DIRECTED ACYCLIC GRAPH BASED RELIABILITY COMPUTATION OF A NETWORK WITH IMPERFECT NODES

Manoj Singhal

Department of Information Technology, AIMT, Gr. Noida (U.P.), India
manoj_singhal@rediffmail.com, manojsinghal1010@gmail.com

ABSTRACT

In this paper, a heuristic is proposed to find out the reliability of a directed network by using directed acyclic graph. This directed network has imperfect nodes as well as imperfect links. Directed acyclic graph based reliability computation involves three main steps: In the first step, built the reliability function of the given directed network which is the union of all min-paths from source to sink. In the second step, apply a heuristic approach to order the given communication links and nodes of the given directed network. Finally apply Shannon's decomposition method to compute the reliability of the given directed network. The paper also shows that the reliability obtained by this method is equal to the reliability obtained by applying the classical inclusion-exclusion method on the given directed network.

KEYWORDS: *Binary Decision Diagrams (BDD), Directed Acyclic Graph (DAG), Computer communication Network (CNN), Modified Binary Decision Diagram (MBDD), Ordered Binary Decision Diagram (OBDD), Dual Binary Decision Diagram (DBDD).*

I. INTRODUCTION

Network reliability analysis receives considerable attention for the design, validation, and maintenance of many real world systems, such as computer, communication, or power networks. The components of a network are subject to random failures, as more and more enterprises become dependent upon computer communication network (CCN) or networked computing applications. Failure of a single component may directly affect the functioning of a network. So the probability of each component of a CCN is a crucial consideration while considering the reliability of a network. Hence the reliability consideration is an important factor in CCN [1]. The IEEE 90 standard defines the reliability as "the ability of a system or component to perform its required functions under stated conditions for a specified period of time." Many algorithms have been presented to solve the problem of network reliability. These algorithms are based on the exact methods as well as the approximation methods [2, 3, 4]. Some of them are based on the min-paths/min-cuts methods. In these methods we first enumerate all the min-paths and min-cuts of the given CCN, then these min-paths/min-cuts are manipulated to get their counterpart in the sum of disjoint product form. Min-cuts methods have been used since 1960 to compute the reliability of a network [5]. The authors [6] have shown that min-cuts based algorithms are more efficient then the min-paths based algorithms only for the networks where number of min-cuts are less than the number of min-paths. However the number of min-paths/min-cuts increases exponentially as the size of the networks increases. It is impractical to enumerate all the min-paths/min-cuts of a very large network. Some of the others are based on the factoring theorem [7, 8]. Moskowitz was the first to use the factoring theorem directly to compute the reliability of a network [9]. The factoring theorem divides the reliability problems into two sub problems and the formula is given below:

$$R(G) = P_e \, . \, R(G/\text{ edge e functions}) + (1 - p_e) \, . \, R(G/\text{edge e fail}) \qquad (1)$$

This factoring formula must be applied only when there is no reduction on the graph is possible. It has shown that the optimal binary structure of the factoring algorithm for undirected networks can be generated by means of pivoting. Before applying the factoring, we must apply the reduction techniques like polygon to chain or series-parallel [10]. If a network has imperfect nodes as well as imperfect links, then such type of structure will increase the complexity to compute the reliability of the network. The most commonly used method for nodes failure is incident edge substitution [11]. In this method, for an edge e_i, we put $v_i e_i v_j$ in the min-path function for the perfect nodes, because if we consider an edge e_i, then this edge must contains two end vertices say v_i and v_j. So we have to simplify the Boolean function. The min-path function for the perfect nodes is the union of all min-paths from source to sink. By performing such type of operations we need large memory. One more feasible solution is to slightly change the probability function used in the factoring theorem and factor on links that have at least one end point [12]. The authors [13] have shown an efficient and exact method to compute the reliability of a network with imperfect vertices. One other method was shown by Xing with imperfect coverage and common cause failure [14].

One other author has tried to convert an undirected network in to a directed network and then compute the reliability of a network. This algorithm generates result with minor errors within reasonable time. This algorithm also generates bad result for some networks. This has been shown by Y. Chen [15]. One of the others algorithm is the brute-force algorithm. It uses the path function and have presented by V. A. Netes [16].

Our network model is a directed graph G = (V, E), where V is the vertex set, and E is the set of directed edges. An incidence relation which associates with each edge of G a pair of nodes of G, called its end vertices. The edges and nodes are the components of a network that can fail with known probability. In real problems, these probabilities are usually computed from statistical data. The reliability of a network is the probability that at least one path is operational from source to sink.

This paper is organized as follows. The brief introduction to Binary decision diagrams (BDD) is described in section 2. Three types of network reliability are discussed in section 3. The computation of the network reliability by an exact method is described in section 4. In section 5, the author has described the description of proposed method for computing network reliability by using directed acyclic graph (DAG). Finally the author draws some conclusions.

II. BINARY DECISION DIAGRAMS

A BDD is a DAG. Akers [17] first introduced BDD to represent Boolean functions i.e. a BDD is a data structure used to represent a Boolean Function. Bryant [18] popularized the use of BDD by introducing a set of algorithms for efficient construction and manipulation of BDD structure. The BDD structure provides compact representations of Boolean expressions. A BDD is a directed acyclic graph (DAG) based on the Shannon decomposition. The Shannon decomposition for a Boolean function is defined as follows:

$$f = x.\ f_{x=1} + \bar{x}.\ f_{x=0} \qquad (2)$$

where x is one of the decision variables, and f is the Boolean function evaluated at x = i.

By using Shannon's decomposition, any Boolean expression can be transformed in to binary tree. The authors [19] have shown a method to minimize Boolean expression with sum of disjoint product functions by using BDD. BDD are used to work out the terminal reliability of the links. In the network reliability framework, Sekine & Imai [20] have shown how to functionally construct the corresponding BDD. The authors [21] have shown an alternate approach to find the network reliability by using BDD.

Figure 1 shows the truth table of a Boolean function f and its corresponding Shannon tree. Sink nodes are labelled either with 0, or with 1, representing the two corresponding constant expressions. Each internal node u is labelled with a Boolean variable var(u), and has two out-edges called 0-edge, and 1-edge. The node linked by the 1-edge represents the Boolean expression when $x_i = 1$, i.e. $f_{xi=1}$; while the node linked by the 0-edge represents the Boolean expression when $x_i = 0$, i.e. $f_{xi=0}$. The two outgoing edges are given by two functions low(u) and high(u).

x1	x2	x3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

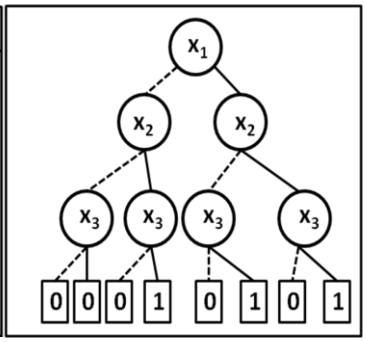

Figure 1. Truth table of a Boolean Function f and its corresponding Shannon's Tree

Indeed, such representation is space consuming. It is possible to shrink by using following three postulates.

Remove Duplicate Terminals: Delete all but one terminal vertex with a given label, and redirect all arcs into the deleted vertices to the remaining one.

Delete Redundant Non Terminals: If non terminal vertices u, and v have var(u) = var(v), low(u) = low(v), and high(u) = high(v), then delete one of the two vertices, and redirect all incoming arcs to the other vertex.

Delete Duplicate tests: If non terminal vertex v has low(v) = high(v), then delete v, and redirect all incoming arcs to low(v).

The shrinking process is shown in figure 2.

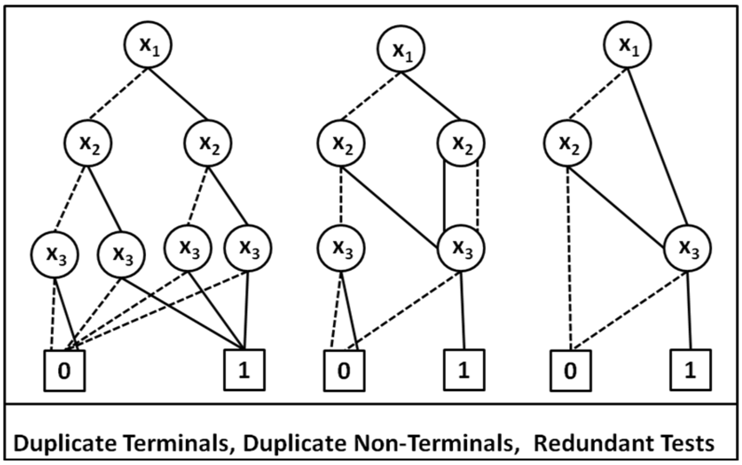

Figure 2. Reduction Process of the decision tree

2.1 Ordered Binary Decision Diagram

For an ordered BDD (OBDD), we impose a total ordering $\prec$ over the set of variables and require that for any vertex u, and either non terminal child v, their respective variables must be ordered. The authors [22, 23] have shown two different methods to find the reliability of the network by using OBDD.

2.2 Dual Binary Decision Diagram

If two or more BDD have the same size and representing the same Boolean function, then these BDD are known as Dual BDD, because they are Dual of each other [24]. The size of the BDD means the total number of non terminal vertices and the number of non terminal vertices at particular level [25]. A particular sequence of variables is known as a variable ordering. It has been observed that the size of the BDD strongly depends on the ordering of variables [26]. There are three types of variable ordering (optimal, good and bad) depending on the size of the different BDD [27]. An ordering is said to be optimal if it generates the minimum size BDD. A new approach for finding various optimal variable ordering to generate minimum size BDD has shown by Singhal [28]. Herrmann has shown the process how to improve the reliability of a network by using augmented BDD [29, 30].

2.3 Modified Binary Decision Diagram

The modified binary decision diagram (MBDD) is a binary decision diagram which is either dual BDD or the smaller size BDD. [31, 32, 33, 34].

III. NETWORK RELIABILITY

The reliability of a network G is the probability that G supports a given operation. We distinguish three kinds of operation and hence three kind of reliability [35].

3.1 Two Terminal Reliability

It is the probability that two given vertices, called the source and the sink, can communicate. It is also called the terminal-pair reliability [36].

3.2 K Terminal reliability

When the operation requires only a few vertices, a subset k of N(G), to communicate each other, this is K terminal reliability [37].

3.3 All Terminal Reliability

When the operation requires that each pair of vertices is able to communicate via at least one operational path, this is all terminal reliability. We can see that 2-terminal reliability and all terminal reliability are the particular case of K-terminal reliability.

IV. RELIABILITY COMPUTATION OF THE NETWORK

Let us take an example of a directed network represented in the form of a directed graph G (V, E) with single source S and single sink T as shown in figure 3. The network has four nodes and five edges. The network has three min-paths from source S to sink T.

These are $H_1 = \{e_1, e_2\}$, $H_2 = \{e_3, e_4\}$ and $H_3 = \{e_3, e_5, e_2\}$.

Let H_1, H_2,-----H_n be the n min-paths from source to sink in a network then the network connectivity function C can be represented as a logical OR of its min-paths.

$C = H_1 U H_2 ----U H_i -----U H_n$

So the point to point reliability is:

$R_s = Pr\{C\} = Pr\{ H_1 U H_2 ----U H_i -----U H_n\}$ --------(3)

So the network connectivity of our network can be expressed as
$C_{1\text{-}4} = e_1 e_2 \text{ U } e_3 e_4 \text{ U } e_3 e_5 e_2$ ------- (4)

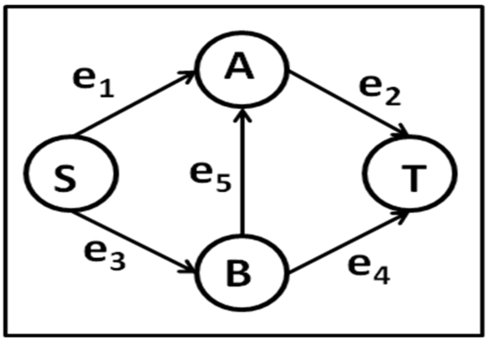

Figure 3. A Directed Network

The probability of the union of non-disjoint events, as in Formula (3), can be computed by several techniques (Exact Methods) [3]. Here we apply the inclusion-exclusion method.

Inclusion-exclusion Method: One method of transforming a Boolean expression $\Phi(G)$ into a probability expression is to use Poincare's theorem, also called inclusion-exclusion method [3]. The inclusion-exclusion formula for two minimal paths H_1 and H_2 is express as follows:
$E(H_1 + H_2) = E(H_1) + E(H_2) - E(H_1.H_2)$

Let P_i denote the probability of edge e_i of being working, by applying the Classical inclusion-exclusion formula for calculating the probability of given network (figure 3), we get

$$Pr = p_Sp_3p_Bp_4p_T + p_Sp_3p_Bp_5p_Ap_2p_T + p_Sp_1p_Ap_2p_T - p_Sp_3p_Bp_1p_5p_Ap_2p_T - p_Sp_3p_Bp_4p_5p_Ap_2p_T - p_Sp_3\, p_Bp_4p_1p_Ap_Ap_Ap_2p_T + p_Sp_3p_Bp_4p_1p_5p_Ap_2p_T \text{ -------------- } (5)$$

V. DAG BASED RELIABILITY COMPUTATION

The conversion of the given network into a DAG involves two main steps. These are as follows:

1. First, the reliability function of the given network is built. The reliability function is the union of all min-paths from source to sink.
2. In the second step, apply a heuristic approach to order the given communication links.

The reliability function R of the given network for perfect nodes is the union of all min-paths from source to sink and is given below:

$$R = H_1 \text{ U } H_2 \text{ U } H_3 \qquad \text{----------------(6)}$$

Again to built the reliability function by considering that the nodes and the links in the CCN may fail. It means the CCN has imperfect vertices as well as imperfect links. To generate the reliability function for imperfect nodes, consider a directed edge e_i as shown in figure 4, then it must contain two end vertices say v_i and v_j. Therefore its incident edge function is $e_i = v_i\, e_i\, v_j$. So by substituting $v_i\, e_i\, v_j$ for e_i in the path function of perfect nodes. We get the reliability function with imperfect nodes [38].

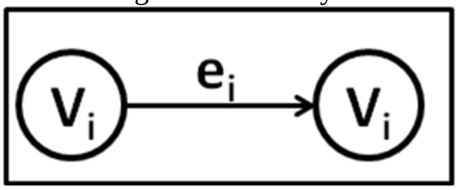

Figure 4: A directed Edge e_i with End Vertices

Therefore the reliability function of the given network with imperfect vertices are defined as

$$\mathbf{Se_1Ae_2T \text{ U } Se_3Be_4T \text{ U } Se_3Be_5Ae_2T} \quad \ldots\ldots\ldots..(7)$$

My proposed heuristic to order the given communication links is as follows:
Heuristic Approach: The heuristic approach is given below:
(i) Traverse the graph from source S to sink T. Find all the min-paths from source to sink. These are $H_1 = \{e_1, e_2\}$, $H_2 = \{e_3, e_4\}$ and $H_3 = \{e_3, e_5, e_2\}$.
(ii) Check whether these paths are disjoint or not. If all the paths are disjoint then we can select any one disjoint path then second then third and so on.
(iii) If all min-paths are not disjoint then find only those min-paths which are disjoint. I have found that the min-paths H_1 and H_2 are disjoint. I can move from source S via min-path H_1 or H_2. To choose either H_1 or H_2, I have found the out degree of the intermediate nodes A and B. Since the out degree of nodes A is one and out degree of node B is two then i have given the preference to the higher out degree node B i.e. preference will be given to min-path H_2, then the middle edge e_5 and then min-path H_1.
By applying our heuristic on the given network, i have found the given variable ordering

$$\mathbf{e_3 < e_4 < e_5 < e_1 < e_2} \text{ (for perfect vertices)}$$
$$\mathbf{S < e_3 < B < e_4 < e_5 < e_1 < A < e_2 < T} \text{ (for imperfect vertices)}$$

The computation of the probability of the BDD can be calculated recursively by resorting to the Shannon decomposition.

$Pr\{F\}= p_1 Pr\{F_{x_1}=1\}+(1 - p1)Pr\{F_{x_1}=0\}= Pr\{F_{x_1}=0\}+ p_1(Pr\{F_{x_1}=1\}-Pr\{F_{x_1}=0\})$ (8)

where p_1 is the probability of the Boolean variable $x_1$to be true and $(1- p_1)$ is the probability of the Boolean variable x_1 to be false.

The DAG and its probability computation for imperfect vertices are shown in figure 5.

5.1 Result: Our program is written in the C language and computations are done by using a Pentium 4 processor with 512 MB of RAM. The computation speed heavily depends on the variables ordering. If the probability of each node or link is 0.9, then from equation 5

Pr = 0.590 + 0.478 + 0.590 – 0.430 – 0.430 – 0.387 + 0.387 = 0.798
Now from our method Pr_s = 0.798

Here I have found that the reliability obtained by DAG is equal to the reliability obtained by inclusion-exclusion formula.

$Pr = Pr_S = p_S p_3 p_B p_4 p_T + p_S p_3 p_B p_5 p_A p_2 p_T + p_S p_1 p_A p_2 p_T - p_S p_3 p_B p_1 p_5 p_A p_2 p_T - p_S p_3 p_B p_4 p_5 p_A p_2 p_T - p_S p_3 p_B p_4 p_1 p_A p_A p_2 p_T + p_S p_3 p_B p_4 p_1 p_5 p_A p_2 p_T$

In a real network the nodes and link may fail. . The complexity of the reliability analysis will grow exponentially if the nodes of the network are imperfect. To overcome this problem, our algorithm based on DAG is proposed to compute the reliability of a network with imperfect nodes.

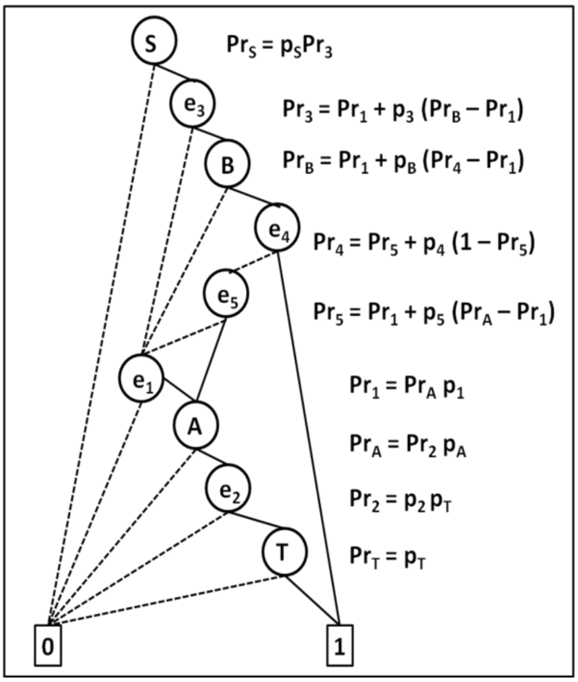

Figure 5. DAG and its reliability Computation

VI. CONCLUSIONS

I have found that the reliability obtained by both the method is same. The size of the BDD is minimal because it has only nine non-terminal vertices and there are only nine variables in the reliability function so its BDD must have at least nine non-terminal vertices. The ordering obtained by our method is optimal. In future we extend our algorithm to compute the K-terminal reliability with imperfect nodes.

REFERENCES

[1] A. Rauzy (1993), "New algorithms for fault tolerant trees analysis", Reliability Engineering and System Safety, vol. 5, no. 59, pp. 203–211.
[2] Andrea Bobbio, Caterina Ferraris, Roberta Terruggia (2006), "New Challenges in Network Reliability Analysis", Technical Report, pp. 1-8, TR-INF-UNIPMN.
[3] C. Lucet and J.-F. Manouvrier (1999), "Exact methods to compute network reliability", in Statistical and Probabilistic Models in Reliability, D. C. Ionescu and N. Limnios, Eds. Birkhauser Boston, pp. 279–294.
[4] D. Torrieri (1994), "Calculation of node-pair reliability in large networks with unreliable nodes," IEEE Trans. Reliability, vol. 43, no. 3, pp. 375–377.
[5] P. A. Jensen and M. Bellmore (1969), "An algorithm to determine the reliability of a complex system," IEEE Trans. Reliability, vol. 18, no. 4, pp. 169–174.
[6] Y. G. Chen and M. C. Yuang (1996) , "A cut-based method for terminal-pair reliability," IEEE Trans. Reliability, vol. 45, no. 3, pp. 413–416.
[7] A. Satyanarayana and M. K. Chang (1983), "Network reliability and the factoring theorem," Networks, vol. 13, pp. 107–120.
[8] L.B. Page and J. E. Perry (1989), "Reliability of directed networks using the factoring theorem", IEEE Trans. Reliability 38: 556-562.
[9] F. Moskowitz (1958), "The analysis of redundancy networks," AIEE Trans. Communications and Electronics, vol. 39, pp. 627–632.
[10] M. Resende (1986), "A program for reliability evaluation of undirected networks via polygon-to-chain reductions," IEEE Trans. Reliability, vol. 35, no. 1, pp. 24–29.
[11] K. K. Agarwal (1975), J.S. Gupta and K. B. Mishra, "A simple method for evaluation of a communication system", IEEE Trans. Communication 23:563-566.
[12] O. Theologou and J. Carlier (1991), "Factoring and reductions for networks with imperfect vertices", IEEE Trans. Reliability, vol. R-40, pp. 210–217.
[13] Sy-Yen Kuo, Fu-Min Yeh, and Hung-Yau Lin (2007), "Efficient and Exact Reliability Evaluation for Networks With Imperfect Vertices", IEEE Transactions on Reliability, vol. 56, no. 2, pp.288-300.
[14] Liudong Xing (2008), "An Efficient Binary Decision Diagram Based Approach for Network Reliability and Sensitivity Analysis", IEEE Trans. On Systems, Man and Cybernetics, pp. 105-115.
[15] Y. Chen, A. Q. Hu, K. W. Yip, X. Hu, and Z. G. Zhong (2003), "A modified combined method for computing terminal-pair reliability in networks with unreliable nodes," in Proceedings of the 2nd Int'l Conference onMachine Learning and Cybernetics, , pp. 2426–2429.
[16] V. A. Netes and B. P. Fillin (1996), "Consideration of node failure in network reliability calculations", IEEE Trans. Reliability 45: 127-128.
[17] B. Akers (1992), "Binary decision diagrams", IEEE Trans. Computers, vol. C-27, pp.509–516, 1978.
[18] R. E. Bryant, "Symbolic Boolean manipulation with ordered binary-decision diagrams", ACM Computing Surveys, vol. 24, no. 3, pp. 293–318.
[19] D. Sensarma, S. Naskar and S. S. Sarma (2012), "On An Optimization Technique using Binary Decision Diagram", IJCSEA, Vol.2 No. 1, pp73-86
[20] H. Imai, K. Sekine, and K. Imai (1999), "Computational investigations of all terminal network reliability via BDDs", IEICE Transactions on Fundamentals, vol. E82-A, no. 5, pp.714–721.
[21] Manoj Singhal, R. K. Chauhan, Girish Sharma(2010) , "An alternate approach to compute the reliability of a computer communication network using Binary Decision Diagrams", Communications in Computer and Information Science, pp. 160-170, IC3 2010, Springer-Verlag Berlin Heidelberg.
[22] F. Yeh, S. Lu, and S. Kuo (2002), "OBDD-based evaluation of k-terminal network reliability", IEEE Trans. Reliability, vol. R-51, no. 4, pp. 443-451.
[23] F. M. Yeh, and S. Y .Kuo (1997), "OBDD based network reliability calculation", IEEE Electronics Letter 33(9) : 759-760.
[24] Manoj Singhal, R. K. Chauhan, Girish Sharma (2012) "Computing Network Reliability with Imperfect Nodes Using Modified Binary Decision Diagram", International Journal of Advances in Engineering and Technology, Vol. 3, issue 2, pp. 606-614.
[25] Manoj Singhal, R. K. Chauhan, , Girish Sharma (2010), "A New Optimal Approach for evaluating the size of BDD (Binary Decision Diagram) for calculating the Reliability of a CCN (Computer Communication Network)". International Journal of Advanced Networking and Applications. Vol. 1 issue 4. pp. 230-235.
[26] Manoj Singhal, R. K. Chauhan, , Girish Sharma (2010), "Effects of Variable Ordering on Binary Decision Diagrams for Computation of Reliability of a Computer Communication Network", Journal of Computer Science, Vol. 4, issue 6, pp. 1837-1845.
[27] Manoj Singhal, R. K. Chauhan, Girish Sharma (2010), "Network Reliability Computation using Different Binary Decision Diagrams", International Journal of Distributed and Parallel Systems, Vol. 1, No. 1, pp. 82-91.

[28] Manoj Singhal, R. K. Chauhan, Girish Sharma (2011), “A New approach for finding the Various Optimal Variable Ordering to generate the Binary Decision Diagrams (BDD) of a Computer Communication Network”, International Journal of Computer Applications, Vol. 31, No.3, pp. 1-8.

[29] J. U. Herrmann (2009), "On Augmented OBDD and Performability for Sensor Networks," Int'l J. Performability Engineering, accepted for publication 2009.

[30] J. U. Herrmann (2010), “Improving Reliability Calculation with Augmented Binary Decision Diagrams”, IEEE International Conference on Advanced Information Networking and Applications, pp. 328-333.

[31] Manoj Singhal, R. K. Chauhan, Girish Sharma (2009), “Use of Modified Binary Decision Diagrams in Reliability Evaluation of a Directed Computer Communication Network”, The Icfai University Journal of Computer Sciences, Vol. III No. 3, pp. 22-30.

[32] Manoj Singhal, (2012), “Reliability Computation with Perfect and Imperfect Vertices via Binary Decision Diagram of a Computer Communication Network”, International Journal of Electronics communication and Computer Engineering, vol. 3, number 6, pp. 1290-1295.

[33] Manoj Singhal, (2012), “Binary Decision Diagram based Reliability Evaluation”, International journal of Engineering and Advanced technology, vol. 2, issue 2, December 2012. (Impact Factor 1.0)

[34] Manoj Singhal, (2012), “An alternate Approach to Compute the Reliability of a Network with Imperfect Nodes using Binary Decision Diagrams”, Oriental Journal of computer Science & Technology, vol. 5, No. 2, pp. 181-188, December 2012. (**Impact Factor 0.1890**)

[35] G. Hardy, C. Lucet, and N. Limnios (2005), “Computing all-terminal reliability of stochastic networks with binary decision diagrams”, in Proc.11th International Symposium on Applied Stochastic Models and Data Analysi, pp. 1468-1473.

[36] S. Kuo, S. Lu, F. Yeh (1999), “Determining terminal pair reliability based on edge expansion diagrams using OBDD”, IEEE Trans. Reliability, Vol. 48, no. 3, pp. 234-246.

[37] Gary Hardy, Corinee Lucet and Nikolaos Limnios (2007), “K-Terminal Network reliability Measures with Binary Decision Diagrams”, IEEE Tran. On Reliability Vol.56, No. 3.

[38] F. M. Yeh, H. Y. Lin and Sy-Yen Kuo (2002), “Analyziang network reliability with imperfect nodes using OBDD”, IEEE proceedings of PRDC’02.

Manoj Singhal is currently working as an Associate Professor, Department of Information Technology in AIMT, Greater Noida (U.P.), India. He had done M.Sc. (Mathematics), MCA and M. Tech (Computer Engg.). Recently he has completed Ph.D (Computer Science & Applications) Kurukshetra University, Kurukshetra. He has written one book on Discrete Mathematics. His research interests include design and development of algorithms.