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

M. Abd-El-Barr, S. M. Sait, and A. S. Al Mulhem   Ahmer Zakir

Computer Engineering Department
King Fahd University
of Petroleum & Minerals
Dhahran-31261, Saudi Arabia
mostafa, sadiq, almulhem@ccse.kfupm.edu.sa

Information Technology Center
King Fahd University
of Petroleum & Minerals
Dhahran-31261, Saudi Arabia
azakir@ccse.kfupm.edu.sa

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 enumerative and iterative based techniques were proposed to solve this problem. In this paper (Part I), we present and compare the different proposed enumerative techniques for optimizing different aspects (reliability, fault tolerance, and cost) of the designed networks, while Part II will contain a survey of the different iterative techniques to solve the problem of topological optimization.

1 Introduction

One major requirement of computer networks is their ability to function even in the presence of some faults in the network. Reliable communication between some 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. The quality of a designed network can be judged by its reliability. The reliability of a network depends upon the reliability of its nodes, reliability of the links used and the network topology. A topological design involves the determination of the sub-set of links that should be established for an effective communication among the network nodes. This sub-set of links is selected from a pre-specified set of links. A network topology is mostly determined by geographical or physical constraints such as the case of hospitals, business centers, and universities. Under these conditions, the problem is to choose a set of links for a given set of nodes to either maximize reliability given a cost constraint or to minimize cost given a minimum network reliability [1]. It should be noted that if N denotes the number of nodes, the (maximum) number of links in a fully connected network is given by $N(N-1)/2$.

Existing enumerative-based techniques include [2], [3], [4] and [5]. In these papers, the authors have proposed different enumerative techniques for finding the optimal network topology. Aggarwal and Chopra *et al.*, [2] and [3] deal with the terminal reliability while [5] 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. Fault tolerance is an important network design aspect. A fault tolerant network is able to function even in the presence of some faults in the network. This is a basic requirement of a computer network.

Abd-El-Barr and Zakir [6] have proposed one algorithm for optimizing the terminal reliability and another for optimizing the network reliability while improving the fault tolerance aspects of the designed networks.

This paper is organized as follows. In Section 2 we provide some background material. A review of existing enumerative techniques is provided in Section 3. Some analysis and observations are done for the enumerative techniques.

2 Background Material

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

Figure 1.

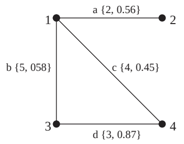

Figure 1: Graph representation of a network.

In designing a network, we have considered fault tolerance and reliability the objectives to achieve, while cost is considered as the constraint.

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). □

Definition 2: *Terminal Reliability* is defined as the probability that a given pair of nodes in a network is connected. □

So in the context of terminal reliability, 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 for terminal fault tolerance

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

Based on this fault tolerance measure, a 1-fault tolerant network is one which retains a single established path between the source-destination pair in the presence of a fault.

Definition 3: *Network Reliability* [5] is defined as the probability that all the nodes in a network are connected. □

Network reliability is concerned with the ability of each and every network node to be able to communicate with all the other nodes.

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). □

In the context of network reliability, we introduce the following measure for network fault tolerance:

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

3 Enumerative Techniques

In this section, we present the main features of existing enumerative-based techniques.

3.1 Terminal Reliability Techniques

3.1.1 Technique 1 [2]

Aggarwal *et al.* [2] proposed a technique for designing a computer network to maximize the *terminal (s-t) reliability* without exceeding the overall permissible cost. The algorithm proceeds as follows.

First, all the paths from the given source to destination are determined. The cost of a path is computed as the sum of cost of all constituent links, whereas the reliability of the path is computed as the product of all the constituent link reliabilities. The path whose reliability to cost ratio is the highest, is chosen. After selection of the first path, other links might be added, depending on the balance cost available. This addition of extra links is done according to the following procedure. A ratio of the increase in terminal (s-t) reliability to the increase in the cost is calculated for every link if it is to be added to the network and the link with the highest ratio is added to the network. This initial path becomes the starting point and the whole procedure is repeated for the remaining possible links, and the cost constraint is kept under consideration throughout this process.

Example: Consider the network of Figure 2(a) as an example with the following specifications. The source s is node 5, while the destination t is node 4. The total cost allowed is $Cost_{max} = 15$ units.

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

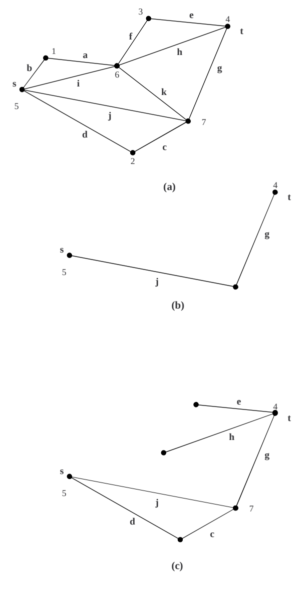

Figure 2: Example of Aggarwal's terminal reliability technique.

The paths are *abef, cdg, abh, efi, hi, gj, cdefk, abgk, cdhk, gik, efjk,* and *hjk.* Now, we select the path *gj* as it has the highest reliability to cost ratio, see Figure 2(b). After placing this initial path, the network has a reliability of 0.6840. Now, we consider one link at a time and since $\Delta D(i)$ for link d is maximum for all i, add the link d to the network. We continue in the same manner and the final network obtained is shown in Figure 2(c). The terminal reliability of this network is 0.7449 with a cost of 13.15. As there is no link whose cost is less than the available cost, the algorithm stops.

3.1.2 Technique 2 [3]

In this approach, Chopra *et al.* proposed a technique that improves over Aggarwal's technique [2]. They do not select any links which cannot provide any additional paths between the source and the destination. The basic difference between this technique and the previous technique is that after the initial path is selected in *Chopra's* technique, a path is selected at a time, rather than trying to add a link at a time to the already placed network [2]. The algorithm proceeds as follows.

Characteristics of all the paths between a source and a destination are first determined. The cost is determined by adding all the costs of the links on the path and the reliability is calculated by multiplying the reliabilities of the selected links. The path for which the reliability to cost ratio is the maximum is selected. This path is ignored from further consideration. The costs of the remaining paths are modified by subtracting the costs of already selected links from their path costs. The remaining paths are arranged in an ascending order of their costs. Any path whose cost exceeds the balance cost. The remaining paths are considered and the increase in cost and reliability for each is determined. The path which has the maximum reliability to cost ratio is retained and the additional links which are included in this path are chosen. This procedure is repeated for the remaining possible paths, and the cost constraint is kept under consideration throughout this process.

Example: Consider the network shown in Figure 3(a). The source s is node 5, while the destination t is node 4. The total cost allowed is $Cost_{max} = 15$ units. We use the same cost and reliability specifications as used in Technique 1.

The paths are *abef, cdg, abh, efi, hi, gj, cdefk, abgk, cdhk, gik, efjk,* and *hjk*. Now, we select the path *gj* as it has the highest reliability to cost ratio, see Figure 3(b). After placing this initial path, the network has a reliability of 0.6840. Now, we subtract the cost of the links g and j, from the cost of all the paths and ignore those paths which exceed the available cost. The reliability to cost ratio of these paths are again calculated, and the path with the highest ratio is selected. Path *cdg* is added to the initial network. The reliability of resultant network is 0.7449 and the cost is 8.6. Since there is no path which can be added within the available cost, the algorithm stops. The network obtained is shown in Figure 3(c).

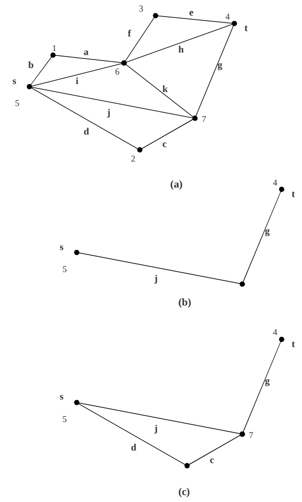

Figure 3: Example of Chopra's terminal reliability technique.

3.1.3 Technique 3 [6]

As can be observed from the above review, there has been no attempt to include the aspect of fault tolerance while optimizing the network reliability. Abd-El-Barr and Zakir [6] introduced an algorithm for topological optimization of computer networks subject to fault tolerance and reliability.

Here, 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 previous 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 steps of the algorithm introduced in [6] are shown in Figure 4.

Example: Consider the network shown in Figure 5(a). The total cost allowed is $Cost_{max} = 15$ units. We use the same cost and reliability specifications as used in Technique 1.

There are 12 different paths that can be established between the source-destination pair. These are *abef, cdg, abh, efi, hi, gj, cdefk, abgk, cdhk, gik, efjk,* and *hjk*. We select the path *gj* as it has the highest reliability to cost ratio, see Figure 5(b). After placing this initial path, the terminal reliability of this network is 0.6840. After placing the initial path, the paths that can still be added to the network are: *hi, cdg*, and *abh*.

Path	hi	cdg	abh
# of common links (with path *gj*)	0	1	0
Fault Tolerance	1.0	0.75	1.0

We try to find a path which is totally disjoint from *gj*, and we select the path *abh* as it is totally disjoint

Step 1: Determine all the s-t paths, assuming all the 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 $\triangle D(i)$;
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 $\triangle D(i)\ \forall\ i$ under consideration. Augment the network with links in this path and go back to step 6.
End.(*$of\ algorithm$*)

Figure 4: Terminal reliability algorithm [6].

from *gj*. Although the path *hi* is also totally disjoint from *gj* but the path *abh* yields better $\frac{\triangle R}{\triangle C}$ ratio. The final network is shown in Figure 5(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 in a path, the other one can still be used for communication.

3.2 Network Reliability Techniques

3.2.1 Technique 1 [5]

In this technique, the authors proposed a method for designing a computer network to maximize the *Network Reliability*. The main idea is to enumerate spanning trees of all possible network topologies. The algorithm proceeds as follows.

The algorithm starts by enumerating all the possible spanning trees. As the spanning tree is an open circuit connecting all the nodes, its cost is the sum of the cost of all the constituent links and its reliability is the product of the reliabilities of all the links. Amongst all possibilities, the spanning tree which has the maximum reliability to cost ratio is chosen. Now depending upon the balance of the cost (out of the permissible cost) available, links are added to the network sequentially. For all the presently available link positions, a ratio of the increase in the overall reliability to the increase in the cost is calculated if any particular link is to be added to the network. That link for which this ratio is maximum is added subject to the permissible cost constraint. The augmented network thus obtained becomes the starting point and the whole procedure is repeated for the remaining possible links, while remaining within the cost constraint.

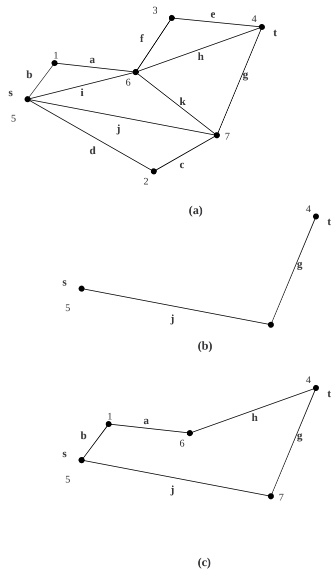

Figure 5: Example of improved version of enumerative technique for terminal reliability.

Example: Consider the network shown in Figure 6(a), with the shown specifications. 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

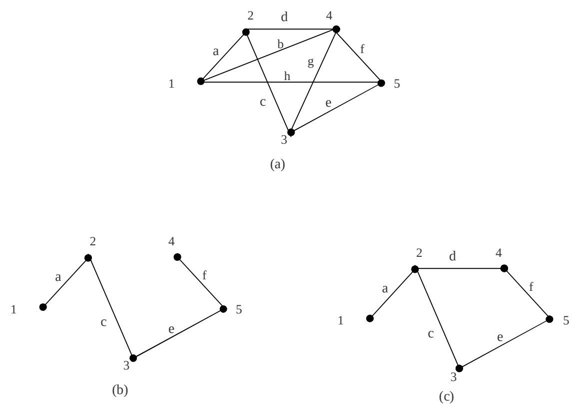

Figure 6: Example of Aggarwal's network reliability technique.

First, we determine all possible spanning trees. For this network, a total of 44 spanning trees are enumerated. We select the spanning tree *acef* as it has the highest reliability to cost ratio, see Figure 6(b). After the addition of the initial spanning tree to the network, a link is added to the network at a time and the reliability to cost ratio is determined. The final network

Step 1: Determine all spanning trees by considering all the 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 $D_N(k) \geq D_N(i)\ \forall\ i = 1, 2, ..., ST$.
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 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.(*$of\ algorithm$*)

Figure 7: Network reliability algorithm [6].

is shown in Figure 6(c). The total cost achieved for the obtained network is 14.2 units and its reliability is 0.6250.

3.2.2 Technique 2 [6]

Abd-El-Barr and Zakir introduced an algorithm for network reliability. The same idea as they applied for the terminal reliability applies to the network reliability, except that here they look for as much disjoint spanning tree as possible. The steps of the algorithm are given in Figure 7.

Example: Consider the network shown in Figure 8(a). The total cost allowed is $Cost_{max} = 16$ units. We use the same cost and reliability specifications as used in Technique 1.

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 8(b). Now, we try to add another spanning tree which has the highest distance from *acef* and which also 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 8(c). As there can be no subnets to add, the algorithm stops.

4 Conclusions

In this paper, the work which has been done to solve the problem of topological optimization of computer networks is reviewed. Different enumerative techniques have been discussed and the shortcoming of the earlier techniques has been highlighted, which overlooked the fault tolerance of a network while designing the network.

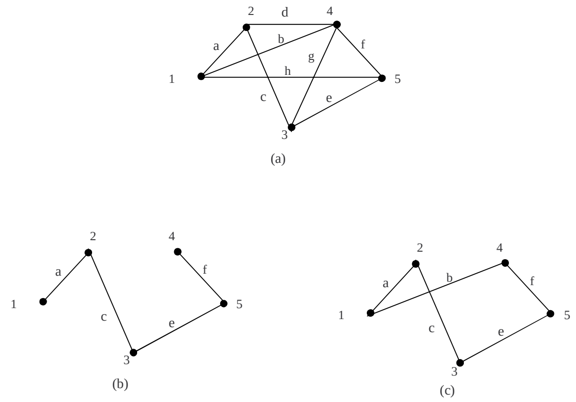

Figure 8: Example of improved version of enumerative technique for network reliability.

Acknowledgments

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

References

[1] B. Dengiz, F. Altiparmak and A. E. Smith, "Local Search Genetic Algorithm for Optimal Design of Reliable Networks," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 3, pp. 179–188, September 1997.

[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] 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.

[4] Usha Sharma, K. B. Misra and A. K. Bhattacharya, "Optimization of CCNs: Exact and Heuristic Approaches," *Microelectronics Reliability*, pp. 43–50, 1990.

[5] 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.

[6] M. Abd-El-Barr and Ahmer Zakir, "Enumerative Techniques for Topological Optimization of Computer Networks Subject to Fault Tolerance and Reliability," *IASTED PDCS conference*, November 2002.