Analysis of Centralized Network Restoration[1]

C. Edward Chow, Viji Narasimhan
Department of Computer Science
University of Colorado at Colorado Springs
Colorado Springs, CO 80933-7150
Phone: (719) 593-3110
Email: {chow, vnarasim}@sanluis.uccs.edu

and

Sami Syed
MCI Telecommunications
4678 Alpine Meadows Lane
Colorado Springs, CO 80919
Phone: (719) 592-1461
Email: 0002912712@mcimail.com

Abstract

The three phases of the centralized network restoration, i.e., diagnosis, path finding, and connection re-establishment, are analyzed in detail. Methods applied at each phase of the restoration process are presented and their simulation results compared. These different centralized approaches together with four distributed network restoration approaches were simulated under the same network assumptions where the warning messages can be filtered and only essential warning messages are sent to the Network Operation Center. For the node failures, the simulation results indicated all the centralized approaches outperforms the distributed network restoration algorithm that handles node failures. For the link failures, the performance of centralized network restoration approaches is close to that of distributed network restoration approaches.

Keywords: Network Restoration, Network Diagnosis, Centralized Algorithm, Distributed Algorithms, Simulation

I. Introduction

Since the publication of Grover's Self-Healing Network [GROV87], a number of different distributed network restoration algorithms have been proposed for DCS and ATM based networks. These algorithms are: the Self-Healing Network (SHN) [GROV87] and [GROV89], FITNESS [YH88], RREACT [CHOW92], Komine [KCOMS90] and Two Prong [CHOW93]. There are few published results on the analysis of centralized network restoration algorithm and on the comparison of centralized and distributed network restoration approaches. In this paper the performance of centralized network restoration approach is examined in detail. Both centralized and distributed network restoration approach-

---

[1] This research was supported by MCI with grant #01-80046.

es are simulated with the same network assumptions and their simulation results are analyzed. It is hoped that the simulation results and the analysis presented here can provide data for improving the reliability of future networks.

The rest of this paper is organized as follows: Section II briefly overviews the centralized network restoration approaches. Section III analyzes the network diagnosis phase of the centralized network restoration approaches. Section IV analyzes the path finding phase. Section V analyzes the connection re-establishment phase. Section VI discusses the simulation results of centralized and distributed network restoration approaches. Section VII is the summary.

II. Centralized Network Restoration

In a centralized network restoration system, the restoration of disrupted paths due to a node failure or link failure goes through three phases: diagnosis, path finding, connection re-establishment. In diagnosis phase, the Network Operation Centre (NOC) has to identify what the failure is. In the path finding phase, it finds alternate paths for the traffic disrupted by the failure. In the connection re-establishment phase, NOC sends connection re-establishment messages to the nodes for establishing the new paths. To achieve a low restoration time, each phase has to be completed in as little time as possible. In the following, we consider each of these phases in detail and discuss the different methods used to obtain a low restoration time.

III. Network Diagnosis

The NOC makes a diagnosis about a failure by analyzing the warning messages that it receives. The time taken by the NOC to make this diagnosis is mainly determined by the number of messages it has to process. When there is a node or a link failure in a network, warning messages may be sent by the immediate nodes as well as the other nodes in the paths which are affected by the failure. This would result in a large number of messages to be processed by the NOC and hence a larger diagnosis time. If the number of messages that the NOC has to process can be reduced, then the diagnosis time can be reduced. It is found that the NOC can make a correct diagnosis by correlating the messages from the immediate neighbors of a failed node, or from the nodes at the two ends of a failed link. The NOC does not need those messages from the other nodes along the path. There two ways we can reduce the warning message sent to NOC: The messages from the nodes adjacent to the failed node or link can be tagged as *direct* messages and those from the other nodes in the path can be tagged as *indirect* messages. The NOC can then process the direct messages alone. The other way is to implement a filtering mechanism that will prevent the other nodes from generating any warning messages. Only the nodes immediately next to the failure will generate warning messages. This will reduce the number of messages that the NOC has to process to two for a link failure, and a maximum of ‘n’ messages for node failure, where ‘n’ is the number of neighbors of the failed node. Since the NOC has the network topology information, it has knowledge of previous node failures. Therefore if one of the neighbors of a failed node is already down, the NOC will know not to expect any messages from that node.

A. Analysis of Diagnosis Time for Link Failure

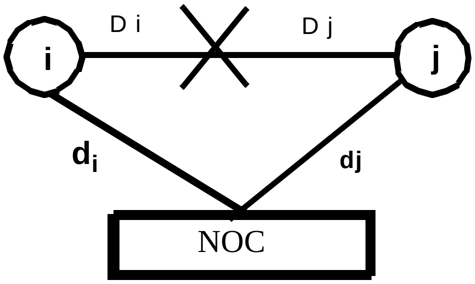

Figure 1. Diagnosis of Link Failure

In the case of the link failure shown in Figure 1, the time taken for diagnosis would be the maximum of the time taken by the warning messages from node i to reach NOC and the time taken by the warning message from node j to reach NOC, plus the processing time at the NOC.

$$Tij = max(Di/s + pi + di/s + ti + 2 \times lp, Dj/s + pj + dj/s + tj + 2 \times lp) + pnoc$$

Tij: Time taken by the NOC to detect link failure

Dx: distance between point of disruption and node x

dx: distance between node x and NOC

tx: transmission delay for message from node x

lp: layer propagation delay

s: speed of light

The layer propagation delay is the time it takes for the message to travel through the various protocol layers.

B. Analysis of Diagnosis Time for Node Failure

In the case of the node failure shown in Figure 2, the NOC waits for warning messages from nodes i, l, and m which are neighbors of the failed node j. The time taken for each of these messages to reach NOC is the sum of the propagation delay between the failed node and that node, the propagation delay from node to NOC, the processing time at the node, the transmission delay, and the layer propagation delay. The total time would be the sum of the maximum of the time taken by the messages from nodes i, l, m, and the processing time at NOC.

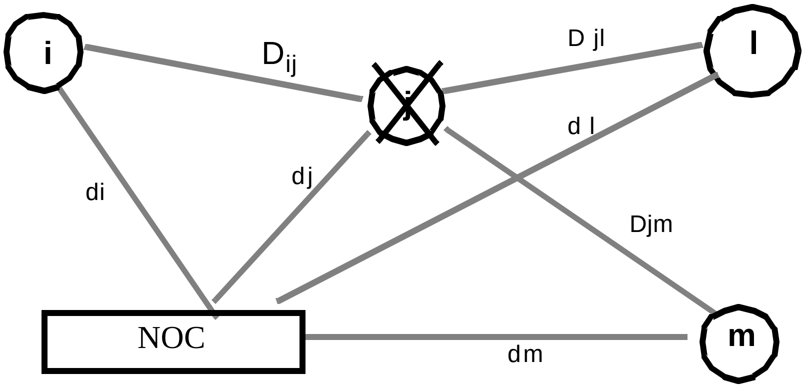

Figure 2. Diagnosis of Node Failure.

$$Tj = max(Dij/s + pi + di/s + ti + 2 \times lp, Djl/s + pl + dl/s + tl + 2 \times lp, Djm/s + pm + dm/s + tm + 2 \times lp) + pnoc$$

Tj: Time taken for the NOC to diagnose failure of node j

Dxy: distance between nodes x and y

dy: distance between NOC and node y

px: processing time at node x

tx: transmission delay for message from node x

lp: layer propagation delay

s: speed of light

IV. Path Finding for Network Restoration

After making a diagnosis, the NOC has to find alternate paths to restore the disrupted traffic. The problem of finding alternate paths can be mapped to the maxflow problem. The maxflow algorithm proposed by Goldberg and Tarjan [GT88] determines the maximum capacity between two nodes in a network by manipulating a preflow on the network. A preflow is a real valued function on vertex pairs (v,w) that satisfies:

$$f(v, w) \leq c(v, w), f(v, w) = -f(w, v) \forall ((v, w) \in V \times V)$$

where f(v, w) is the flow between nodes v and w and c (v,w) is the capacity between nodes v and w and

$$\Sigma f(u, v) \geq 0 \forall v \in V - \{source\}$$

The algorithm tries to push as much flow from the source to its immediate neighbor nodes estimated to be closer to the sink and to have a positive flow excess. The goal is to push as much excess to the sink. If the sink is not reachable from the node with positive excess, the algorithm pushes the excess back towards the source. Thus the algorithm will eventually reach a state when all vertices other than the source and sink will have zero excess. We use an implementation of this algorithm to find alternate paths. The maxflow algorithm is used in three different ways to find alternate paths: A) path-based restoration, B) link-based restoration and C) Combination method.

A. Path-based Restoration

In the path restoration approach, an alternate path is found for each path affected by a link or a node failure. This is done by calling maxflow program with the source and destination nodes of the affected path. The Maxflow program finds the maximum available capacity between the two nodes. A path tracing algorithm then traces the flow for the paths with the required bandwidth.

B. Link-based Restoration

In the link-based restoration approach the maxflow algorithm is used differently. If the failure is a link failure, then the maxflow algorithm is used to restore all the traffic through the link. If it is a node failure, the maxflow algorithm is used to reroute the affected paths by connecting the node just before the failed node to the node immediately after the failed node. This can be done in two ways: each path can be restored separately, or all the paths that have the same previous node and next node can be consolidated and alternate paths between the two nodes for the required bandwidth can be found. The new paths bypass the failed node, or the failed link.

The new paths found using the link-based restoration approach may contain cycles in them. This is because even though both the original path and the restored segment are unique, when the two are combined, some nodes may be duplicated. This results in unnecessary use of spare channels.

C. The Combination Restoration

The combination method improves the link-based restoration by inserting the rerouted path segments in the affected path, followed by removing any cycles that are present. The channels on the removed cycles are released.

Removal of cycles

If a path has more than one duplicate node, the order in which these nodes are removed becomes critical. The removal order determines the length of the resulting path. For example, in the path 1-2-5-4-2-1-5-4-3, nodes 1,2,4 and 5 are repeating nodes. If the order of removal is (1,2,5,4) the resulting path is 1-5-4-3. But if the order of removal is (5,4,1,2), the resulting path is 1-2-5-4-3, which is longer.

To make sure that the path obtained after cycle removal is the shortest, all possible orders of removal have to be examined. If there are ‘k’ duplicate nodes, the number of all possible ways in which the ‘k’ nodes can be ordered is k!. This is the upper limit on the number of iterations that have to be performed to find the shortest path.

Heuristics can be used to reduce the number of actual iterations. For example, if there is no direct link between two nodes then the shortest path between the two nodes has to be at least two hops in length. Therefore as soon as a path of length two hops is found the algorithm can stop. Another heuristic is that even if a link exists between two nodes, if the spare capacity along that link is zero then the shortest path is at least two hops in length.

D. Comparison of the three methods

Table 1-4 shows the simulation result of three centralized network restoration approaches based on the above methods for a 11-node, 23-link network. On average, link-based restoration is faster than the other two methods. But it may result in unnecessary use of a large number of spare channels. The path-based restoration method is slower because maxflow is called once for each affected path. The combination method is in the middle in terms of speed, but makes better use of spare channels than the link-based restoration.

Path-based restoration was able to obtain 100% restoration for all failures. Since this method restores each affected path separately, it makes better use of the spare channels. For node NO5 failure, removing the cycles and releasing the spares resulted in 100% restoration by the combination method, as compared to 90% by the link-based restoration method. But in some cases, removing the cycles and releasing channels does not result in extra bandwidth for that restoration. This can be seen in the node N01 failure situation. Even after the removal of cycles and the release of spares (82 spares) the level of restoration is the same as that of link-based restoration. This is because the released spares are at the corners of the network and there is no spare channel available in the middle of the network. Therefore paths from one corner of the network to the other cannot be established.

V. Connection Re-Establishment

In the connection establishment phase, the NOC sends messages to the nodes for setting up the new paths. In the path-based restoration method, the NOC sends connection messages to all the nodes in the new path. In the link-based restoration method, a segment of a path with failure is replaced with a new path and connection messages are sent only to the nodes in the new path. In the combination method, after removing the cycles a new path is obtained, connection messages are sent to all the nodes in the new path. The time taken to send a connection message to a node is the sum of the propagation delay from the NOC to the node, the transmission delay and the layer propagation delay at the two ends. After the connection message reaches a node, the node makes a connection between the spare channels in its links indicated in the message. The connection incurs some delay. Sending connection messages to the nodes can be done in two ways: Unconsolidated Method and Consolidated Method.

A. Unconsolidated Method

In the unconsolidated method, as soon as a path is found, the NOC sends out connection messages to all the nodes in the path and the connection for the path is made. The connection messages for the paths are sent sequentially.

B. Consolidated Method

In the consolidated method, no connection messages are sent out until all possible restoration paths are found. After all the paths have been traced, the connection messages to the nodes are consolidated. For each node, the connection information it needs to establish all the paths which the node involved in are consolidated in a single message. In this method, the maximum number of connection messages that have to be sent is the number of non-duplicate nodes in the set of restoration paths.

The unconsolidated method results in some bandwidth being restored early on in the restoration process. But the time taken for full restoration is longer. The consolidated method results in restoration starting later than in the unconsolidated method. But the total restoration time is smaller than that of the unconsolidated method.

Figure 3 shows the simulation results of the consolidated and unconsolidated connection reestablishment on link failure between N07-N08 using the path-based approach in the New Jersey network shown in Figure 5. The unconsolidated approach can restore 8.53% of channels as early as 0.0762 seconds. By the time the consolidated approach starts to restore the first 6% of channels at 0.2714 seconds, the unconsolidated approach has already achieved 37.7% restoration level. However, due to the higher level of parallelism, the consolidated approach re-establishes disrupted channels very quickly and reach 100% in just 0.0804 seconds. Due to the sequential nature of the unconsolidated approach, it will not reach 100% restoration until 0.864 seconds. Note that in terms of restoration level, before the 0.287 second mark, the unconsolidated approach outperforms the consolidated approach. Between the 0.287 and 0.864 second marks, the consolidated approach has the higher restoration level.

Note that if a time limit is set for network restoration before the network should tear down the disrupted paths that are not yet restored, then we may decide to choose the proper connection re-establishment approach depending on the time limit. If the channels have different priority levels, the unconsolidated approach has the advantage of re-establishing those higher priority paths.

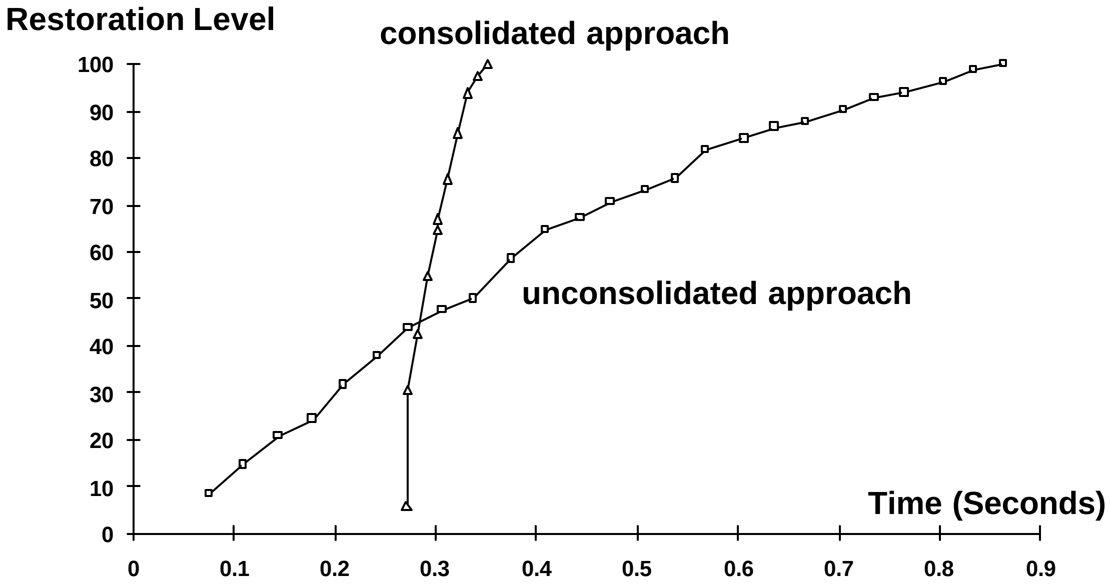

Figure 3. Consolidated vs. Unconsolidated Connection Re-establishment

Tables 1-4 show that the unconsolidated method takes longer to achieve full restoration. But the restoration starts early on. The consolidated method produces full restoration in a shorter time, but it takes longer for the restoration to start.

Figure 4 shows a breakup of the total time taken for the restoration of node N05 failure, using path-based restoration method, link-based restoration method, and combination method. The connection establishment method used is the consolidated connection re-establishment method. The link-based restoration method has a smaller path finding time than those of the path-based restoration and combination method. This is because in link-based restoration all paths that have the same previous node before node N05 and the same node next to node N05 are consolidated. The maxflow algorithm is called once to trace all the required paths for that bandwidth. The path-based restoration takes longer time in the path finding phase, as it considers each affected path separately and calls maxflow individually for each path. The path finding time of the combination method includes the path tracing time for the maxflow and the time taken to remove the cycles present in the new paths.

The combination method has a smaller connection re-establishment time as the cycles have been removed and the paths are on average shorter. In the link-based restoration method, the connection re-establishment messages are sent only to the segments of the paths that now replaces the failed part of the original paths. In the path-based restoration method, new paths are found to replace the affected paths and the connection re-establishment messages have to be sent to all the nodes in the path.

DT = 90 msec | PF = 160 msec | CE = 160 msec

a.Path Restoration

DT = 90 msec | PF = 140 msec | CE = 100msec

b.Link Restoration

DT = 90msec | PF = 160msec | CE = 80msec

c.Combination Method

DT = Diagnosis Time

PF = Path Finding Time

CE = Connection Reestablishment Time

Fig. 4. Breakup Of Total Time Taken For Restoration Of Node N05

VI. Comparison of Centralized and Distributed Network Restoration Approaches

The three versions of centralized network restoration algorithms and four distributed network restoration algorithms, Two Prong, FITNESS, RREACT, and Komine algorithms were implemented using the NETRESTORE simulation system developed at the University of Colorado at Colorado Springs. All algorithms were tested on the ‘New Jersey’ LATA test network which was defined in the FITNESS [YH88] paper and is shown in Figure 5. A traffic pattern which describes the individual paths over this network for testing the node failures is detailed in Appendix A.

All algorithms were identically tested under the following assumptions:

a. All messages have equal priority.

b. All messages are serviced by a node in the order they are received.

c. It requires 10 msec to process any incoming messages.

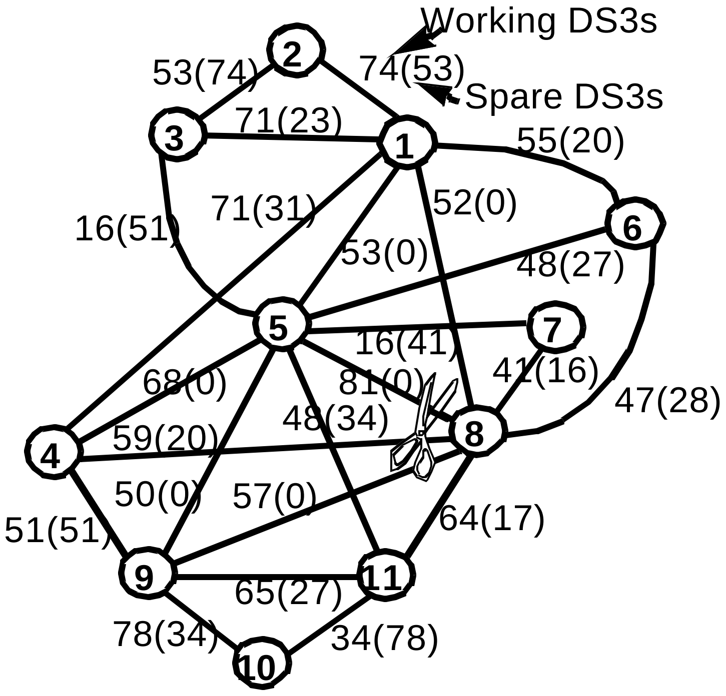

Figure 5. The New Jersey LATA Network.

d. It requires an additional 10 msec to generate each outgoing message.

e. Propagation speed of messages is 200,000 km/sec.

f. Transmission delays are computed for variable length messages, using a 100,000 baud transfer rate (i.e., 1 msec per byte).

The Two Prong and Self-Healing Network algorithms are currently being implemented on the NETRESTORE simulator. Independent test results are not fully available for these algorithms.

Table 5 shows the simulation results of network restoration algorithms for link failures. In terms of time to restoration performance metric, when the number of restored paths is small, the Two Prong algorithm outperforms other algorithms. When the number of paths increases to more than three, the centralized approaches perform better than the distributed approaches. The spare channel usage varies among these algorithms. There are no definite winners. In general, the Two Prong algorithm performed better than the other distributed algorithms with respect to the time to restoration performance metric. This is primarily due to the aggressive nature of the algorithm in identifying, selecting and connecting restoration paths. The algorithm’s time to restoration performance is also enhanced in that the hand shaking required to make final connection of the disrupted ends is done over connected paths. In smaller networks, the RREACT algorithm is able to compete relatively closely to the Two Prong algorithm in terms of time to restoration, but as network size increases, the Two Prong algorithm is able to clearly outperform RREACT in this regard. This is due to a high degree of congestion at the Chooser node in the RREACT algorithm, while the Origin nodes in the Two Prong algorithm have a much lower level of congestion.

Table 6 shows the simulation results of network restoration algorithm for node failures. It was a surprise that our implementation of Komine algorithm, which sends multicast messages to the upstream nodes that are two hops away and deploys a multi-wave method, did not perform as well as the three centralized network restoration algorithms. The noticeable differences are almost two orders of magnitude of the message volume and four to ten times the execution time. The reason is the huge overhead of message processing due to messages generated for the large number of paths involved with the failure node. The centralized approaches on the other hand have very low message volume and each of the connection re-establishment messages contributes to the path restoration. The consolidated approach mentioned in Section IV.B is used in this simulation. For each node involved with the connection re-establishment, only one message is sent from the NOC to the node.

VII. Summary

The three phases of the centralized network restoration, i.e., network diagnosis, path finding, connection re-establishment are examined in detail. It is found that the centralized network restoration can perform more efficient network diagnosis if the warning messages from the disrupted area can be filtered and reduced. With the computation power of today’s workstations, the path finding based on heuristic max flow algorithm [Tarjan88] can be done in subseconds. The link-based path finding is the fastest but there are not much difference between the time spent on path-based path finding and that on the link-based path finding. Although the modified path finding approach, which starts with link-based path finding, merges the rerouted segments with the original path, and then removes the cycles, can achieve better spare channel utilization but the cycle removal process is NP problem. However on existing networks, the restoration paths are usually not long and the number of cycles and duplicate nodes are not big. The cycle removal process did not result in long computation time. The connection re-establishment can be done by issuing path restoration messages either on a path-by-path basis, or by consolidating all the connection operations to be performed by each node and only sending a single restoration message to each involved node. Preliminary simulation results indicate that for link failure situations, the distributed network restoration performs better than the centralized network restoration, but for node failure situations, the centralized network restoration is more efficient. The centralized network restoration also generates fewer messages.

Acknowledgment

We would like to thank Lars Starbuck and Jim Williams for sharing their implementation of the max flow algorithm and thank John Bicknell for sharing the preliminary simulation of Komine algorithm.

Reference

[CHOW92] C.-H.E. Chow, S. McCaughey, and S. Syed, “RREACT: A Distributed Protocol for Rapid Restoration of Active Communication Trunks”, UCCS Tech Report EAS-CS-92-18, November 1992.

[CHOW93] C.-H. E. Chow, J. Bicknell, S. McCaughey, and S. Syed, “A Fast Distributed Network Restoration Algorithm,” *Proceeding of 12th International Phoenix Conference on Computers and Communications*, March 1993, Tempe, Arizona.

[GROV87] W. D. Grover, “The self-healing network: A fast distributed restoration technique for networks using digital cross-connect machines,” in *Proceedings of GLOBECOM ‘87*, pp. 28.2.1–28.2.6, 1987.

[GROV89] W.D. Grover, “SELFHEALING NETWORKS: A Distributed Algorithm for k-shortest link-disjoint paths in a multi-graph with applications in real time network restoration”, in *Doctoral Dissertation for the Department of Electrical Engineering, University of Alberta*, Fall 1989.

[KCOMS90] H. Komine, T. Chujo, T. Ogura, K. Miyazaki, and T. Soejima, “A distributed restoration algorithm for multiple-link and node failures of transport networks,” in *Proceedings of GLOBECOM ‘90*, (San Diego), pp. 403.4.1–403.4.5, Dec. 1990.

[YH88] Yang, C. H. and S. Hasegawa, “FITNESS: Failure Immunization Technology for Network Service Survivability,” *Proc. of GLOBECOM ‘88*, pp. 47.3.1-47.3.6, November 1988

[GT88] A.V. Goldberg and R.E.Tarjan, “A new approach to the Maximum-Flow Problem,” *Journal of the ACM*, pp. 921-940,October 1988

Table 1: Link Failure Restoration Results Using Unconsolidated Approach

Scenario	Perf. Metric	Path Res.	Link Res.	Combination
New Jersey Single Link Failure N01 - N02 74 ch. lost	Time msec Level # of Msgs (# of Paths)	784 100% 74 19	714 100% 98 11	726 100% 100 11
New Jersey Single Link Failure N01 - N03 71 ch. lost	Time msec Level # of Msgs (# of Paths)	688 100% 82 15	529 100% 36 5	559 100% 68 5

Table 1: Link Failure Restoration Results Using Unconsolidated Approach

Scenario	Perf. Metric	Path Res.	Link Res.	Combination
New Jersey Single Link Failure N01 - N05 53 ch. lost	Time msec	529	464	468
	Level	100%	100%	100%
	# of Msgs	89	37	51
	(# of Paths)	13	7	7
New Jersey Single Link Failure N02 - N03 53 ch. lost	Time msec	390	408	469
	Level	100%	100%	100%
	# of Msgs	44	46	65
	(# of Paths)	13	9	11
New Jersey Single Link Failure N04- N05 16 ch. lost	Time msec	446	404	416
	Level	100%	100%	100%
	# of Msgs	43	50	50
	(# of Paths)	13	11	11
New Jersey Single Link Failure N04 - N08 59 ch. lost	Time msec	730	571	567
	Level	100%	100%	100%
	# of Msgs	**89**	55	85
	(# of Paths)	19	14	14
New Jersey Single Link Failure N05 - N08 81 ch. lost	Time msec	645	801	748
	Level	100%	100%	100%
	# of Msgs	56	72	63
	(# of Paths)	15	13	12
New Jersey Single Link Failure N06 - N08 47 ch. lost	Time msec	775	473	499
	Level	100%	100%	100%
	# of Msgs	64	48	59
	(# of Paths)	13	8	9
New Jersey Single Link Failure N07 - N08 41 ch. lost	Time msec	852	673	727
	Level	100%	100%	100%
	# of Msgs	94	61	62
	(# of Paths)	19	6	6
New Jersey Single Link Failure N08 - N11 64 ch. lost	Time msec	734	722	780
	Level	100%	100%	100%
	# of Msgs	114	104	113
	(# of Paths)	17	13	13

Table 1: Link Failure Restoration Results Using Unconsolidated Approach

Scenario	Perf. Metric	Path Res.	Link Res.	Combination
New Jersey Single Link Failure N09 - N11 65 ch. lost	Time msec	340	348	344
	Level	100%	100%	100%
	# of Msgs	48	67	46
	(# of Paths)	11	11	11

Table 2: Node Failure Restoration Results Using Unconsolidated Approach

Scenario	Perf. Metric	Path Res.	Link Res.	Combination
New Jersey Single Node Failure N01 221ch. lost	Time msec	913	750	732
	Level	100%	81.9%	81.9%
	# of Msgs	113	78	83
	(# of Paths)	28	15	14
New Jersey Single Node Failure N03 16ch. lost	Time msec	222	199	199
	Level	100%	100%	100%
	# of Msgs	23	18	25
	(# of Paths)	6	4	4
New Jersey Single Node Failure N04 52 ch. lost	Time msec	263	209	209
	Level	100%	100%	100%
	# of Msgs	33	28	29
	(# of Paths)	7	4	4
New Jersey Single Node Failure N05 148 ch. lost	Time msec	820	850	850
	Level	100%	90.1%	100%
	# of Msgs	98	100	93
	(# of Paths)	24	21	21
New Jersey Single Node Failure N06 55 ch. lost	Time msec	291	233	241
	Level	100%	100%	100%
	# of Msgs	27	25	23
	(# of Paths)	8	5	5
New Jersey Single Node Failure N08 197 ch. lost	Time msec	1034	1015	985
	Level	100%	100%	100%
	# of Msgs	122	121	113
	(# of Paths)	31	26	25

Table 2: Node Failure Restoration Results Using Unconsolidated Approach

Scenario	Perf. Metric	Path Res.	Link Res.	Combination
New Jersey Single Node Failure N11 52 ch. lost	Time msec	348	315	315
	Level	100%	100%	100%
	# of Msgs	49	54	54
	(# of Paths)	8	7	7

Table 3: Link Failure Restoration Results Using Consolidated Approach

Scenario	Perf. Metric	Path Res	Link Res	Combination
New Jersey Single Link Failure N01 - N02 74 ch. lost	Time msec	275	246	309
	Level	100%	100%	100%
	# of Msgs	9	12	13
	(# of Paths)	19	11	11
New Jersey Single Link Failure N01 - N03 71 ch. lost	Time msec	258	214	162
	Level	100%	100%	100%
	# of Msgs	10	7	12
	(# of Paths)	15	5	5
New Jersey Single Link Failure N01 - N05 53 ch. lost	Time msec	189	218	218
	Level	100%	100%	100%
	# of Msgs	12	7	11
	(# of Paths)	13	7	7
New Jersey Single Link Failure N02 - N03 53 ch. lost	Time msec	199	206	230
	Level	100%	100%	100%
	# of Msgs	11	10	13
	(# of Paths)	13	9	11
New Jersey Single Link Failure N04- N05 16 ch. lost	Time msec	208	174	151
	Level	100%	100%	100%
	# of Msgs	10	12	12
	(# of Paths)	13	11	11

Table 3: Link Failure Restoration Results Using Consolidated Approach

Scenario	Perf. Metric	Path Res	Link Res	Combination
New Jersey Single Link Failure N04 - N08 59 ch. lost	Time msec	356	209	190
	Level	100%	100%	100%
	# of Msgs	12	13	13
	(# of Paths)	19	14	14
New Jersey Single Link Failure N05 - N08 81 ch. lost	Time msec	376	467	576
	Level	100%	100%	100%
	# of Msgs	11	12	12
	(# of Paths)	15	13	12
New Jersey Single Link Failure N06 - N08 47 ch. lost	Time msec	434	333	193
	Level	100%	100%	100%
	# of Msgs	10	10	13
	(# of Paths)	13	8	9
New Jersey Single Link Failure N07 - N08 41 ch. lost	Time msec	277	365	336
	Level	100%	100%	100%
	# of Msgs	12	9	12
	(# of Paths)	19	6	6
New Jersey Single Link Failure N08 - N11 64 ch. lost	Time msec	411	417	249
	Level	100%	100%	100%
	# of Msgs	12	13	13
	(# of Paths)	17	13	13
New Jersey Single Link Failure N09 - N11 65 ch. lost	Time msec	340	348	174
	Level	100%	100%	100%
	# of Msgs	10	12	12
	(# of Paths)	11	11	11

Table 4: Node Failure Restoration Results Using Consolidated Approach

Scenario	Perf. Metric	Path Res	Link Res	Combination
New Jersey Single Node Failure N01 221ch. lost	Time msec	368	396	301
	Level	100%	81.9%	81.9%
	# of Msgs	16	13	15
	(# of Paths)	28	15	14
New Jersey Single Node Failure N03 16ch. lost	Time msec	107	120	126
	Level	100%	100%	100%
	# of Msgs	12	9	25
	(# of Paths)	6	4	4
New Jersey Single Node Failure N04 52 ch. lost	Time msec	263	209	209
	Level	100%	100%	100%
	# of Msgs	33	28	11
	(# of Paths)	7	4	4
New Jersey Single Node Failure N05 148 ch. lost	Time msec	337	273	345
	Level	100%	90.5%	100%
	# of Msgs	18	18	18
	(# of Paths)	24	21	21
New Jersey Single Node Failure N06 55 ch. lost	Time msec	171	140	148
	Level	100%	100%	100%
	# of Msgs	11	9	9
	(# of Paths)	8	5	5
New Jersey Single Node Failure N08 197 ch. lost	Time msec	454	474	422
	Level	100%	100%	100%
	# of Msgs	17	17	17
	(# of Paths)	31	26	25
New Jersey Single Node Failure N11 52 ch. lost	Time msec	242	182	182
	Level	100%	100%	100%
	# of Msgs	14	14	14
	(# of Paths)	8	7	7

Table 5: Centralized vs. Distributed Network Restoration (Link Failure Situations)

Scenario	Perf. Metric	Path Res	Link Res	Combination	Two Prong	FITNESS	RREACT
New Jersey Single Link Failure N01 - N02 74 ch. lost	Time msec	275	246	309	482	1096	582
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	312	—	318	343	252
	# of Msgs	9	12	13	126	136	78
New Jersey Single Link Failure N01 - N03 71 ch. lost	Time msec	258	214	162	120	654	195
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	196	—	160	160	160
	# of Msgs	10	7	12	106	64	73
New Jersey Single Link Failure N01 - N05 53 ch. lost	Time msec	189	218	218	138	645	214
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	160	—	116	157	116
	# of Msgs	12	7	11	95	61	63
New Jersey Single Link Failure N02 - N03 53 ch. lost	Time msec	199	206	230	321	724	741
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	134	—	186	261	189
	# of Msgs	11	10	13	97	101	149
New Jersey Single Link Failure N04- N05 16 ch. lost	Time msec	208	174	151	107	343	207
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	237	—	48	48	48
	# of Msgs	10	12	12	89	30	107
New Jersey Single Link Failure N04 - N08 59 ch. lost	Time msec	356	209	190	595	1127	594
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	262	—	222	278	222
	# of Msgs	12	13	13	157	112	108
New Jersey Single Link Failure N05 - N08 81 ch. lost	Time msec	376	467	576	273	1756	402
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	232	—	210	289	204
	# of Msgs	11	12	12	144	200	79

Table 5: Centralized vs. Distributed Network Restoration (Link Failure Situations)

Scenario	Perf. Metric	Path Res	Link Res	Combination	Two Prong	FITNESS	RREACT
New Jersey Single Link Failure N06 - N08 47 ch. lost	Time msec	434	333	193	146	1004	301
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	147	—	141	141	141
	# of Msgs	10	10	13	127	101	94
New Jersey Single Link Failure N07 - N08 41 ch. lost	Time msec	277	365	336	140	672	228
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	143	—	123	123	123
	# of Msgs	12	9	12	102	86	68
New Jersey Single Link Failure N08 - N11 64 ch. lost	Time msec	411	417	249	475	1827	739
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	301	—	215	239	233
	# of Msgs	12	13	13	133	197	114
New Jersey Single Link Failure N09 - N11 65 ch. lost	Time msec	340	348	174	141	748	592
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	—	257	—	189	254	189
	# of Msgs	10	12	12	99	88	90

Table 6: Centralized vs. Distributed Network Restoration (Node Failure Situations)

Scenario	Perf. Metric	Path Res	Link Res	Combination	Komine*
New Jersey Single Node Failure N01 221ch. lost	Time msec Level # of Msgs (# of Paths)	368 100% 16 28	396 81.9% 13 15	301 81.9% 15 14	1445 57% 1413 14
New Jersey Single Node Failure N04 52 ch. lost	Time msec Level # of Msgs (# of Paths)	263 100% 33 7	209 100% 28 4	209 100% 11 4	2025 100% 1647 7
New Jersey Single Node Failure N05 148 ch. lost	Time msec Level # of Msgs (# of Paths)	337 100% 18 24	273 90.5% 18 21	345 100% 18 21	2337 91% 1633 21

*: This is our simulation result of the Komine Algorithm [KCMOS90].

Appendix A. The path traffic pattern of the New Jersey Net used for the simulation.

BW	PATHS
21	N01-N02
21	N02-N01
45	N02-N03
45	N03-N02
14	N01-N03
14	N03-N01
35	N01-N04
35	N04-N01
4	N03-N05
4	N05-N03
17	N01-N05
17	N05-N01
12	N06-N05
12	N05-N06
64	N04-N05
64	N05-N04
31	N05-N09
31	N09-N05
54	N05-N08
54	N08-N05
6	N05-N07
6	N07-N05
11	N06-N08
11	N08-N06
45	N04-N08
45	N08-N04

BW	PATHS
24	N05-N11
24	N11-N05
40	N09-N10
40	N10-N09
8	N10-N11
8	N11-N10
24	N08-N11
24	N11-N08
7	N07-N08
7	N08-N07
18	N01-N08
18	N08-N01
10	N01-N06
10	N06-N01
16	N09-N08
16	N08-N09
65	N09-N11
65	N11-N09
25	N04-N09
25	N09-N04
6	N01-N05-N07
6	N07-N05-N01
16	N01-N04-N09
16	N09-N04-N01
9	N01-N08-N09-N10
9	N10-N09-N08-N01

BW	PATHS
10	N01-N05-N11
10	N11-N05-N01
4	N02-N03-N05-N04
4	N04-N05-N03-N02
10	N02-N01-N05
10	N05-N01-N02
9	N02-N01-N06
9	N06-N01-N02
5	N02-N01-N08-N07
5	N07-N08-N01-N02
10	N02-N01-N08
10	N08-N01-N02
10	N02-N01-N06-N05-N08-N09
9	N09-N08-N05-N06-N01-N02
10	N02-N01-N04-N09-N10
10	N10-N09-N04-N01-N02
4	N02-N03-N05-N11
4	N11-N05-N03-N02
10	N03-N01-N04
10	N04-N01-N03
9	N03-N01-N06
9	N06-N01-N03
4	N03-N05-N07
4	N07-N05-N03
9	N03-N01-N06-N05-N08
9	N08-N05-N06-N01-N03
9	N03-N01-N06-N08-N09

BW	PATHS
9	N09-N08-N06-N01-N03
10	N03-N01-N05-N09-N10
10	N10-N09-N05-N01-N03
10	N03-N01-N08-N11
10	N11-N08-N01-N03
9	N04-N08-N05-N06
9	N06-N05-N08-N04
5	N04-N08-N07
5	N07-N08-N04
10	N05-N11-N10
10	N10-N11-N05
9	N06-N08-N07
9	N07-N08-N06
9	N06-N05-N09
9	N09-N05-N06
9	N06-N08-N09-N10
9	N10-N09-N08-N06
9	N06-N08-N11
9	N11-N08-N06
5	N07-N08-N09
5	N09-N08-N07
5	N07-N08-N11-N10
5	N10-N11-N08-N07
5	N07-N08-N11
5	N11-N08-N07
11	N08-N11-N10