Reliability and Fault Tolerance based Topological Optimization of Computer Networks - Part I: Enumerative Techniques

Mostafa Abd-El-Barr*, Ahmer Zakir**, Sadiq M. Sait*, and Abdulaziz Almulhem*

***Abstract*— Topological optimization of computer networks is concerned with the design of a network by selecting a subset of the available set of links such that the fault tolerance and reliability aspects are maximized while a cost constraint is met. A number of enumeration-based techniques were proposed to solve this problem. They are based on enumerating all possible paths (for *Terminal* reliability) and all the spanning trees (for *Network* reliability). Existing enumeration-based techniques for solving this network optimization problem ignore the fault-tolerance aspect in their solution. We consider fault tolerance to be an important network design aspect. In this paper, we propose one algorithm for optimizing the terminal reliability and another for optimizing the network reliability while improving the fault tolerance aspects of the designed networks. Experimental results obtained from a set of randomly generated networks using the proposed algorithms are presented and compared to those obtained using existing techniques. It is shown that improving the fault tolerance of a network can be achieved while optimizing its reliability however at the expense of a reasonable increase in the overall cost of the network.**

Keywords: Topological optimization of Networks, Fault Tolerance, Reliability, Enumerative Techniques, Spanning Trees, Computer Networks.

I. Introduction

One major requirement of computer networks is their ability to function even in the presence of some faults in the network. Reliable communication among nodes within a maximum permissible cost is a basic consideration in the design of a computer network. The cost of a network depends in part on the topological layout of the links, their costs and their reliabilities.

Existing enumerative-based techniques include [1], [2], and [3]. In these papers, the authors have proposed three different enumerative techniques for finding the optimal network topology. Aggarwal and Chopra *et al.*, [1] and [2] deal with the terminal reliability while [3] deals with the network reliability. These techniques are based on enumerating all possible paths (for *Terminal* reliability) or all spanning trees (for *Network* reliability). The main shortcoming of these techniques is that they ignore the fault-tolerance aspect in their considerations. We consider fault tolerance to be an important network design aspect. A fault tolerant network is able to function even in faults in the network. This is a basic requirement of a computer network.

The authors are with the Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran - 31261, Saudi Arabia. E-mail: mostafa, sadiq, almulhem@ccse.kfupm.edu.sa. ** Author is with the Information Technology Center, King Fahd University of Petroleum and Minerals. E-mail: azakir@kfupm.edu.sa.

In this paper, we propose one algorithm for optimizing the terminal reliability and another for optimizing the network reliability while improving the fault tolerance aspects of the designed networks.

II. Background Material

A computer network is modeled as a graph in which vertices (or nodes) correspond to computers (and switches) in the network and edges correspond to the links connecting these computers. Every link has a cost and reliability assigned to it. These are shown in the parentheses in Figure 1.

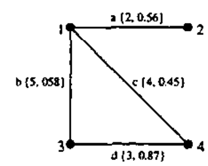

Fig. 1. Graph representation of a network.

- Definition 1: Link *Reliability* is defined as the probability that the link performs its function over a period of time. This reliability has a range from 0 (never operational) to 1 (perfectly reliable).

The reliability of a network can be seen from two different view-points [4], [3]. These are defined below.

- Definition 2: *Terminal Reliability* is defined as the probability that a given pair of nodes in a network is connected.
- Definition 3: *Network Reliability* is defined as the probability that all the nodes in a network are connected.
- Definition 4: A network is said to be *Fault Tolerant* if in the presence of some fault(s), data from a source to a destination can still be routed through some alternate path(s).

The following notations are used in presenting the proposed algorithms.

Common Notations

G	an undirected graph.
N	set of given nodes.
L	number of links.
$c(j)$	cost of link j.
$\Delta D(i)$	a column vector, where $\Delta D(i) = \frac{\Delta R(i)}{\Delta C(i)}$
$\Delta R(i)$	increment in reliability of the network after adding path(spanning tree) i.
$\Delta C(i)$	increment in cost of the network after adding path(spanning tree) i.
$Cost_{max}$	maximum permissible cost for the network.
$SYSCOS$	present cost of the designed system.
$SYSREL$	present reliability of the designed system.

Terminal Reliability Algorithms	
$NPATHS$	number of paths.
P	path array, where $P(i,j) = 1$, if link j is present in path i. $P(i,j) = 0$; $i = 1, 2, \ldots NPATHS$, $j = 1, 2, \ldots L$.
p_j	probability of success of link j.
q_j	probability of failure of link j.
P_c	path cost matrix, where $P_c(i,j) = c(j)$, the cost of link j, if it exists in the path i = 0, otherwise.
P_r	path reliability matrix where, $P_r(i,j) = p_j$, the reliability of link j, if it exists in path i = 1, otherwise.
$Ratio_{Disjoint}$	disjoint ratio matrix where, $Ratio_{Disjoint}(i) = 1 - \frac{\#\ of\ common\ links\ b/w\ paths}{Total\ links\ present\ in\ network}$
C	column vector showing the costs of all paths where $C(i) = \sum_{j=1}^{L} P_c(i,j)$.
R	column vector giving the reliabilities of all paths where $R(i) = \prod_{j=1}^{L} P_r(i,j)$.
D	column vector with entries as the ratio of $R(i)$ and $C(i)$ $\forall\ i$, where $D(i) = \frac{R(i)}{C(i)}$.

Network Reliability Algorithms	
ST	number of spanning trees.
S_c	spanning tree cost matrix, where $S_c(k,j) = C(j)$, the cost of link j, if this link exists in the spanning tree k; = 0 otherwise. $k = 1, 2, \ldots, ST$.
S_r	spanning tree reliability matrix where $S_r(i,j) = p_j$; the reliability of link j, if the link exists in the spanning tree k; = 1 otherwise.
C_N	column vector showing the costs of all spanning trees where $C(k) = \sum_{j=1}^{L} S_c(k,j)$.
R_N	column vector giving the reliabilities of all spanning trees where $R(k) = \prod_{j=1}^{L} S_r(k,j)$
D_N	column vector with entries as the ratio of $R(k)$ and $C(k)$ for all values of k where $D_N(k) = \frac{R_N(k)}{C_N(k)}$.
$Distance$	column vector, where $Distance(k)$ = Distance between the initial spanning tree and spanning tree k.
O	number of spanning trees that have been added to the network.

III. PROPOSED ALGORITHMS

In this section, we introduce two new algorithms for topological optimization of computer networks subject to fault tolerance and reliability. In designing a network, we have considered fault tolerance and reliability as the objectives to achieve, while the cost is considered as the constraint. The proposed algorithms consider *terminal* and *network* reliabilities.

A. Proposed Terminal Reliability Algorithm

For the Terminal Reliability, the idea is that after choosing the first path, we try to find a *totally* disjoint path (instead of adding any path as was done in existing techniques). We start by adding the path which is totally disjoint to the already selected one, and then we continue to add lesser disjoint paths to the network, while not exceeding the cost constraint. The proposed enumerative technique for network reliability is given in Figure 2.

We consider a network to be fault tolerant if there exists two or more totally disjoint paths between the given source-destination pair. In this case, we introduce the following measure of fault tolerance

$$FT = 1 - \left[\frac{\#\ of\ common\ links\ between\ paths}{Total\ \#\ of\ links\ present\ in\ the\ network}\right] \quad (1)$$

Example: Consider the network shown in Figure 3(a). The following specifications are assumed for this network. The total cost allowed is $Cost_{max} = 15$ units.

Algorithm *Proposed Terminal Reliability Algorithm*:
Step 1: Determine all the s-t paths, assuming all possible links in position;
Step 2: Generate the path-cost matrix, P_c, and path reliability matrix, P_r;
Step 3: Generate the matrix C;
Step 4: Generate the matrix R;
Step 5: Generate the matrix D;
Step 6: Choose k such that $D(k) \geq D(i)\ \forall\ i$. Determine $C(k)$ and $R(k)$;
Step 7: Compute the balance cost as $[Cost_{max} - C(k)]$;
If $[Cost_{max} - C(k)]$ is < 0, let $D(k) = 0$, go to Step 6;
If $[Cost_{max} - C(k)]$ is 0, this kth path is the optimum solution; STOP.
Else if $[Cost_{max} - C(k)]$ is > 0, go to the next step;
Step 8: Remove the links already used from further consideration and remove any paths whose cost exceeds the balance cost available. If all the paths are removed, STOP; otherwise go to the next step;
Step 9: Generate matrix $\Delta D(i)$;
Step 10: Generate the matrix $Ratio_{Disjoint}$. Choose the path which has maximum value of $Ratio_{Disjoint}$. If two or more paths have the same $Ratio_{Disjoint}$, select the path which has the maximum $\Delta D(i)\ \forall\ i$ under consideration. Augment the network with links in this path and go back to step 6.
End.(*of algorithm*)

Fig. 2. Proposed terminal reliability algorithm.

Link	a	b	c	d	e	f
Cost	3.30	3.70	1.35	1.25	2.55	7.95
Reliability	0.84	0.76	0.90	0.89	0.94	0.73
Link	i	j	k	g	h	
Cost	6.0	3.0	9.15	3.0	2.0	
Reliability	0.49	0.90	0.78	0.76	0.92	

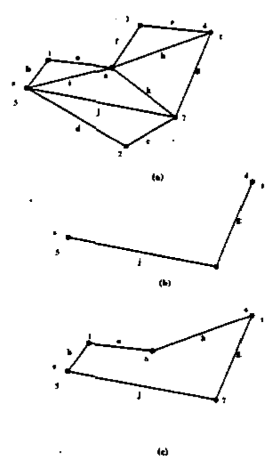

Fig. 3. Example of proposed enumerative technique for terminal reliability.

After placing the initial path *gj*, the paths that can still be added to the network are: *hi*, *cdg*, and *abh*. We try to find a path which is totally disjoint from *gj*, and we select the path *abh* as it is totally disjoint from *gj*. Although the path hi is also totally disjoint from gj but the path abh yields better $\frac{\Delta R}{\Delta C}$ ratio. The final network is shown in Figure 3(c). The terminal reliability of this network is 0.8696, with a cost of 15. The benefit that we obtained by adopting this approach is that now we have 2 totally disjoint paths, which means that in the presence of some fault on a path, the other one can still be used for communication.

Path	hi	cdg	abh
# of common links	0	1	0
Fault Tolerance	1.0	0.75	1.0

B. Proposed Network Reliability Algorithm

For Network Reliability, the idea is that we look for as much disjoint spanning tree as possible to add to the network. We introduce the following measure for fault tolerance

$$FT = \frac{\#\ of\ nodes\ with\ node\ degree\ \geq 2}{Total\ \#\ of\ nodes\ in\ the\ network} \quad (2)$$

The proposed enumerative technique for network reliability is given in Figure 4.

Example: Consider the network shown in Figure 5(a) with the specifications shown below. The total cost allowed is $Cost_{max}$ = 16 units.

Link	a	b	c	d	e	f	g	h
Cost	2.0	3.7	2.7	2.5	4.0	3.0	3.2	3.5
Reliability	0.9	0.6	0.8	0.5	0.9	0.7	0.7	0.8

We determine all the possible spanning trees, and then we select *acef* as it yields the maximum reliability to cost ratio. The cost of this network is 11.7 and the reliability is 0.4536. The network is shown in Figure 5(b). Now, we try to add another spanning tree which has the highest distance from *acef* and which does not exceed the given cost. Based on this criteria, we add *abce* as our second spanning tree. The cost of this network is 15.4 and the network reliability = 0.6685. The resultant network is shown in Figure 5(c).

IV. Results and Comparison

In this section, we compare the results obtained from our proposed enumerative techniques with those obtained using the previous techniques reported in [1], [3].

A. Terminal Reliability Algorithms

Test cases consisting of 11 randomly generated networks are used. The results obtained from applying these techniques are shown in Table I.

As can be seen from the table, in most of the cases, the reliability obtained using the proposed technique is better than that obtained using Chopra's method. Moreover, we were able to achieve 1-fault tolerance in almost all the cases. The only exception is the case of network 2 in Table I. For this network, it is not possible to have a path which is disjoint from the initially placed path *aceh*, from source (1) to destination (6).

But as could be expected, this fault tolerance comes at the expense of a higher cost, as compared to the Chopra's method. This seems reasonable enough because when we try to add totally disjoint path(s) to the network for making it fault-tolerant, we are adding new links to the network, which adds to the cost of the network. It can also be seen that the runtimes for both algorithms are almost equal.

Algorithm *Proposed Network Reliability Algorithm*:

Step 1: Determine all the spanning trees by considering all possible links.
Step 2: Generate S_c;
Step 3: Generate S_r;
Step 4: Generate the matrix C_N;
Step 5: Generate the matrix R_N;
Step 6: Generate the matrix D_N;
Step 7: Choose k such that $D_N(k) \geq D_N(i) \; \forall \; i = 1, 2, \ldots, ST$.
Step 8: Compute the balance cost as $[Cost_{max} - C_N(k)]$:
If $[Cost_{max} - C_N(k)]$ is < 0, let $D_N(k) = 0$, go to Step 7;
If $[Cost_{max} - C_N(k)] = 0$, this is the optimal solution; STOP.
Else if $[Cost_{max} - C_N(k)]$ is > 0, go to the next step;
Step 9: Remove the links already used from the spanning trees to be considered and remove all such spanning trees whose addition is not possible since their cost exceeds the balance cost. If all the spanning trees are removed, STOP; otherwise go to the next step;
Step 10: Generate the matrix $Distance$.
Step 11: Select that spanning tree which has the maximum $Distance(i)$. If two or more spanning trees are equally distant, select the spanning tree which makes the node degree of the nodes 2 having lesser than 2 node degree, the most.
Step 12: Augment the network with links in spanning tree k and go back to step 7.
End(*$of\ algorithm$*)

Fig. 4. Proposed network reliability algorithm.

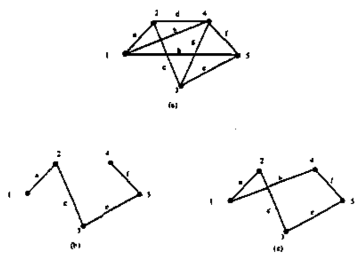

Fig. 5. Example of proposed enumerative technique for network reliability.

B. Network Reliability Algorithms

In the network reliability algorithm, we add fault tolerance by selecting a spanning tree which is as much disjoint as possible, from the already placed spanning tree(s). Test cases consisting of 8 randomly generated networks are used. The results obtained using the previous and the proposed techniques are listed in Table II.

It is observed that the fault tolerance resulting from using our technique is always the same or better than that obtained by using the Aggarwal's method. It is also noted that increasing the fault tolerance of a network is synonymous to increasing the cost of the network. The runtime required by the proposed algorithm is less than that of Aggarwal's and the reason is that we add a spanning tree after placing the initial spanning tree while Aggarwal adds a link at a time to the network.

V. Conclusions

In this paper, the problem of topological optimization of computer networks subject to fault tolerance and reliability constraints is addressed. Two new enumerative techniques, one for the terminal and the other for network reliability, have been proposed and compared with the previous techniques. The results of the proposed techniques are encouraging and we are able to incorporate the issue of fault tolerance in the design process at a reasonable increase in the overall cost of the network.

Acknowledgments

The authors would like to acknowledge the continued support provided by the King Fahd University of Petroleum & Minerals, Saudi Arabia.

References

[1] Y. C. Chopra, B. S. Sohni, R. K.Tiwari, and K. K. Aggarwal, "Network Topology for Maximizing the Terminal Reliability in a Computer Communication Network," *Microelectronics Reliability*, vol. 5, no. 5, pp. 911–913, 1984.

[2] K. Aggarwal, Y. Chopra, and J. S.Bajwa, "Topological layout of links for optimizing the S-T reliability in a Computer Communication System," *Microelectronics Reliability*, vol. 22, no. 3, pp. 341–345, 1982.

[3] K. Aggarwal, Y. Chopra, and J. S. Bajwa, "Topological Layout of Links for Optimizing the Overall Reliability in a Computer Communication System," *Microelectronics Reliability*, vol. 22, no. 3, pp. 347–351, 1982.

[4] A. E. Smith and D. L. Deeter, "A Chapter in Telecommunications Optimization: Heuristic and Adaptive Methods," .

TABLE I

COMPARISON BETWEEN TERMINAL RELIABILITY ALGORITHMS

Network			Chopra's Algorithm				Proposed Algorithm			
N	L	$Cost_{max}$	Rel	Cost	FT	Time	Rel	Cost	FT	Time
5	7	18.0	0.6734	14.0	0.75	0.33	0.7501	18.0	1.0	0.33
6	8	19.0	0.5909	14.0	0.66	0.60	0.7519	19.0	0.857	0.55
7	11	15.0	0.7449	8.6	0.75	4.51	0.8696	15.0	1.0	4.36
8	12	25.0	0.9190	23.3	0.80	9.45	0.8940	24.2	1.0	8.68
9	14	31.1	0.7011	29.8	0.80	52.89	0.7521	30.4	1.0	53.24
10	15	37.2	0.8470	33.5	0.60	73.16	0.7965	36.5	1.0	72.83
11	17	27.0	0.8441	23.8	0.75	315.0	0.8341	25.2	1.0	312.64
12	18	22.2	0.6886	20.6	0.80	668.61	0.7390	21.9	1.0	669.23
13	20	19.0	0.8421	16.3	0.75	3137.86	0.8601	18.4	1.0	3130.50
14	21	27.9	0.7272	24.1	0.80	6203.54	0.7935	27.6	1.0	6175.52
15	23	34.0	0.7374	31.6	0.60	10547.59	0.7950	34.0	1.0	10519.64

TABLE II

COMPARISON BETWEEN NETWORK RELIABILITY ALGORITHMS

Network			Aggarwal's Algorithm				Proposed Algorithm			
N	L	$Cost_{max}$	Rel	Cost	FT	Time	Rel	Cost	FT	Time
5	8	16.0	0.6250	15.2	0.800	0.44	0.6685	15.4	1.0	0.35
6	8	21.0	0.5599	19.0	0.833	0.35	0.6184	21.0	0.833	0.26
6	12	30.0	0.7483	29.0	0.833	2.40	0.7720	30.0	1.0	1.86
7	15	65.0	0.7014	63.4	0.714	81.97	0.7224	64.4	0.857	78.95
8	16	88.0	0.8108	85.7	0.75	105.96	0.8458	87.8	1.0	98.39
9	18	58.5	0.4836	56.6	0.888	2503.69	0.5108	58.3	0.888	2489.92
10	20	69.4	0.5477	68.5	0.800	7784.62	0.6011	69.2	1.0	7751.74
11	21	76.2	0.6741	73.5	0.818	19820.12	0.7111	75.9	1.0	19523.85