International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-2 Issue-4, September 2012

Identifying Faulty Node and Alternative Path in a Network

Ramander Singh, Vinod Kumar, Ajay Kumar Singh, Santosh Kumar Upadhyay

Abstract: ***This Paper describes a good algorithm to find the Faulty node in any given complex Network and provides the alternative path and also tells us the number of faulty node. As the Technological advances increasing number of node day by day in a Network. If a node fails, the system continues to operate with degraded performance until the faulty node is repaired. If the repair operation will take an unacceptable amount of time, it is useful to replace the faulty node with a spare node. However, the appropriate procedures must be followed and precautions must be taken so you do not interrupt I/O operations and compromise the integrity of your data that we have presented in our paper.***

Index Terms- Alternative path, Faulty node, Buffer, Computer

I. INTRODUCTION

Today scenario a computer network is divided into mainly two parts first one is wired computer network and wireless computer network [1]. Wired computer network are those network in which all computer are connected to each other via wire or common wire and wire-less computer network are those in which all computer are connected to each other without a wire. It means that the network may be of any type of the computers or any computer embedded device connected to each other by a LAN cable.

If any one computer or client wants to communicate to each other that's related information in another computer, the information is accessed from that computer by which client is communicating, In case if it is not directly communicate to that computer(in wire-less network). Same case in wired computer network if any node is failed during the communication then the integrity of data communication [2] is lost. So to overcome this type of problem there is a need of alternative path to communicate to that node.

As we know that failed nodes may be decreases the quality of service [3] of the entire network. It is necessary to study the faulty node detection methods in entire network for the following reasons:

1) Enormous low-cost nodes are generally deployed in Uncontrollable and unfriendly environments [4]. Therefore failure in nodes can occur more easily in comparison to other systems.
2) The applications of networks are being widened. So fault detection for nodes having great importance.
3) It is inconvenience and not practically possible to manually examine the nodes are functioning properly or not.
4) As we know that the correct information unobtainable by the control centre because failed nodes may produce incorrect data.

The node status in entire network can also be categorized into two types: normal and faulty. Faulty node can be "permanent" or "static" [5]. Here "permanent" means failed nodes will remain faulty until and unless they are replaced as shown in figure 1, and here "static" means the faulty node can be repaired. So our work is regarded to static node and provides the alternative path to maintain the data integrity.

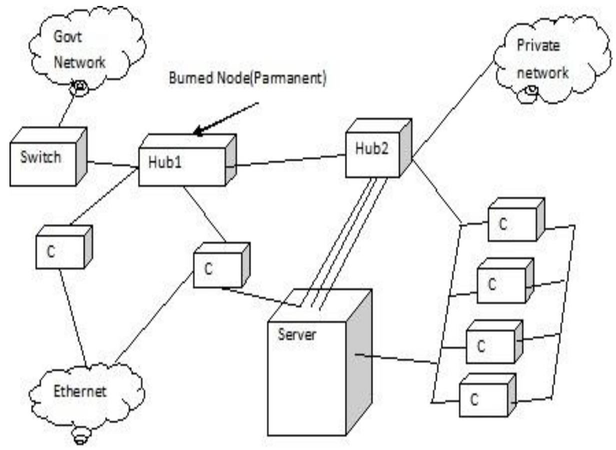


Fig. 1 Permanent and Static Node.

Proposed System:

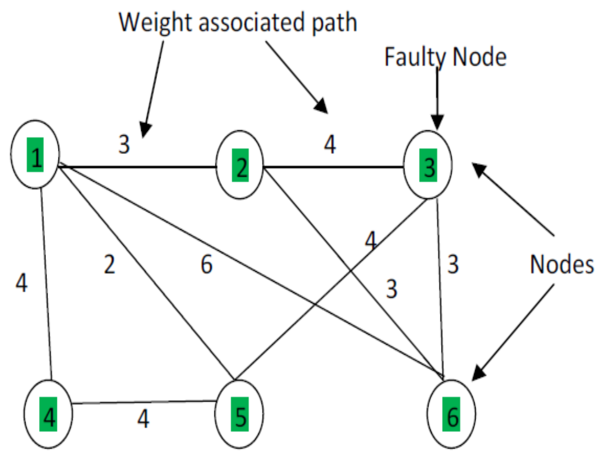


Fig. 2 A computer Network.

Figure 2. Shows the Network of computer in which Number of nodes are connected to each other directly and indirectly to each other. The Adjacency Matrix [6] of given Computer Network:

Manuscript received September 02, 2012.

Ramander Singh Computer Science and Engineering, Mewar University, Rajasthan, India.

Vinod Kumar Computer Science and Engineering, Mewar University, Rajasthan, India.

Ajay Kumar Singh, Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut, India.

Santosh Kumar Upadhyay Computer Science and Engineering, Mewar University, Rajasthan, India.

Nodes

$$\begin{array}{c|c} & 1\ 2\ 3\ 4\ 5\ 6 \\ \begin{array}{c}1\\2\\3\\4\\5\\6\end{array} & \begin{pmatrix} 0&3&0&4&2&6\\ 3&0&4&0&0&3\\ 0&4&0&0&4&3\\ 4&0&0&0&4&0\\ 2&0&4&4&0&0\\ 6&3&3&0&0&0 \end{pmatrix}\end{array} \qquad \begin{array}{c|c} & 1\ 2\ 3\ 4\ 5\ 6 \\ \begin{array}{c}1\\2\\3\\4\\5\\6\end{array} & \begin{pmatrix} 0&3&0&4&2&6\\ 3&0&0&0&0&3\\ 0&0&0&0&0&0\\ 4&0&0&0&4&0\\ 2&0&0&4&0&0\\ 6&3&0&0&0&0 \end{pmatrix}\end{array}$$

Fig. 3 (a) Old Matrix (b) New Matrix

Proposed algorithm:

1. 1st we determine the old matrix for a given node in the network [7] before passing the message.
2. Now we determine the new matrix for a given node in the network after passing the message.
3. Now compute the sum of rows and column of both matrix and store into some buffer for both matrixes separately.
4. Now we compare these sum either row wise or column wise of both matrix.
5. If in step 4 the sum is same then there is no fault and go to step 11 otherwise go to next step.
6. Now we check the row wise or column wise sum of new matrix which contain 0.
7. Loop
8. from i = 1 to size of the row or column
9. If A[i] == 0
10. Then ith will be faulty node.
11. End

Floyd Wars hall's Algorithm:

The Floyd–Warshall algorithm [8] is mainly graph analyzing algorithm for finding shortest roots in a weighted graph and it also find the transitive closure of a given relation R. After the execution, the algorithm will compute the summed weights of the shortest roots between all pairs of vertices, but it will not return the details of the roots themselves. This algorithm is a concept of dynamic programming. The Floyd–Warshall algorithm analyzes all possible roots through the network graph for each pair of vertices. This algorithm will do this analysis with only $\Theta(|V|^3)$ comparisons in a network graph. This is remarkable considering that there may be up to $\Omega(|V|^2)$ edges in the given graph, and all combination of edges is checked. It does calculate an estimate on the shortest root between two vertices till the estimate is optimal.Here W (i, j) shows the weight of the edges between vertices *i* and *j*, shortest root (i, j, L) can be define by the following recursive formula: the terminating case is Shortest root (i, j, 0) =w (i, j) and the recursive case is

Shortestroot (i,j,L)=min(Shortestroot (i,j,L-1),shortestroot (I,L,L-1)+Shortestroot (L,j,L-1))

This formula is the heart of the Floyd–Warshall algorithm. The algorithm works by first computing shortestroot (*i*, *j*, d) for all (*i*, *j*) pairs for d = 1, then d = 2, etc. This process continues until d = *n*, and we have found the shortest root for all (*i*, *j*) pairs using any intermediate vertices.

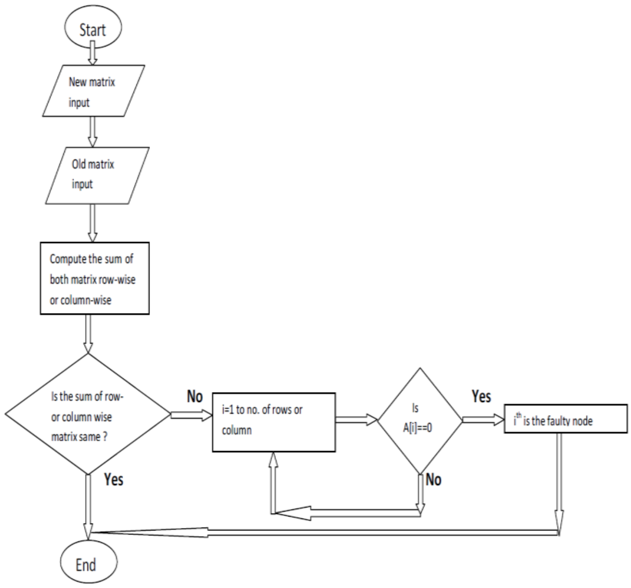


Fig. 4 Flow chart for faulty node

1. /* Here we are assuming that a function Edge_cost (i,j) that return the cost of edge from i to j */
2. /* Here we are assuming that n will be the number of vertex in the network and Edge_cost (i,i) = 0 for self */
3. int root[][]; /* A two-dimensional matrix. At each step in the algorithm, root[i][j] is the shortest root from i to j by using the intermediate vertices (1..L−1). Each root[i][j] is going to initialized in the algorithm with Edge_cost(i,j)value.*/
4. Function FLOYDWARSHALL ()

   for L = 1 to n

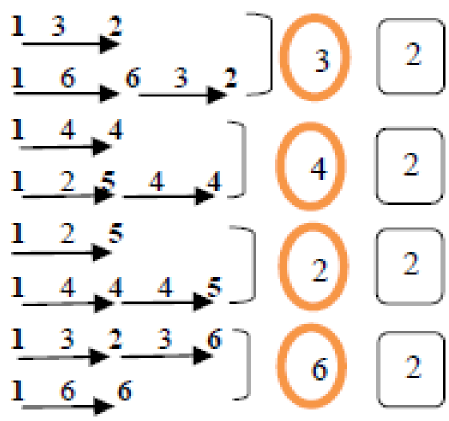
   for i = 1 to n

   for j = 1 to n

   root [i][j] = min (root [i][j], root [i][d]+ root [d][j]);

The shortest and Alternative path for all source and Destination:

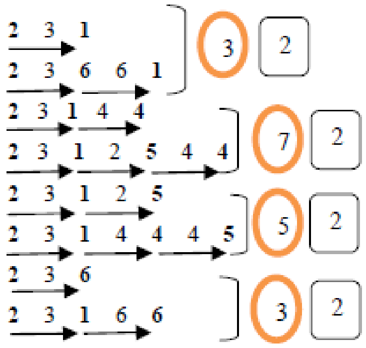
For node: 1



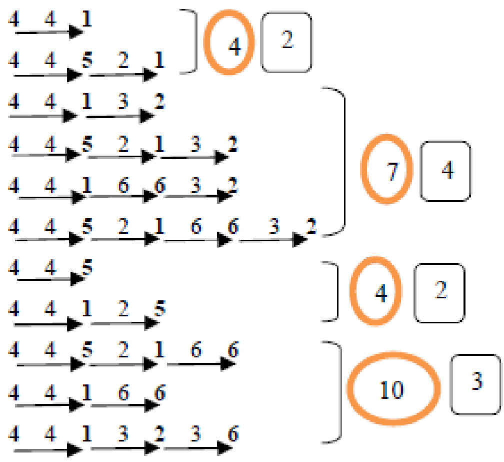
Denotes shortest path from source to destination

Number of alternative path

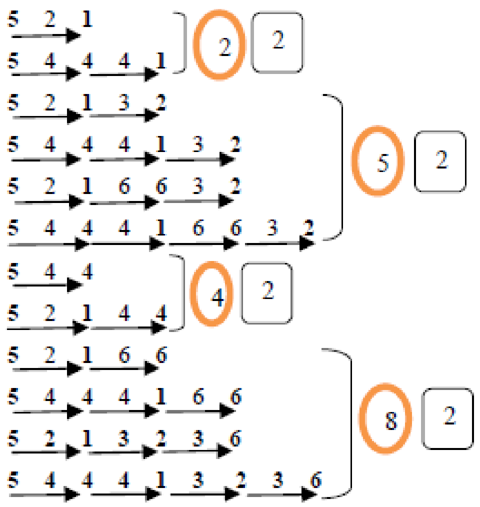
For node: 2



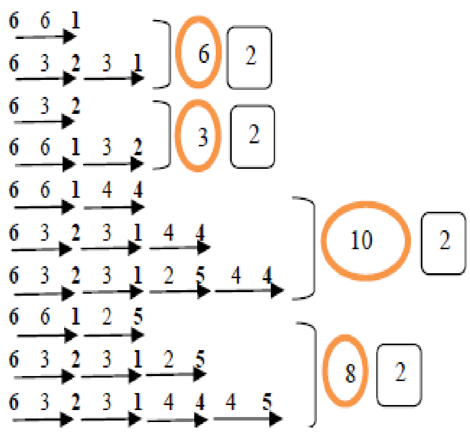
For Node: 4



For node: 5



For node: 6



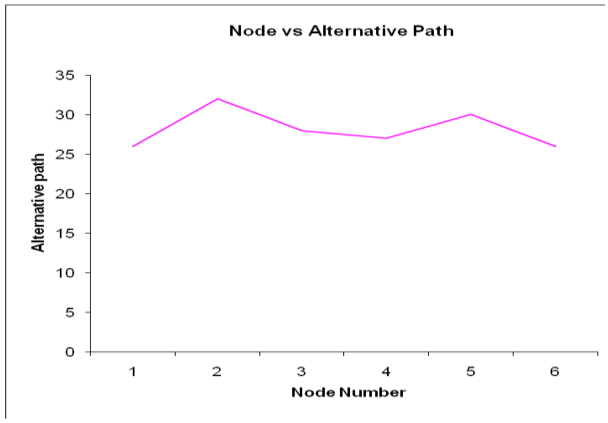
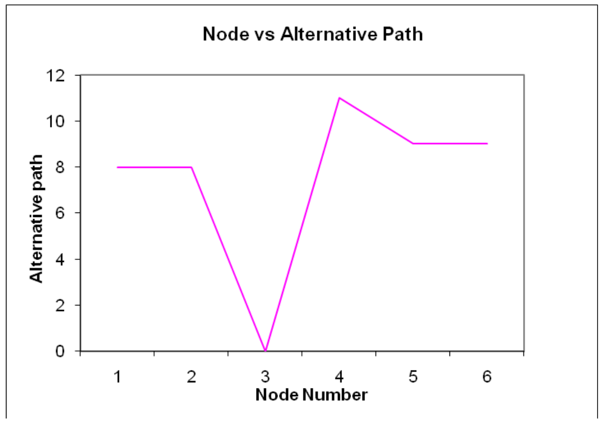


Fig. 5 (a) Alternative Path Vs Node Number without Third as Faulty Node.



(b) Alternative Path Vs Node Number with Third as Faulty Node.

Figure 5(a) describes the alternative path when there is no faulty node in the given network. We are not describing all procedure or solution to find the alternative path [9] between all source and destination because of the limitation of number of pages of this paper.

Although we have given the all possible paths between all source and six as destination node, when third node is a faulty node as shown in figure 5(b).

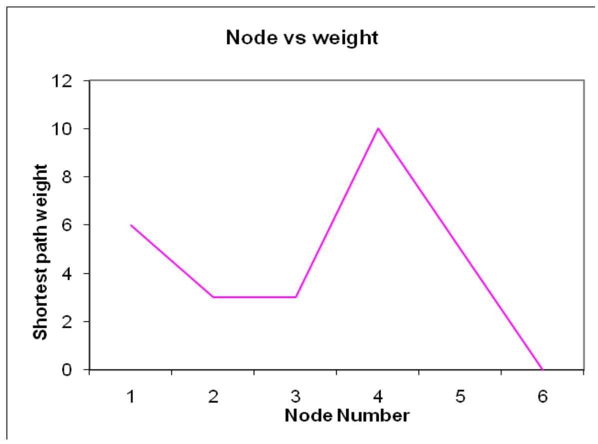
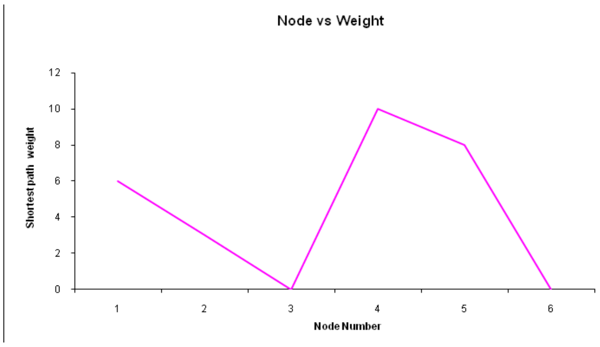


Fig. 6 (a) Shortest Path between All Sources and Six as

Destination Node without Faulty Node.



(b) Shortest Path between All Sources and Six as Destination Node with Faulty Node.

Fig 6 (a) and (b) shows the shortest path with or without faulty node respectively in the given network as shown in figure 2. Due to faulty node in a network as shown in figure 2 the network may hang [10] if there is no any alternative path available as well as the packet or data may choose longer path to reach their destination. Due to this we are providing an alternative shortest path for sending a packet or data to reach their destination. Therefore the network hanging problem is get reduced.

Conclusion: Whenever a fault occurs in the Computer Network, the network may hang and there is a delay or failure of packet or data sent their destination. We proposed the solution to over come this problem by providing the alternative shortest path. The further enhancement can be done by providing security to send the information on alternative shortest path for fastest communication on available alternative path. We can do automation also of this proposed model.

REFERENCES

1. Boncheol Gu, Jinman Jung, Kyongdong Kim, Junyoung Heo, Namhoon Park,Gwangil Jeon, and Yookun Cho., "SWICOM: An SDR-Based Wireless Communication Gateway for Vehicles," vol. 59, no. 4, pp. 1593-1605, May 2010.
2. Yan Li, Jianping Wang, Chunming Qiao, Ashwin Gumaste, Yun Xu, and Yinlong Xu, "Integrated Fiber-Wireless (FiWi) Access Networks Supporting Inter-ONU Communications," vol. 28, no. 5, pp. 714-724 , March 1, 2010.
3. Howard Bowman, Lynne Blair, Gordon S. Blair and Amanda G. Chetwynd, "A formal description technique supporting expression of quality of service and media synchronization, " Lecture Notes in Computer Science, vol. 882, Multimedia Transport and Teleservices, pp. 145-167, 1994.
4. L. Cappietti and P. Aminti, "Rehabilitation of Highly Protected Beaches by Using Environment-Friendly Structures," NATO Science Series: IV: Earth and Environmental Sciences, vol. 53, Environmentally Friendly Coastal Protection, Chapter 2, pp. 163-175, 2006.
5. Julia Poncela Casasnovas, "The Prisoner's Dilemma on Static Complex Networks," Springer Theses, Evolutionary Games in Complex Topologies, Part 1, pp. 51-76, 2012.
6. A. A. Kozlov, V. A. Nosov and A. E. Pankratiev, "Matrices and graphs of essential dependence of proper families of functions," Journal of Mathematical Sciences, vol. 163, no. 5, pp. 534-542., 2009.
7. Xudong Wang, Ping Yi., "Security Framework for Wireless Communications in Smart Distribution Grid," vol. 2, no. 4, pp. 809-818, Dec 2011.
8. Yv Mei Liao and Jie Zhong, "FLOYD Algorithm Based on the Shortest Path in GIS," Communications in Computer and Information Science, Information and Business Intelligence, Part 4, pp. 574-579.
9. Barry J. Welch, "Aluminum production paths in the new millennium," Journal of the Minerals (JOM), Metals and Materials Society, vol. 51, no 5, pp. 24-28, 1999.
10. Travis Russell, "Signaling System" Publisher: McGraw-Hill, Fifth Edition, ISBN: 9780071468794, 2006.

AUTHORS PROFILE

Ramander Singh: Born in 1982 in Bijnor (Uttarpradesh). Phone Number +919557273811 & E- Mail Id: ramendera@gmail.com. He had done B.E (Computer Science & engineering) from Institute of Engineering & Technology Khandari Campus Agra. M.Tech (Computer Science & Engineering) from Mewar University Chittorgarh (Rajasthan).He has 4 year Teaching Experience from Number of Engineering Institute before his M.Tech. He published one Paper in International Conference (ICIAICT). He is IBM DB2 Certified. The major field of Study Area is Algorithm, Programming and Computer Architecture.

Vinod Kumar: Born in November 1984 in Gaya (Bihar). Phone number: +919368269427 & E-mail id: **vinod4r@gmail.com**. He had done B.E (Computer Science & Engg.) from Bhagalpur college of Engineering (T.M.B. University). M.Tech (Computer Science & Engineering) From Mewar University Chittorgarh (Rajasthan).He has 3 year Teaching Experience from Various Engineering Institute before his M.Tech . The major field of Study Area is Automaton Theory, Compiler Construction,Algorithm, Programming and Operating System.

Dr. Ajay Kumar Singh: Born in 1974 at Dhanbad (Jharkhand). He had done B.E (Computer Science & Engg.) from Kumaon Engineering College, M. Tech (I.T) Allahabad, Ph. D (Computer Science & Engg.) Jaypee University of Information Technology. Work Experience: He had been in different institution / university like Radha Govind Engineering College, Meerut, (U.P), Sir Padampat Singhania University, Bhatewar, Udaipur, Rajasthan, Jaypee University of Information Technology, Waknaghat, Solan (H.P), Mody College of Engineering and Technology, Lakshmangarh, Sikar, Rajasthan, Regional Engineering College (Now N.I.T.) Kurukshetra (Haryana), Software Solution Integrated Ltd. (Delhi), Computer Centre CMC. Now he is working with MIET, Meerut, U.P. He has published 6 papers in international Journals like PIER, Asia Magazine EFY Elsevier, JSIP, 8 papers in international Conference out of which 4 of them in IEEE, Published 4 papers in National Conferences, 1 in EFY and presented his papers at Bangalore, Pune, IT B.H.U, USA (Washington DC)

Santosh Kumar Upadhyay: Birth Place & Date - Azamgarh (U.P.), INDIA on 05/08/1982. Master of Technology from Tezpur University (A central University), Tezpur (Assam), INDIA. He is silver medalist in Master of Technology in Tezpur University. The major fields of study is network security, data mining,Algorithm. He has more than seven years experience in teaching and research as Assistent Professor. He is working at Galgotias College of Engineering & Technology, Greater Noida (U.P.), INDIA. He has published many papers in international Journals and Conference