the node and linking number. The linking number of all the nodes of G_m in table T_m is initialized to the corresponding degree of the nodes. The linking number of all the nodes of G_n in table T_n is initialized to zero. The counter Z is initialized to zero.

If connectivity of both the graph are contra distinct, then cut sets of all cardinalities C_i are constructed for the network graph G_n, where $n \leq C_i < m$, i.e $C_i = n, n+1, n+2, \ldots, m-1$[5].

For each cutset, corresponding to the cardinality C_i the components S of G_n are constructed and the resultant components are arranged in an increasing order of the number of vertices. The counter Z is incremented by one, when the linking number of all the nodes in Table T_n is increased by one [6] [7]. The resultant network graph G so obtained will be the network graph with connectivity *m*.

Algorithm: Network Merging

Input:

(a). G_m (V_m, E_m) : network with connectivity *m*

(b). G_n (V_n,E_n) : network with connectivity *n*

Output: G (V, E): Merged network with connectivity max (*m*, *n*)

Method: Initialize G to a disconnected network with two components G_m and G_n.

a. Number the nodes of network G_m
b. Number the nodes of network G_n
c. Construct table T_m for the network $G_m$with following field (Node, Degree of node, Updated degree of node)
d. Initialize updated degree of node of two degree of the corresponding nodes
e. Construct Table T_n for the network G_n with following field (Node, Degree of node, Updated degree of node)
f. Initialize updated degree of node of T_n to 0
g. Set Z = 0.
h. If (*m* = *n*) then
   (a). Select first *m* vertices from both G_m and G_n and establish links between selected vertices.
   Else
i. Construct cut sets for network G_n of all cardinalities C_i, such that *ṇ* C_i<*m*, where i= *n*,*n*+1,*n*+2,...,*m*-1
j. For each cardinality C_i do
   (a). Construct the components S corresponding to the cutsets.
   (b). Arrange the resultant components S in ascending order of nodes.
   (c). For each S_j S
i. If updated degree of every vertex is Z, then select a node from S_i which has least degree. If there is a tie then select a node (X) with least node label index.

ii. Select a vertex from G_m with least update degree, if there is a tie, select a node with least degree, if there is a tie select a node (Y) with least node label index.

iii. Update the network G by establish a link between X and Y.

iv. Update the update degree corresponding to X and Y in Table 1 and Table 2 respectively.

For End.

If there exist any node (A) in G with degree less then C_i+1, then establish a link between (A) and a node (B) of G_m (selected as described in step 2), and subsequently update the update degrees corresponding to A and B in table 1 and table 2.

Increment Z by 1 when the updated degree of all the nodes of the network 2 is increased by at least 1.

For end.

Algorithm end

III. ILLUSTRATION

In this section, the proposed method is illustrated by taking two network graphs $G_m$and G_n, where G_m is a 5-connected and G_n is 1-connected network. The merged graph G will be 5-connected [8].A 5-connected graph G_m and 1-connected network graph G_n are shown in Figure 1 and Figure.2

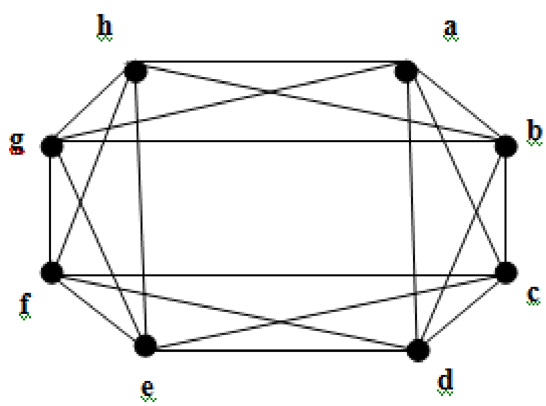

Figure: 1 5-connected network graph

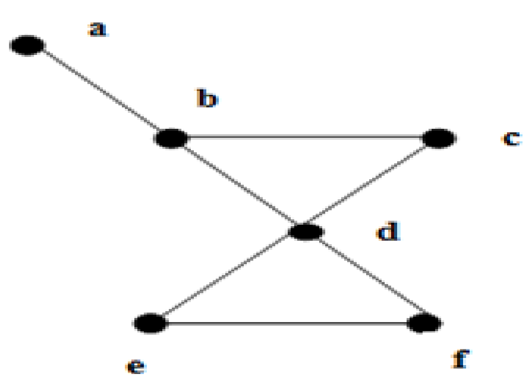

Figure: 2 1-connected network graph G_n

For the network graph G_m the Table1 is constructed with the fields- node, degree of the node, linking number. To start with, the linking number of every node of the graph G_m is same as the degree of the node [9][10].

Table: 1 Information table for network graph G_m

Nodes	Degree of the nodes	Linking Number				
		Series 0	Series i	Series ii	Series iii	Series iv
a	5	e	f	f	g	g
b	5	e	f	f	g	g
c	5	e	e	f	f	g
d	5	e	e	f	f	g
e	5	e	e	f	f	g
f	5	e	e	e	f	g
g	5	e	e	e	f	g
h	5	e	e	e	f	g

The main advantage of this method is that it is based on the concept of cut sets. For the network graph G_n, the Table is constructed with the fields- node, degree of the node, linking number. To start with, the linking number of every node of the graph G_n is initialized to zero. The component S_1 is considered. This component has only one node, i.e node ***a***. The linking number of node ***a*** equals 0, which is less than or equal to Z. This node is selected. A node from G_m which has least degree is selected. If there is a tie, a node which has the smallest node number is selected. Since, all nodes of G_m have the same linking number and same degree, the node ***a*** of G_m is selected as it has the least node index label. A link between node ***a*** of G_m and node ***a*** of G_n is established. Simultaneously, the linking number of both the nodes is increased by 1[11].

The component S_2 is considered. This component has two nodes namely node ***e*** and node ***f***. The degrees of nodes ***e*** and ***f*** are the same. Therefore node number ***e*** is selected from this component as ***e*** is less than ***f***. In G_1, the linking numbers of nodes ***b, c, d, e, f, g, h*** are less than the linking number of node ***a***. Further, the degree of every node is the same. Since, node ***b*** has the least node index number, it is selected. A link between node ***b*** of G_m and node ***e*** of G_n is established. Simultaneously, the linking number of both the nodes is increased by 1.The component S_3 is considered. Here, the linking number of all the nodes is not less than or equal to Z. Therefore, no node is selected from this component. For similar reasons, no node is selected from the component S_4 [12].

The resultant graph G from the above iteration is a 2-Connected network graph and is as shown in Figure 3.

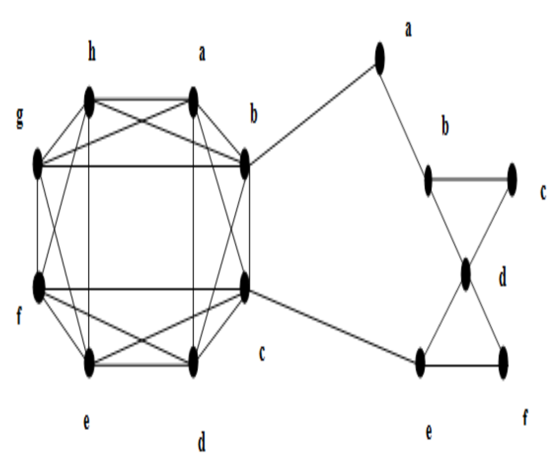

Figure: 3 2-connected merged network graph

The counter Z is not incremented as linking number of all the nodes are not increased by one.

The components corresponding to the cut-set of C_2 are constructed and the components are arranged in the increasing order of their cardinality. The component S_1= {***a***} is considered. Since, the linking number of the node ***a*** in G_n equals 1 which is greater than Z, node ***a*** is not selected. For similar reasons, node ***e*** of S_2 is not selected. The component S_3={***f***} is considered. The linking number of node ***f*** equals 0 which is less than or equal to Z. In G_m, the linking numbers of nodes ***c, d, e, f, g*** and ***h*** are less than the linking number of node ***a*** and ***b***. Further, the degree of every node is the same. Since node ***c*** has the least node index number, it is selected. A link is established between node ***c*** of G_m and node ***f*** of G_n

The component S_6 is considered. Here, the linking number of all the nodes is not less than or equal to Z. Therefore, no node is selected from this component. For similar reasons, no node is selected from the component S_6, S_7 and S_8.The component S_5= {***b, c***} is considered. Here, the linking numbers of both the nodes are equal to zero, which is less than or equal to counter Z. Therefore, a node from S_5 with least degree is selected. Hence, node ***c*** is selected. In G_m, the linking numbers of nodes ***d, e, f, g*** and ***h*** are less than the linking number of node ***a, b*** and ***c***. Further, the degree of every node is the same. Since node ***d*** has the least node index number, it is selected [13]. A link is established between node ***d*** of G_m and node ***c*** of G_n.

Since, the degree of every node in the resultant graph should be greater than or equal to 3 after this iteration, it is observed that the degree of the node ***a*** of G_n is 2. Hence, the node ***a*** of G_n is selected, in G_n. The linking numbers of nodes ***e, f, g*** and ***h*** are less than the linking number of node ***a, b, c*** and ***d***. Further, the degree of every node is the same. Since node ***e*** has the least node index number, it is selected. A link is established between node ***e*** of G_m and node ***a*** of G_n. The counter Z is not incremented as linking number of all the nodes are not increased by one. The components corresponding to the cut-set of C_3 are constructed and the components are arranged in an increasing order of their cardinality. The component S_1= {a} is considered. Since, the linking number of the node ***a*** in G_n equals b which is greater than Z, node ***a*** is not selected. For similar reasons, node ***e*** of S_3 and node ***f*** of S_4 is not selected.

The component S_2= {b} is considered. The linking number of node ***b*** equals 0 which is less than or equal to Z. In G_m, the linking numbers of nodes ***c, d, e, f, g*** and ***h*** are less than the linking number of node ***a*** and ***b***. Further, the degree of every node is the same. Since node ***c*** has the least node index number, it is selected. A link is established between node ***c*** of G_m and node ***d*** of G_n. Simultaneously, are updated the corresponding tables.

The component S_5 is considered. Here, the linking number of all the nodes is not less than or equal to Z. Therefore, no node is selected from this component. For similar reasons, no node is selected from the component S_6, S_7, S_8, S_9 and S_{10} [14].

Since the degree of every node in the resultant graph should be greater than or equal to 4 after this iteration, it is observed that the degree of the nodes ***a, c, e*** and ***f*** does not satisfy this condition. Hence, the nodes ***a, c, e*** and ***f*** of G_n, are selected, in G_m. The linking numbers of nodes ***g*** and ***h*** are less than the linking number of node ***a, b, c, d, e*** and ***f***. Further, the degree of every node is the same. Since, node ***g*** has the least node index number, it is selected. A link is established between node ***g*** of G_m and node ***a*** of G_n. Similarly, links are established between node ***h*** of G_m and node ***c*** of G_n, node ***a*** of G_m and node ***e*** of G_n and node ***b*** of G_m and node f of G_n. The corresponding tables are updated. The resulting network graph is shown in Figure 4

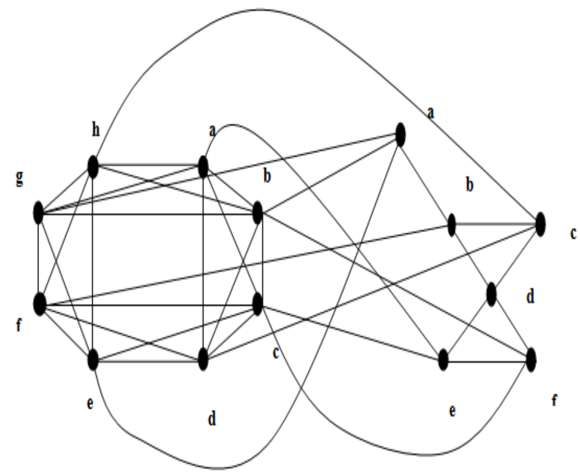

Figure: 4 4-connected merged network graph

The counter Z is not incremented as linking number of all the nodes are not increased by one. The components corresponding to the cut-set of C_3 are constructed and the components are arranged in an increasing order of their cardinality. The component S_1={a} is considered. Since, the linking number of the node ***a*** in G_n equals 3 which is greater than Z, node ***a*** is not selected. For similar reasons, node ***b*** of S_2, node ***e*** of S_4 and node ***f*** of S_5 is not selected. The component S_3= {d} is considered. The linking number of node ***d*** equals 0 which is less than or equal to Z. In G_m, the linking numbers of nodes ***f, g*** and ***h*** are less than the linking number of node ***a, b, c, d*** and ***e***. Further, the degree of every node is the same. Since, node ***f*** has the least node index number, it is selected. A link is established between node ***f*** of G_1 and node ***b*** of G_2. Simultaneously, the corresponding tables are updated.

Since, the degree of every node in the resultant graph should be greater than or equal to 5 after this iteration, it is observed that the degree of the nodes ***a, b, c, e*** and ***f*** of G_2 does not satisfy this condition. Hence, the nodes ***a, b, c, e*** and ***f*** of G_n are selected, in G_m. The linking numbers of nodes ***d, e, f, g*** and ***h*** are less than the linking number of node ***a, b*** and ***c***. Further, the degree of every node is the same. Since, node ***d*** has the least node index number, it is selected. A link is established between node ***d*** of G_m and node ***a*** of G_n. Similarly, links are established between node ***e*** of G_m and node ***b*** of G_n, node ***f*** of G_m and node ***c*** of G_n, node ***g*** of G_m and node ***e*** of G_n and node ***h*** of G_m and node ***f*** of G_n. The corresponding tables are updated.

IV. CONCLUSIONS

The main advantage of this method is that it is based on the concept of cut sets. The connectivity of the merged graph increases by one after every iteration. Hence, we get all possible networks with connectivity between *n* and *m*. Thus far, methods have been proposed to construct *m*-connected networks and to merge two or more networks into a single network for maximum connectivity. Once a network is constructed, and a claim is made that the constructed network is *m*-connected, to verify the claim, the next chapter proposes a method to compute the connectivity of a network in a single iteration.

V. REFERENCES

[1]. A. Frank and E. Tardos, 1989, Vol. 114/115, pp. 329-348. ‘An application of submodular flows’, Linear Algebra and its Applications,

[2]. Fred Halsall, 2000, ‘Multimedia Communications: Applications, Networks, Protocols and Standards’, First Edition, Addison-Wesley.

[3]. Gaoxi Xiao, Yiu-Wing Leung and Kwok-Wak Hung, 2001, pp.1102-1115, ‘Two-stage cut saturation algorithm for designing all-optical networks’, IEEE Communications Society.

[4]. M. Gerla, H. Frank and J. Eckl, 1974, pp. 1074-1085, ‘A cut saturation algorithm for topological design of packet switched communication networks’, Proc. NTC.

[5]. GruiaCalinescu and Peng-Jun Wan 2006, Vol.11, No.2, pp.121-128, ‘Range assignment for biconnectivity and k-edge connectivity in wireless Ad Hoc networks’, Springer Netherlands.

[6]. HananShpungin and Michael Segal, 2005, pp.89-9, ‘k-Fault resistance in wireless Ad-Hoc networks’, 2005 joint workshop on Foundations of mobile computing.

[7]. Harold N. Gabow and Suzanne Gallagher, 2008, pp. 550-559, ‘Iterated rounding algorithms for the smallest k-edge connected spanning subgraph’, Nineteenth annual ACM-SIAM symposium on Discrete algorithms.

[8]. Harold N. Gabow, 2006, Vol.53, No.5, pp.800-844, ‘Using expander graphs to find vertex connectivity‘, Journal of the ACM.

[9]. Heinrich Moser and Bernd Thallner, 2006, pp.35-44, ‘Construction of a fault-tolerant wireless communication topology using distributed agreement’, 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks.

[10]. Henri Koskinen, 2004, Vol. 26, pp. 321-338, ‘A simulation- based method for predicting connectivity in wireless multihop networks’, Telecommunication Systems.

[11]. M. Henzinger, S. Rao and H. Gabow, 2000, Vol. 34, pp 222-250, ‘Computing vertex connectivity, new bounds from old techniques’, Journal of Algorithms.

[12]. Huiqiang Wang, Guosheng Zhao and Jian Wang, 2008, ‘Survivable network system: An immune approach’, Proceedings of the 2008 Int. Conf. on Internet Computing in Science and Engineering ,IEEE Computer Society, pp. 329-331.

[13]. IzhakRubln, 1976, pp. 263-267, ‘On reliable topologies for computer networks’, 2nd International conference on Software Engineering.

