MultiCraft

International Journal of Engineering, Science and Technology
Vol. 3, No. 1, 2011, pp. 41-51

INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND TECHNOLOGY

www.ijest-ng.com

Relieving critical nodes and designing of fault tolerant data networks

Mou Dasgupta*, G.P. Biswas

Department of Computer Science & Engineering, Indian School of Mines, Dhanbad, INDIA
**Corresponding Author: e-mail: elle.est.mou@gmail.com, Tel +91-9308769191, Fax.+ 91-326-2296563*

Abstract

The communication networks having different types of nodes, links and other resources cause uneven distribution of the traffic loads. Also the chances of failure increase with the growth of the network size. To avoid such uneven traffic distribution and failure, and to continue with the usual traffic load, some precautions must be taken. This paper as a remedy, introduces a concept of critical nodes (CN) and the formation of critical rings (CR) around CN to relive the networks from such unequal traffic distribution and failure. In brief, a node is termed as critical node that establishes maximum source-terminal paths and a critical ring is defined as the nearest ring around a CN by considering the nodes directly connected to the CN, which may require inserting one or more links. The merits of this CR are that it can bypass some of the routes of the CN and convert it into non-critical node, and can tolerate fault(s) occurring on or around CN, thus make the network fault-tolerant. The proposed concepts have been applied to a sample network of 11 nodes and above mentioned remedies have been achieved.

Keywords: Data Networks, Fault Tolerance, Critical Nodes, Reliability.

1. Introduction

In today's world, data networks such as Internet, World Wide Web, peer-to-peer network, etc have a very significant role. These networks have drawn a significant amount of attention and interest among researchers (Wang *et al.,* 2008). With the span of time, the ever increasing networks leads to the increase of traffic, this in turn results in the occurrence of fault, congestion etc. The focus of studying these large data networks is to increase the fault tolerance of the network and to control the increasing traffic congestion and thereby to improve the efficiency of data transmission. Thus, the ultimate goal is on the performance improvement of data networks. One major characteristic of improving network performance is to modify the underlying network structure (Wang *et al.*, 2006). But for such modifications the balance between the cost and system efficiency is pivotal. The designing strategies of data network consist of finding the topology at a minimal cost while at the same time maximizing a certain performance criterion (Banerjee *et al.*, 2007). This stage of design can be described as network topological optimization.

Another major aspect is to develop efficient routing schemes. In order to perform efficiently, the routing schemes largely depend on network topology and its traffic characterization. In this regard the critical node problem is treated as a crucial concern. Critical nodes are the nodes whose deletion results in minimum pair wise connectivity among the remaining nodes (Arulselvan *et al.*, 2009). For understanding the network topological characteristics and network connectivity properties, it is important to identify the critical nodes of a network.

The objective of this paper is to present a technique that will convert the critical node into non-critical. As the critical node on a network becomes hotspot, it leads to early network performance degradation. To alleviate early saturation of the critical node area, we propose a concept of critical ring. When employed, the critical node of the network does not remain critical, together with having an alternate path if the critical node becomes faulty. At the same time performance of the network is also increased. One important performance measure of a communication network is reliability. Reliability analysis of networks has been studied for many years (Hwang *et al*., 1981), and numerous algorithms and evaluation techniques have been proposed. The notion of reliability is defined as the probability that every pair of node is connected with every other pair. In this paper the performance of communication network is analyzed in terms of reliability. A typical communication network structure is composed of three levels. Backbone networks, local access networks (LAN), and mixed networks. The backbone networks are dedicated to the delivery of

information from source to destination (end to end) using its switching nodes. On the other hand LANs are typically centralized systems allowing users to access hosts or local servers (Leondes *et al.*, 2002). A large scale network usually is a mixed network. This paper mainly deals with the backbone performance issues.

The paper is organized in the following manner. Section 2 gives a detailed background theory. In section 3 the explanation of the proposed technique is presented. In section 4 the numerical results along with an example is depicted. Section 5 deals with the performance issues related to the proposed method. Finally, conclusion is given in section 6.

2. Detailed Background of the Proposed Work

A data communication network is a system whose components are autonomous computers and other devices that are connected together usually over a certain physical distance. Each computer has its own operating system and there is no direct cooperation between the computers in the execution of programs. A network is characterized by the basic feature that its components are connected by physical links through which information is transmitted in some pattern for communication to occur. Moreover, the efficiency of a network is highly dependent upon how the components are connected within the network. For this reason detecting critical nodes of a network is important to comprehend the topological characteristics and connectivity pattern of the network. It is also helpful in designing strategies for communication breakdowns in data communication networks.

The critical node problem has several applications other than in data communication networks as in the field of social network analysis. Social networks have attracted attention of several researchers in recent years and their study reveal that various properties which are the most common in network depiction of social interactions including cohesion, transitivity, diameter, connectivity etc, are important for better understanding of them. They are responsible for social contagion and provide scope for containment of epidemic breakthrough (Albert *et al.*, 1999). The critical node identification and minimization approach can be used to solve the above mentioned problems.

One simplified algorithm for critical nodes identification is that for every node within a network, a subgraph is obtained by removing this node and all of its adjacent edges. And finally it is tested whether the subgraph formed is connected or not. If it is not connected, the corresponding node is a critical node (Jorgic *et al.*, 2004). As a result, the node having only one neighbour is not critical.

A faster global algorithm for detecting critical nodes was described by Duque-Anton, Bruyaux, and Semal *et al.* (2000), who used depth first search. This algorithm is basically a centralized one but it can also be implemented globally in a distributed manner. While executing DFS (depth first search) on an undirected graph, a node is chosen arbitrarily at the beginning which becomes the root node. Edges are traversed and the nodes that are visited are marked. On the way the nodes are pushed into a stack. This process is repeated until a node is reached which is only connected to nodes already visited. At this point the backtracking process may be carried out up to a vertex which has edges connected to the nodes that have not been yet visited. As a matter of fact, such a node will always be a critical node of the graph.

When a node in a network has been identified as critical there are primarily two consequences associated to it. Sharing its load and in case of a breakdown, resuming with the usual traffic flow, i.e. fault tolerance.

Load balancing is a technique to distribute work among a group of computers, network links, or other resources, in such a manner so as to achieve high resource utilization and maximum throughput. A scheduling and load balancing strategy may be used at a node having multiple outgoing links to distribute its total loads among the connected links. These strategies improve the network efficiency when employed. Multiple tasks are distributed to the nodes uniformly. The goal is to get the optimal result from multi choice field. There is still no algorithm to get the optimal solution. Liu *et al.* (2003) proposed a heuristic self-adapting inherit optimization algorithm to tackle the problem of load balancing on network and multi-objective route optimization with chaos group (Liu *et al.*, 2004). The authors (Xiu *et al.*, 2002) have proposed a dynamic route choice method of traffic engineering based on path measurement, bandwidth and hops. However, all the methods mentioned above do not involve load problem of boundary link of the network.

As the number of nodes in the network increases, the chance of critical node failure also increases. Therefore, the ability to tolerate failure is equally important. Fault tolerance is usually achieved by designing and implementing fault tolerant routing algorithms so that whenever faulty components are encountered, these algorithms can find an alternative path for data transmission (Gu *et al.*, 2007). But this method is popular in mesh and torus networks because of inherent path diversity provided by their topologies. Usually these networks have certain fault tolerant routing algorithms that are based on fault ring (*f*- ring). A faulty region is surrounded by a set of fault free components. An *f*- ring is defined as a ring forming when these components are connected. Occurrence of any fault leads to the formation of *f*- ring and a fault tolerant routing algorithm directs the flow to follow the route on the *f*- ring. The major disadvantage of this method is that it can be implemented to mesh and torus networks only. Also, the *f*- ring usually becomes too congested, and hence the network performance is degraded. Thus, the other way of achieving fault tolerance is through adding extra components to the network.

The technique provided in the next section solves the mentioned problems and achieves fault tolerance for a variety of networks.

3. The Proposed Technique

In this section the proposed method is introduced. The first part provides a method for identifying the critical nodes of the network. Next, an optimal design technique is developed to convert the critical nodes into non-critical.

A. Detecting the critical nodes

This section proposes a method of finding the critical nodes of a network. A critical node can be defined as the node whose deletion or malicious behavior disconnects or significantly degrades the performance of the network. Once identified, a critical node can be the focus of more resource intensive monitoring or other diagnostic measures. In order to detect a critical node a graph theoretic approach of network representation has been considered. In this approach the concerned network is represented as a graph $G(V, E)$, where the vertex set, V corresponds to the set of nodes and the edge set, E corresponds to the set of links of the network. Since the removal of the critical node causes minimization of connectivity among the remaining nodes, hence the critical node can be viewed as the node through which most of the network traffic flows.

Let the path set ρ in G defines a set of paths for all ordered pairs (x, y) of vertices of G. The path $\rho_i(x, y)$ specifies the path that carries data transmitted from the source x to the destination y, where $x, y = 1, 2 \ldots V$ and $x \neq y$. i is suffixed to ρ for providing the path number from x to y. Out of ρ, ρ_x is the summation of paths from x to all y. That is,

$$\rho_x = \sum_{y=1,\, y\neq x}^{V} \rho_i(x, y) \tag{1}$$

Also,

$$\sum_{x=1}^{V} \rho_x = \rho \tag{2}$$

Let $\tau_x(G, \rho)$ be the number of paths specified by ρ going through x. The parameter $\alpha(\rho)$ defines one half of the sum of paths in ρ. That is,

$$\alpha(\rho) = \frac{1}{2} \sum_{x=1}^{V} \rho_x \tag{3}$$

For every $\tau_x(G, \rho)$ which is greater than $\alpha(\rho)$ is included in a set called critical set, C_S. Finally, the critical node does have

$$\tau_x{}^{C}(G, \rho) = max\{ \tau_x(G, \rho) : \tau_x(G, \rho) \in C_S\} \tag{4}$$

The corresponding node, x of $\tau_x{}^{C}(G, \rho)$ is the critical node of the network G.

The algorithm is given below.

Algorithm 1

Input:
Network Topology, Number of Nodes-n, Number of Links-m

```
Critical_Node()
{
        ρ ← 0
        for i = 1 to n do
           k ← 1
           P[i] ← 0
           for j = 1 to n, j≠ i do
                  P_ij[k] ← paths from i to j
                  k ← k + 1
                  P[i] ← P[i] + P_ij[k]
           end for
           ρ ← ρ + P[i]
        end for
        A_ρ ← ρ/2
        for i = 1 to n do
```

```
        T_i ← paths going through i
        if ( T_i > A_ρ ) then
                C[i] ← T_i
        else
                C[i] ← Null
    end for
    for i = 1 to n do
        C_max ← max{ C[i] }
        C_x ← i
    end for
}
```

Output:
Critical node of the network, C_x

B. Formation of critical ring and converting critical node into non-critical

Once the critical node of the network is being found, the objective is to make it non critical. This is done in two phases. At the first phase it is tested if the critical node does have a critical ring. The critical ring of a critical node is defined as the nearest loop around the critical node considering the nodes that are directly connected to the critical node. The intuition behind having the critical ring is that the load of the critical node will be shared, thereby relieving the critical node. The traffic will traverse on that ring bypassing the critical node that previously used to traverse through the critical node. Second, if for any reason the critical node becomes non operational, then many source-destination pairs loose their connectivity. But if a critical ring exists, then irrespective of the critical node all other node pairs will be connected via alternative paths of the critical ring. Another advantage of having a critical ring is in case of resource shortage. With critical ring employed, the network traffic becomes more balanced and congestions are alleviated while average latency will be shortened. If any critical ring is not found then the second phase commences. In this stage a critical ring is created by inserting the missing links of the ring iteratively.

Theorem 1*: For a network G if a critical node exists, then it will not have a critical ring.*

Proof: Let the path set in G be ρ and the total number of paths in ρ be n. Let x^C be the critical node.

According to the definition of the critical node from equation *(4)*,

$$\tau_x^C (G, \rho) > n/2 \tag{5}$$

Thus, more than *(n/2)* paths have x^C as one of their intermediate nodes.

Let, for all other nodes in G

$$\tau_{x \neq x}{}^C (G, \rho) \leq n/2 \text{ for all } x \in V \tag{6}$$

Assuming that x^C has a critical ring. Then the paths which were previously going through x^C, will now have an alternate way without x^C, as the nodes on the critical ring are connected among themselves.

Thus, for some other node, say z, we will have

$$\tau_z (G, \rho) > n/2 \tag{7}$$

Inequality (7) contradicts with inequality (6). Hence, it follows that a critical node cannot have a critical ring.

■

The algorithm for creating critical ring is given below.

Algorithm 2

Input:
Critical Node-C_x

```
Critical_Ring()
{
        D_n ← directly connected nodes of C_x
```

```
        for i = 1 to n do
            if ( a link exists between Di and Di+1 ) then
                    CRi ← 1
            else
                    CRi ← 0
        end for
        call Find_Critical_Ring(CR, n)
        if (Find_Critical_Ring = = false ) then
        call Make_Critical_Ring(D, CR, n)
}
```

Output:
Critical ring of C_x

At first the Find_Critical_Ring() function is triggered. This function finds a ring that is formed among the directly connected nodes of the critical node. When invoked the Find_Critical_Ring() function returns true if such ring is found. Otherwise the Make_Critical_Ring() function commences. Given below are the two above mentioned functions.

```
Find_Critical_Ring(CR, n)
{
        S ← 0
        for i = 1 to n do
            if (CRi = = 1) then
                    S ← S + 1
        end for
        if ( S = = n) then
            R ← true
        else
            R ← false
        return R
}


Make_Critical_Ring(D, CR, n)
{
        for i = 1 to n do
            if (CRi = = 1) then
                    continue
            else
                    add a link from Di to Di+1
                    CRi ← 1
        end for
}
```

This insertion of links relieves the critical node from being critical. It has been observed that all of the links are not required. A few link additions improve the network performance significantly. A detail evaluation is provided in section V. Now, additions of links make some other node to be critical. Hence, a termination condition of the algorithm has been defined so that after certain iteration the network topology is balanced and no critical node exists. For this, the dispersion of number of paths passing through each node in the critical set is measured after relieving a critical node using standard deviation. After certain iteration when the standard deviations of two adjacent iterations do not vary significantly, then the algorithm stops.

4. Illustrative Example and Numerical Results

The algorithms in the previous section are coded in FORTRAN 77 and the simulation is done on a Xeon processor, 3.4 GHz PC with 4 GB RAM. In order to test the performance of the method, a series of computational experiments were performed using these algorithms. Consider a network with 11 nodes and 19 links, shown in Figure 1 as an example. The network is presented by the authors in (Belovich *et al.*, 1995). This network has been considered for easy comparison, although the proposed technique can be implemented in any network.

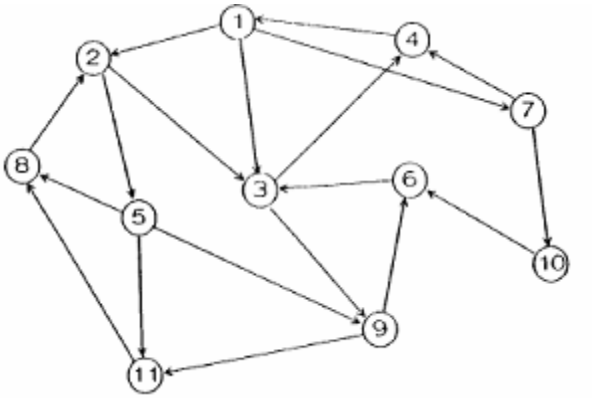

Figure 1. The topology structure of 11 nodes Network

To detect the critical node of the network in Figure 1, all the paths from every source to destination pairs are calculated. For this at first ρ_x is calculated. Let us consider any arbitrary node of the network, say node *5*. Therefore,

$$\rho_5 = \sum_{y=1,\, y\neq 5}^{11} \rho_i(5, y) \tag{8}$$

i.e., $$\rho_5 = \rho_i(5, 1) + \rho_i(5, 2) + \ldots + \rho_i(5, 11) \tag{9}$$

where, *i* indicates the path number of the node pairs, as for a node pair there may be more than one path between them. The number of paths for all pairs of node *5* is given in Table 1.

Table 1. Number of paths of node 5

Node-pair	Paths
5-1	4
5-2	4
5-3	4
5-4	4
5-6	6
5-7	4
5-8	3
5-9	3
5-10	4
5-11	3

In a similar manner all paths of each and every node of the network can be found. Next, from equation *(2)* we have,

$$\rho = \sum_{x=1}^{11} \rho_x$$

i.e., $$\rho = \rho_1 + \rho_2 + \rho_3 + \ldots + \rho_{11} \tag{10}$$

i.e., $$\rho = 287 \tag{11}$$

From equation *(3)* we have,

$$\alpha(\rho) = 144 \tag{12}$$

Now, the paths passing through each node is calculated and shown in Table 2.

Table 2. Number of paths passing through each node

Nodes	No. of Paths
Node 1	125
Node 2	151
Node 3	167
Node 4	115
Node 5	81
Node 6	80
Node 7	40
Node 8	69
Node 9	117
Node 10	40
Node 11	75

According to equation *(12),* the number of paths that pass through nodes *2* and *3,* i.e., *151* and *167* are included in C_S. Hence, the node corresponding to the maximum number of paths going through a node is node *3*, which is the critical node. Since the critical node of the concerned network is being found out to be node *3*, the next step is to check whether the critical node has a critical ring or not. From Figure 1, it is clear that node *3* is directly connected to nodes *1, 2, 4, 6,* and *9.* These nodes are not directly connected to themselves. Hence, node *3* does not have any critical ring. According to algorithm *2*, links are added so as to make a critical ring surrounding node *3*. Figure *2* shows the addition of links from node *2* to *9* and from node *6* to *4*.

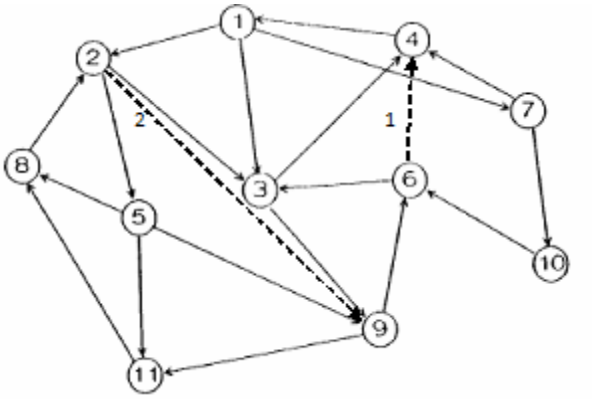

Figure 2 The critical ring of node 3

As the critical ring is being formed, node *3* does not remain the critical node. After insertion of link *(6-4)* of the critical ring, the total number of paths of the network becomes

$$\rho = 416 \tag{13}$$

Therefore,

$$\alpha(\rho) = 208 \tag{14}$$

Hence, C_S should include only those paths passing through each node of the network that are greater than $\alpha(\rho)$. Table 3 depicts the number of paths that include the network nodes in them after inserting link *(6-4)* of the critical ring.

Table 3. Number of paths passing through each node

Nodes	No. of Paths
Node 1	223
Node 2	225
Node 3	212
Node 4	226
Node 5	118
Node 6	187
Node 7	52
Node 8	91
Node 9	190
Node 10	58
Node 11	99

Clearly, the number of paths passing through nodes *1, 2, 3,* and *4* are included in the set C_S. Out of these, the maximum number of paths go through node *4*. Hence, node *4* is the new critical node instead of node *3*. The standard deviation of the critical set C_S is calculated in both cases. When node *3* is critical the standard deviation is *8* and when node *4* is critical the standard deviation is *5.59*. That is, after the first iteration the number of paths going through each node is more evenly distributed. Thus, it can be said that the overhead which was previously associated with node *3* was more, than that which is now associated with a new node, i.e. node *4*. This process is continued until the variations in standard deviations are reduced.

5. Performance Analysis

To test how the algorithms explained in section *2* respond to the issue of network performance, the network with *11* nodes shown in figure *1* is considered as an example. In this paper reliability is taken as the parameter to measure the network performance. The reliability approach considered here is that of source-to-terminal reliability, which is applied to all the nodes of the network. Figure *3* shows the performance of all the nodes of the original network as in figure *1*. The traffic considered is uniform for all cases. A complete reference for estimating the node reliabilities of a network is given by Belovich *et al*. (1995).

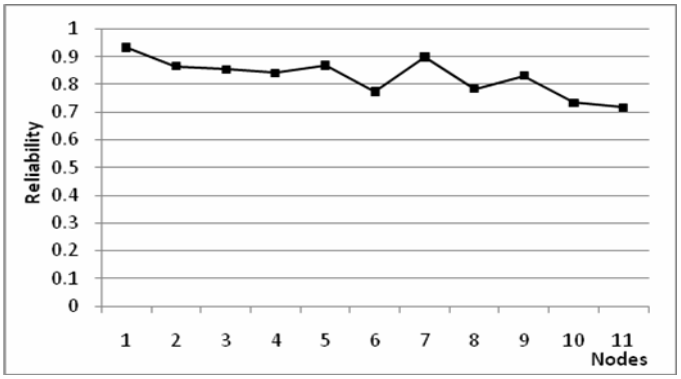

Figure 3 Performance of the original network

The original network's critical node is node *3*. Hence, this node has the maximum overhead than any other node of the network. Now, to evaluate how the network performs it is necessary to consider that the critical node is faulty. The performance of the network when the critical node is faulty is shown in figure *4*.

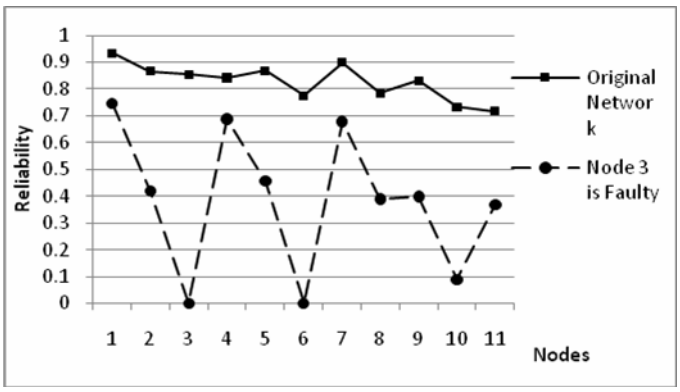

Figure 4 Performance of the network when node 3 is faulty

From figure *4* it is evident that the network performance abruptly degrades if the critical node is removed from it. Also, the critical node has the most probable chance of being faulty in a practical network environment. Our target is to improve the performance of the network by making the critical node non-critical. According to the proposed technique, when the critical ring is formed around the critical node, the load on critical node is shared by the nodes of the critical ring. In figure *5* the performance of the network after the critical ring being formed is shown where the critical node is considered as faulty.

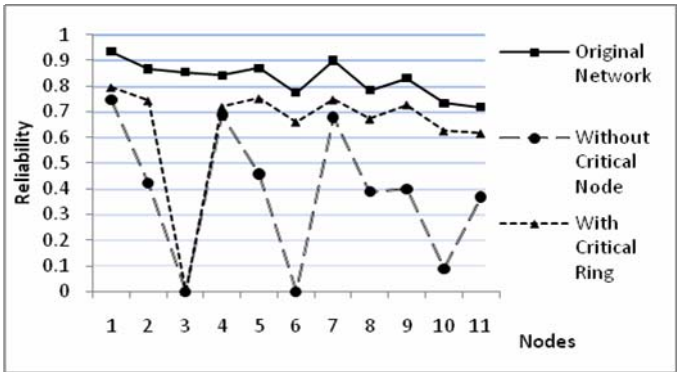

Figure 5 Performance of the network with the critical ring

Thus, by merely adding two links the efficiency of the network is increased noticeably. Now we analyze the effect of adding each link on the network performance individually. Figure *6* depicts all the cases including the addition of links *(2-9)* and *(6-4)*. Out of these two links of the critical ring, the addition of link *(2-9)* has very little effect on performance whereas the effect of adding link *(6-4)* is almost same as that of the results with the critical ring. Thus, link *(6-4)* is very crucial with respect to the network. A new dimension which can be implied is that by inserting only one link i.e. link *(6-4)* to the original network topology the performance of the network is improved much significantly. This is depicted in figure *7*. Thus, the proposed technique can be viewed as network topological optimization technique.

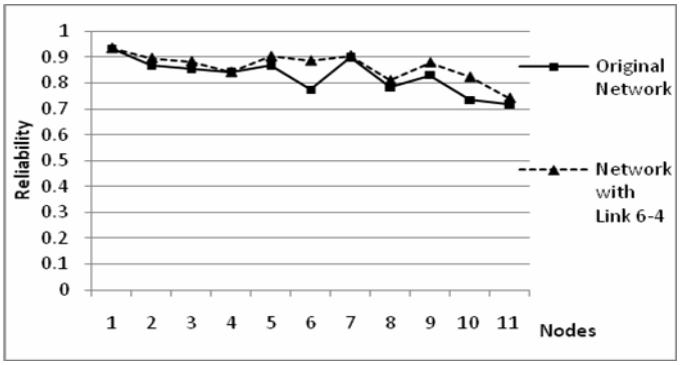

Figure 7 Performance of the Network with Link 6-4.

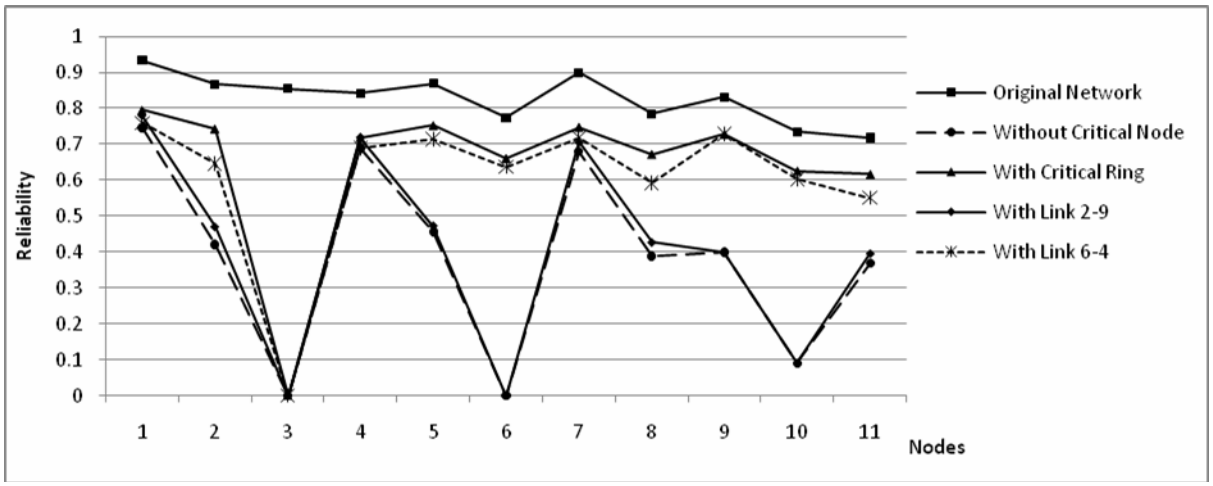

Figure 6 Performance of the Network in Various Cases.

6. Conclusions

This paper introduces the concepts of critical nodes and critical rings in communication networks. Algorithms to locate and form critical nodes and critical rings are proposed respectively. It has been shown that critical ring produces a number of alternate source-terminal paths thus capable of bypassing significant amount of traffic of critical nodes. As a consequence, it also tolerates faults if they occur on or around critical nodes as well. These achievements have been established by applying the proposed scheme to a sample data networks.

References

Albert R. and Barab´asi A.L., 1999, Emergence of scaling in random networks. *Science*, Vol. 286, pp. 509–512.

Alouane A. B. and Bean J. C., 1997, A Generic algorithm for the multiple-choice integer program, *Operations Research*, Vol. 45, No. 1, pp. 92-101.

Arulselvan A., 2009, Detecting critical nodes in sparse graphs, *Computers and Operations Research*, Vol. 36, No. 7, pp.21932200.

Banerjee N., and Kumar R., 2007, Multiobjective network design for realistic traffic models, DOI: 10.1145/1276958.1277341.

Belovich S.G., 1995, A Design Technique for Reliable Networks under a Non-Uniform Traffic Distribution, *IEEE Transaction on Reliability*, Vol. 44, No. 3, pp 377-386.

Boppana R.V., Chalasani S., 1995, Fault-tolerant wormhole routing algorithms for mesh networks, *IEEE Trans. Comput*ing, Vol. 44, No.7, pp. 848–864.

Duque-Anton M., Bruyaux F.,. Semal P, 2000, Measuring the survivability of a network: connectivity and rest-connectivity, *European Transaction of Telecommunications*, Vol. 11, No. 2, 149-159.

Gu H., Zhang J., Wang K., Liu Z. and Kang G., 2007, Enhanced fault tolerant routing algorithms using a concept of balanced ring, *Journal of System Architecture*, Vol. 53, pp. 902-912.

Gu H., Zhang J., Wang K., Liu Z., and Kang G., 2007, Enhanced fault tolerant routing algorithms using a concept of "balanced ring",*Journal of Systems Architecture*, Vol. 53, Issue 12, pp. 902-912.

Hwang C.L., Tillman F.A. and Lee M.H., 1981, System reliability evaluation techniques for complex/large systems – A review, *IEEE Trans. Reliability*, Vol. R-30, pp.416-423.

Hwang C.L., Tillman F.A., and Lee M.H., 1981, System reliability evaluation techniques for complex/large systems – A review, *IEEE Trans. Reliability*, Vol. R-30, pp.416-423.

Jorgic M., Stojmenovic I., Hauspie M., Simplot-Ry1 D., 2004, Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks, *Proceedings of 3rd IFIP Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET 2004).*

Leondes C.T., 2002, Database and Data Communication Network Systems: Techniques and Applications, Vol 1, Chapter-9.

Liu H., 2004, Internet routing optimization control algorithm with multiple objectives, *ACTA ELECTRONICA SINICA*, Vol. 32, No. 2, pp. 306-308.

Liu H., Bai D., and Ding W., 2003, A heuristic adaptive genetic algorithm for load balancing in MPLS networks, *Journal of China Institute Communications*, Vol. 24, No. 10, pp. 39-45.

Qin Y. and Zhao C., 2006, Efficient Split Scheduling Scheme for Traffic Balancing on Multi-link Networks, *Proceedings of IEEE IMSCCS*, pp. 180-185.

Tarjan R., 1972, Depth first search and linear graph algorithms, *SIAM J. Computing*, Vol. 1, No. 2, pp. 146-160.

Wang B.H., 2008, Routing strategies in traffic network and phase transition in network traffic flow, *PRAMANA - journal of physics*, Vol. 71, No. 2, pp. 353-358.

Wang W.X., 2006, Integrating local static and dynamic information for routing traffic, DOI: 10.1103/PhysRevE.74.016101.

Xiu X., Sun Y., and Liu Z., 2002, Traffic engineering dynamic routing based on bandwidth and hops, *ACTA ELECTRONICA SINICA*, Vol. 30, No. 2, pp. 274-278.

Biographical notes

Mou Dasgupta has obtained B.Sc. degree in Economics (honours) from Calcutta University in 2004 and Master of Computer Application from West Bengal University of Technology in 2007. Currently she is a senior research fellow in the Department of Computer Science and Engineering of Indian School of Mines, Dhanbad, and pursuing Ph.D. in Computer Science in the same university. Her main research interests include topology design of data communication networks and multi-objective optimization.

Dr. G.P. Biswas is a Professor in the Department of Computer Science & Engg., Indian school of Mines (ISM), Dhanbad. Before joining ISM, he was engaged in a research project on the "Designing of CA/PCA based BIST structure for VLSI (soft/hard) cores", sponsored by M/s Fujitsu Microelectronics, USA, with the Dept. of Comp. Sc. & Tech., BE College, Sibpur, Howrah, WB. He has around 14 years of teaching and research experience, where he teaches a number of UG/PG papers including Theory of Computation, Computer Org/Arch., Computer Networks, VLSI Designs, Cryptography and Network Security etc., and published around 50 research papers in Journals and Conference/Seminar Proceedings. Dr. Biswas has experience of guiding B. Tech (CSE), M. Tech (CA) and PhD students. His research interest includes Cryptography, Network security, Cellular Automata (CA), VLSI designs, Computer Networks and Data communication.

Received August 2010
Accepted March 2011
Final acceptance in revised form March 2011