Fault Recovery Routing in Wide Area Packet Networks

Wei-Ping Wang,* David Tipper *†
Department of Information Science and Telecommunications
University of Pittsburgh, Pittsburgh, PA 15260
email: tipper@tele.pitt.edu, voice: (412) 624-9421, fax: (412) 624-2788

Bjørn Jæger‡
Department of Informatics
Molde College
Molde, Norway

Deep Medhi*
Computer Science and Telecommunications Program
University of Missouri - Kansas City
Kansas City, MO 64110

Abstract

In this paper, we consider the routing problem for traffic restoration after a failure in a virtual circuit packet switched wide area network. We assume source node routing of virtual circuits. A major factor on network performance after a failure is the transient congestion period that results from restored virtual circuits attempting to send out the backlog of packets accumulated for retransmission since the failure. Standard routing algorithms based on minimizing the steady state network delay may not be appropriate for rerouting the calls after a failure, in view of the transient congestion resulting from the packet backlog. Here we study alternative routing schemes to control the congestion after a failure. One approach is to use Minimum Hop routing to restore connections in order to ensure that the number of nodes directly effected by the rerouting is limited to a minimum. This method attempts to isolate and restrict the physical area of the congestion occurring due to the rerouting from a failure. An alternate approach is to distribute the potential congestion caused by the rerouting among the various links in the network, to take advantage of residual network capacity. This method allows the backlog of traffic to be potentially spread out over the various links in the network in order to not degrade the performance seriously at any particular link. This Load Distribution routing can be either across all the possible routes between the source-destination nodes or across all the links leading out of the source nodes. A precise formulation of the fault recovery routing problem is given and the implementation of the various algorithms is discussed. The results of a simulation study of a 10 node network comparing the performance of the different routing algorithms is presented.

*supported in part by National Science Foundation grant NCR-9506652
†Author for Correspondence
‡supported by Norwegian Research Council Project No: 110912/410

I Introduction

Due to the rapidly growing demand for information transfer, such as voice, data, and video across communication networks, the need for reliable communication service has become increasingly important. This has lead to a growing interest in the design of survivable networks as well as studies of fault recovery techniques to be employed in the event of a network outage (such as a cable cut, node failure or line card failure). Several survivability techniques [6, 21] have been discussed in the literature to minimize the effect of failures, common ones being multiple homing nodes and users, trunk diversity, digital cross connect systems, and self healing ring architectures.

The topic of fault recovery in the event of a network outage in ATM networks has received special attention recently [1, 3, 11, 13, 12, 21] . This work focuses primarily on facility (transmission) level restoration or virtual path layer restoration, and primarily for a link failure (such as a fiber cable cut). At the facility layer most of the work has been on the provisioning of spare network capacity in the form of SONET rings and digital cross-connect systems to provide recovery from single link failures in backbone networks. Note that this spare capacity is unused during normal network operation. In the area of virtual path restoration several different algorithms for fault recovery have been proposed. These algorithms differ by how the locus of rerouting and the rerouting procedure are specified. The locus of rerouting determines which network node in the virtual path is responsible for rerouting. For example in [7] the node adjacent to a failed link is responsible for rerouting, where as in [11] the source node of the virtual path decides the new route. Several different route selection procedures have been proposed ranging from preselected backup routes to dynamic routing selection based on the path load.

Here we focus on fault recovery routing at the individual virtual circuit level in connection oriented wide area packet switched networks. In many networks it may not be cost effective to provide for facility level restoration (e.g. network is based on leased capacity from various vendors, all of whom may not have facility restoration), or facility level restoration may not be applicable in certain failure scenarios (e.g. line card failure). Also, virtual path level restoration may not be able to connect all VCs (in them) in certain failure scenarios since it is rerouting a bundle of connections simultaneously, and the spare bandwidth needed can be higher than available. Hence a multi-layer fault recovery procedure is needed wherein a procedure of individual virtual circuit restoration is possible, especially when the virtual path level or transmission network level can not address a particular failure. Furthermore, fault recovery at the virtual circuit level has the advantages of making maximum use of network resources since the demand is being restored in small units of bandwidth and it is easy to provide priority for individual connections.

A weakness of virtually all of the literature on both survivable network design and traffic recovery procedures is that they are conducted using steady state analysis even though a transient or nonstationary congestion period will occur in the network after a failure. Our studies [19] indicate that for packet switched networks the dominant

factor on network performance after a failure is in fact this transient or nonstationary congestion period triggered by the backlog of packets needing retransmission after a failure. The importance of the transient congestion has been independently verified by Kobza [9]. In [9] he develops a lower bound on the length of the transient following traffic restoration after a failure in a queueing model of a window flow control virtual circuit in a packet switched network. Kobza's results taken along with [19] indicate several realistic situations where steady state assumptions are inappropriate. Note that a study showing the need for transient analysis in a circuit switching network fault recovery context is given in [10]. Hence, traffic restoration techniques designed and evaluated via steady state analysis may not make optimum use of network resources after a failure.

Consider an arbitrary connnection oriented packet switched wide area network such as an ATM network or an IBM NBBS network [8]. We assume that the network uses virtual circuit (VC) service to transport packets and source node routing of the virtual circuits. In source node routing, each network node maintains a database of the network topology and determines the route through the network for all virtual circuits originating at the node. The PNNI routing proposed by the ATM Forum for switched VC routing adopts this approach [22]. The impact of a failure on the network performance will depend on a complex interaction of several factors, some of which are the location and type of fault, network topology, network load, error control procedures, congestion control mechanisms and the routing algorithm. Certain aspects of the generic effects of failures on the behavior of the network can be determined by analyzing the general steps of the network in reacting to failures.

Consider the specific scenario of a link failure. In the event of such a failure we assume that all the virtual circuits using the failed device are disrupted and need to be reconnected if possible. We assume that the source nodes for the virtual circuits that were traversing the failed device are responsible for the restoration of the affected virtual circuits. The reconnection of the virtual circuits takes place only after a time delay which consists of the time take to detect the failure and set alarms, the time to disseminate knowledge of the failure to the fault recovery mechanism, the time to reroute and setup the circuit. In order to provide service continuity it is recommended that this reconnection time occur within two seconds for a voice connection and ten seconds for data connections [6]. During the time delay taken to restore the connection, packets/cells will be lost from the disrupted virtual circuits and depending on the application they may require retransmission from the source. These retransmitted packets/cells can create a backlog of at the traffic source that can result in congestion in the network. As noted in [19], congestion control schemes are not entirely effective in preventing congestion after a major failure since the overload at network nodes is mainly due to the rerouted virtual circuits needing to simultaneously work off their backlogs.

In the framework proposed in [19] for studying link failures, the source nodes of disrupted virtual circuits are called 'primary nodes' and links which emanate from a primary node are referred to as primary links. After a failure, congestion can occur at a primary node due to virtual circuits being rerouted across a particular link at

that node. As each virtual circuit is rerouted, it starts transmitting its entire backlog along its access link into the primary node. The link buffer at the primary node, being of a finite size, can quickly become congested. Any cell arriving at the network link queue and finding the buffer full is dropped. These cells may need to be retransmitted from the source depending on the actual application. Note that a dropped cell may trigger a packet retransmission at a higher level consequently the number of retransmitted cells may be larger then the number dropped. These retransmissions add a positive feedback to the source, further worsening the congestion. Thus, the cell loss rate at the network node can become high, exceeding the grade of service (GOS) level, possibly until the backlogs of each of the restored virtual circuits is completely transmitted. A device failure will typically result in several primary nodes with each having many virtual circuits to restore and a critical issue in the restoration is the path chosen for rerouting. We classify other nodes affected by the transient congestion into secondary and tertiary nodes, where secondary nodes are nodes which act as relay nodes for rerouted virtual circuits and tertiary nodes are nodes which handle traffic that shares a common link with the rerouted traffic. Links emanating from these nodes are referred to as secondary and tertiary links respectively (see [19] for further details). The number of secondary and tertiary nodes that occur after a failure will depend largely on the restoration routing algorithm and network topology. As discussed in [19], the congestion after a failure will start at the primary nodes and spread to the secondary and tertiary nodes.

The standard routing algorithms are normally based on minimizing the steady state network delay and such algorithms may be inappropriate for rerouting the calls after a failure, since at this time, the congestion is a paramount issue. Our preliminary study [2] of several algorithms has indicated that depending on the routing algorithm, the transient behavior can be noticeably different. In this paper, we present an optimization formulation of the rerouting problem by considering residual capacity in the network as well as the decision on whether or not to reconnect a disrupted VC. Our formulation allows us to consider several routing schemes in a unified framework. Note that after a failure many virtual circuits will simultaneously need restoration, thus we formulate the restoration problem as a bandwidth packing problem. This formulation is based on precise information on the network link status and the decision is done in a centralized manner. In an actual implementation in a network, source-node based routing makes decision in a distributed manner based on delayed information about network link status (due to periodic update) – we describe how this can be done. In a network simulation environment, we consider the solution from the optimization model as well as the solution due to distributed implementation to observe their differences. Through extensive simulation under different network load conditions, we present some interesting results on the inter-relation between acceptance (or denial) of reconnection of affected VCs and transient clearance of packet backlog. The rest of the paper is organized as follows: in section 2, we present the optimization model for the rerouting problem, while in section 3, we present the network performance study for solution of the optimization model as well as for distributed implementation.

II The Rerouting Problem

Consider an arbitrary connection oriented packet switched network consisting of n nodes represented by the ordered set N. Let L denote the ordered set of network links and C_ℓ the capacity of link $\ell \in L$. Let $\bar{s}(\ell)$ and $\bar{e}(\ell)$ denote the start and end nodes of directed link ℓ. Also we let A denote the set of source-destination pairs ($A \subset N \times N$); note that at maximum there are $n \times (n-1)$ elements in A. For each $(i,j) \in A$. let $\mathcal{P}_{i,j}$ denote the number of possible loop free paths from i to j that are considered. Note that $\mathcal{P}_{i,j}$ maybe less then the number of possible paths that can be found in the topology. A common approach to limiting the paths considered is to use all possible paths subject to a maximum hop count [11]. Note that all the terms defined above can be determined from the topology of the network under study.

As virtual circuits arrive to the networks they are assigned a unique virtual circuit identifier v. Let d_v denote the equivalent bandwidth (i.e., capacity) of virtual circuit v with source node $sn(v)$ and destination node $dn(v)$. Let K denote the set of virtual circuits disrupted by the failure of a specific device (e.g., link) or group of devices. Let the residual capacity of each link $\ell \in L$ after a failure be denoted by R_ℓ. In general a certain portion of the bandwidth γ_l of each link ℓ maybe reserved for control traffic and to absorb fluctuations in traffic. Thus the usable free capacity for rerouting of connections on link ℓ is given by $R_\ell - \gamma_\ell$. The number of possible paths for restoring disrupted virtual circuit k is denoted by P_k where $P_k \in \mathcal{P}_{sn(k),dn(k)}$. We define the path indicator decision variable to be y_{kj}; this is set to 1 if the jth path is choosen for restoring virtual circuit k ($y_{kj} = 0$ otherwise). The cost of rerouting a disrupted virtual circuit k on it's jth path is defined by the path cost W_{kj}. Similarly we define link indicator functions $\delta_{kj}^{\ell} = 1$ if the jth path for virtual circuit k uses link ℓ ($\delta_{kj}^{\ell} = 0$ otherwise) and the cost of using link ℓ is given by v_ℓ.

Since there may not be enough residual capacity in the network to restore all the virtual circuits that are affected by the failure we define a set of variables corresponding to rejecting each disrupted virtual circuit k. The decision variable for rejection is given by x_k and is set to one if the connection is not restored, otherwise it is zero. The cost of failing to restore (i.e., *rejecting*) the connection is given by O_k. Given the definitions/notation above, the rerouting of virtual circuits $k \in K$ can be determined in a optimal fashion by solving the following optimization problem for y_{kj}, x_k

$$\min_{y_{kj},x_k} \quad \sum_{k=1}^{K} \sum_{j=1}^{P_k} W_{kj} y_{kj} + \sum_{k=1}^{K} O_k x_k \tag{1}$$

$$s.t. \qquad \sum_{j=1}^{P_k} y_{kj} + x_k = 1, \qquad k = 1,2,...K \tag{2}$$

$$\sum_{k=1}^{K} \sum_{j=1}^{P_k} d_k \delta_{kj}^{\ell} y_{kj} \leq R_\ell - \gamma_\ell, \quad \ell = 1,2,...L \tag{3}$$

$$y_{kj} = 0 \text{ or } 1, \quad x_k = 0 \text{ or } 1. \tag{4}$$

In (1), the first term in the objective function is the cost of choosing the jth path for virtual circuit k. The right most term in (1) is the cost of rejecting virtual circuit k. The set of constraints (2) specifies that either a path or connection rejection must be choosen for each virtual circuit. The set of constraints (3) requires that the load offered to each link be feasible in terms of any residual capacity and subject to any additional imposition γ_ℓ to maintain link stability. The last set of constraints (4) ensures that only one route is choosen for a virtual circuit. Note that the optimization problem defined by (1) - (4) is a special case integer programming problem termed a General Assignment Knapsack problem and several algorithms exist for solving this class of problems [17].

The formulation above allows us to consider various restoration scenarios by specifying an appropriate path, link and rejection cost. Normally the path cost is the sum of link costs on the path and this can be used to define several interesting routing options by specifying the link cost. Thus, we can write

$$W_{kj} = \sum_{\ell=1}^{L} \delta_{kj}^{\ell} v_\ell \tag{5}$$

where δ_{kj}^{ℓ} values are determined by the source and destination node of the virtual circuit k and the links used, and is known ahead of time for a given number of paths for all source destination pairs in the network.

We consider the behavior of four different routing algorithms and their effects on call blocking, route selection and network congestion. Standard routing algorithms are normally based on minimizing the steady state network delay. Minimum Delay (MD) routing schemes have been proposed for virtual circuit based networks using a delay cost on each network link proportional to the derivative of the link queueing delay based on an M/M/1 type model as discussed in [4]. However, rerouting connections based on minimizing the steady state network delay may not be appropriate for restoring calls after a failure, in view of the transient congestion. We propose alternative routing schemes to control the congestion after a failure. One approach is to use Minimum Hop (MH) routing to restore connections in order to ensure that the number of nodes directly effected by the rerouting is limited to a minimum. This method attempts to isolate and restrict the physical area of the congestion occurring due to the rerouting from a failure. An alternate approach is to distribute the potential congestion caused by the rerouting among the various links in the network, to take advantage of residual network capacity. This method allows the backlog of traffic to be potentially spread out over the various links in the network in order to not degrade the performance seriously at any particular link. One approach to load distribution is to spread the load across all the possible routes between source-destination node pairs by selecting the route with the maximum residual capacity. We call this approach Load Distribution Among Paths (LDAP). An alternate approach to load distribution is to concentrate on the primary nodes (i.e., source nodes) and attempt to balance the load on all the links leading out of a primary node. This technique is termed Load Distribution among Primary Links (LDPL). Link costs v_ℓ for these four different routing algorithms are given below.

1. Minimum Delay (MD) - assuming steady state M/M/1 queueing model for each link

$$v_\ell = \frac{C_\ell}{(R_\ell)^2}, \ \ \forall\ \ell \tag{6}$$

2. Minimum Hop (MH)

$$v_\ell = 1, \ \ \forall\ \ell \tag{7}$$

3. Load Distribution Among Paths (LDAP)

$$v_\ell = -R_\ell \ \ \forall\ l \tag{8}$$

4. Load Distribution among Primary (source) Links (LDPL) - weight is given to both the status of the primary link as well as the number of hops to reach the destination

$$v_\ell = \frac{C_\ell}{R_\ell} \text{ where } \ell : sn(k) = \bar{s}(\ell) \qquad v_\ell = 1, \ \ \forall \ell : sn(k) \neq \bar{s}(\ell) \tag{9}$$

The cost for rejecting the kth virtual circuit can be specified in several ways. Obviously if the rejection cost O_k is large for all virtual circuits then VCs will be rejected only when there is no way to reroute all VCs under the constraints (in particular (3)). On other hand, if the rejection cost is too small, then all VCs will be rejected. Here we present results for the case where all VCs have the same large rejection cost (e.g., $O_k = 1000\ \forall\ k$). An analysis of the affect of varying the rejection cost on connection blocking and the routes selected is given in the extended version of this paper [20].

The fault recovery restoration routing optimization problem is defined by (1)-(4) together with a specific link cost formula (i.e., one of (5) - (9)) and a rejection cost. For the solution to the optimization problem we adopt the branch and bound approach discussed in [17], which was implemented in MATLAB. The approximate branch and bound algorithm given in [17] for the General Assignment Knapsack problem has an overall time complexity of $O(PKlogK + P^2)$. Here K denotes the number of VCs affected by the failure and P denotes the average number of paths for rerouting the affected VCs (i.e., $P = \frac{1}{K}\sum_{k=1}^{K} P_k$). Several numerical computational experiments are given in [20] confirming the complexity cost.

The solution to the optimization problem above requires global knowledge of the network after the failure, and hence can only be implemented in a centralized fashion. For example, a central network control center could obtain knowledge of the failure and the current link cost in the network and solve the above optimization problem

and distribute the set of optimum recovery routes to the nodes. This is difficult to implement in practice since a quick solution is required for rerouting.

We propose a suboptimal distributed rerouting approach that is consistent with ATM PNNI routing [22]. Each node maintains topology database of the network containing the cost of using each link in the network. Also, to speed up the rerouting computation each node maintains a precomputed set of routes between each source destination pair that is determined from the topology of the network and is restricted by a hop count limit. The database of link cost is updated periodically with each node notifying the other network nodes of its current cost at the update times using a flooding approach. Also, asynchronous updating of the link cost occurs whenever a link utilization changes by an amount exceeding a predefined threshold. Routing is accomplished at connection setup time by the source node of the connection using it's local cost database and the set of predefined paths to select the minimum cost path from the source to destination nodes. A connection setup request is then forwarded along the minimum cost path to ensure that the neccessary resources exist along the path and to modify routing tables. If the necessary resources are available (e.g., spare capacity $\geq$ equivalent bandwidth of VC request) at every node on the path, then the destination node will issue a connection acceptance message back to the source reserving the resources and the connection begins working. When the resources are unavailable along the path selected, the source node must try an alternate route or block the VC.

The fault recovery routing schemes proposed above will easily fit into this format with a simple change (if necessary) in the link cost used for calculation of the route. Note that an interesting issue arises in fault recovery routing that does not occur in the normal operation of the network namely that a primary node (i.e., source node of VC needing restoration) will typically have many VCs to reroute all at once. Thus the order in which they are processed will affect the routes used and, thus, connection blocking. Here we consider two simple approaches to ordering the connections for restoration processing: 1) increasing amount of bandwidth and 2) decreasing amount of bandwidth. In the first scheme the connections are sorted in increasing order of the VC bandwidth required (i.e., increasing order of d_k). This approach should result in the lowest call blocking rate. The second approach is to sort the VCs in decreasing order of the VC bandwidth required, thus maximizing the amount of demand restored.

III Performance Evaluation

A simulation based performance study was conducted to evaluate the restoration routing schemes defined above in terms of VC call blocking, the amount of congestion created (both length of time and spatial distribution) and the paths selected. In order to study the transient congestion, we adopt the nonstationary simulation methodology of [14] to observe the behavior of the average number in the queueing system versus time for all links in the network. The basic approach is to run the simulation a number of times and average the quantities of

interest across the ensemble of independent runs at a particular time instant. Many such points may be obtained at different time instants and the behavior of the system studied as a function of time.

The ten node, $n = 10$, network with $|L| = 42$ links shown in Figure 1 was chosen for study. The connectivity of the network topology as measured by the average node degree D in the network ($D = |L|/n$) is 4.2 and is in the range of many existing networks [11]. The topology was selected in part due to the large number of alternate paths between any pair of nodes in the network. A simulation model of the network was developed using SLAM focusing on a comparative analysis of the routing schemes during fault recovery under a common set of simplified assumptions. The routing algorithms were implemented in the simulation in a distributed fashion with each node maintaining a local cost table of the network links with periodic cost updates. A hop count limit of four (i.e., four links) was used in the simulation to restrict the number of feasible paths and speed up the route selection. In the simulation model the capacity of each link was one, all packets/cells are of fixed length one, and the buffer size at each link is 20 (the system size is 21). Thus, the service rate of each link is one packet/cell per simulation time unit. The packet/cell arrival processes for each VC were modeled as independent Poisson processes with fixed mean rate. A maximum steady-state link utilization threshold of 0.9 was used for all links (i.e., $\gamma_l = 0.1 \; \forall \; l$). The time to detect a failure and begin notification of the source nodes was assumed to be one second. It is assumed that all dropped packets/cells need retransmission from the source and the resulting backlogs at the virtual circuit sources was determined using the normalization approach given in our earlier work [19]. Several different simulation cases were performed with different network loading and failure scenarios. Here we report sample results with additional results given in [20].

In the experimental results shown here 50 virtual circuits between various source-destination nodes were working in the network before the link 2-4 failure which affects $k = 9$ VCs from source node 9 with an aggregate load of 0.84 (see [20] for details). Two cases of background loading are discussed here: LITE - average link utilization in the network of 0.334 before failure, and HEAVY - average link utilization in the network of 0.746 before failure. The traffic rates d_k, number of possible restoration paths P_k and the paths selected by the four rerouting schemes for the 9 affected VCs in the LITE load case are given in Table 1. In the LITE load case the routes selected by solving the optimization problem of Section II are identical to those determined by the distributed scheme regardless of ordering of connections in the distributed scheme. From Table 1 it is clear that the routes selected by the four schemes are as expected from their definition. Specifically, the MH scheme picks 1 and 2 hop paths resulting in the fewest number of links used. Where as, the LDAP algorithm uses four hop paths for all VCs and affects the greatest number of nodes. The LDPL scheme balances the load on the outgoing primary node links better than MH and uses fewer links then the LDAP and MD schemes. The MD approach prefers shorter paths while avoiding heavily loaded links.

As one would expect in the LITE load case since spare capacity is plentiful all VCs are restored regardless of the routing scheme used. Thus by the standard metrics for survivability such as %call blocking or %demand

restored, all the schemes are equivalent. However the transient congestion induced in the network illustrates marked differences between schemes. That the amount of congestion varies with the scheme adopted can clearly be seen in Figure 2 which shows for each routing scheme the average number in the system versus time after traffic restoration for selected network links. Consider the plot for link 9-1 shown in Figure 2. One can see that after the restoration of VCs the buffer at link 9-1 quickly saturates under all four schemes, but the amount of time the link is overloaded varies with the routing scheme. Specifically, the MH approach congest link 9-1 much longer than the other techniques. The results shown in Figure 2 for each scheme are the result of an ensemble average of 400 simulation runs. During the congestion periods there is very little variance in the results obtained from the simulation and the resulting confidence intervals on the simulation results are very narrow and have been left off the plots for the sake of clarity.

By examining plots [20] similar to Figure 2 for all the links in the network effected by traffic restoration, we can identify the links that become congested and the length of time congestion last for the different routing strategies. Here a link was considered congested if the average number in the queueing system at a link exceeded the network buffer size (i.e., 20), and congestion was deemed over when the number in system fell below 5. The right hand column of the row marked LITE in Table 6 shows the number of links used in traffic restoration for each scheme and the number of links that become congested is given in parentheses. Also listed in the table is the average congestion time which is the mean of the time each link is congested for those that become congested. Similarly, the maximum congestion time given in the table is the length of time for all links that become congested to clear.

The same experiments, plotting and tabulation procedure was repeated for the HEAVY load case with the results given in Tables 2-6. The routes selected with the four rerouting algorithms when connections are processed for restoration in distributed fashion in order of increasing amount of bandwidth, in order of decreasing amount of bandwidth, and the centralized solution of the the optimization problem of Section II are given in Tables 2, 3, and 4 respectively. Notice that the routes selected and the VCs rejected for the same routing algorithm (e.g., MD) are different in all three implementations. Tables 5 and 6 show the survivability metrics and congestion metrics for the three different implementations of the four routing schemes. Note that for the HEAVY-OPTIMIZATION results shown, the routes were computed ahead of time with the optimization problem and loaded into the simulation when node 9 detected the failure. As one would expect the call blocking of the centralized optimization approach is less and the demand restored is more than that resulting from the distributed implementations. In comparing the two approaches to ordering the connections in the distributed implementations one can clearly see the trade-off in call blocking versus demand restored. Also as expected the implementations (decreasing order and optimization) that restore more demand generally result in higher lengths of congestion time for each scheme.

Considering Tables 5 and 6 together (and other results in [20]), one can see the the following. For lite to medium loads when all demand is recoverable, the MH approach results in fewest number of links becoming congested.

However, MH results in the largest time period of congestion. In contrast, the load distribution schemes resulted in a greater number of links becoming congested but for a much shorter time period. Hence, at loads where the demand restored is the same, there is a clear trade off between constricting the region effected by congestion versus congesting a larger region for a possibly smaller amount of time. For the HEAVY load case the results are reversed with the LDAP scheme congesting the network for the maximum amount of time, however it provides a lower call blocking. Selecting which scheme is preferred will depend on the need to restrict the effects of failures locally to the region of the failure, the need to reconnect as many virtual circuits as possible and the need to minimize congestion in the network. Obviously, additional work needs to be conducted to precisely quantify the tradeoffs among the schemes. Also, since the Minimum Delay scheme is optimal under steady state conditions, one may want to restructure the paths of the virtual circuits after the congestion period is over.

IV Summary

In this paper we have presented a optimization based formulation of the virtual circuit fault recovery routing problem in wide area packet networks. Routing algorithms suitable for traffic restoration after a failure are proposed and their implementation discussed. Specifically we studied, the Minimum Hop scheme – which ensures that the number of nodes directly effected by the rerouting is minimized, the Load Distribution Among Paths scheme – which picks the source-destination route that has the maximum residual capacity, and the Load Distribution among Primary Links scheme – which balances the load on the outgoing links of node that must reroute several connections. The relative performance of the routing algorithms (along with Minimum Delay routing) was studied using a simulation model of a 10 node sample network. We have shown that the choice of routing algorithm has significant effect on network performance during fault recovery. Futhermore, it was illustrated that in addition to standard network survivability metrics, an important criterion is controling the transient network congestion occurring after restoration both spatially and in terms of duration.

References

[1] A.Banerjea, C.J.Parris and D.Ferrari, "Recovering Guaranteed Performance Service Connections from Single and Multiple Faults," *Technical Report TR-93-066*, International Computer Science Institute, Berkeley, CA, 1993.

[2] K. Balakrishnan, D. Tipper and D. Medhi, "Routing Strategies for Fault Recovery in Wide Area Packet Networks," *Proceedings of 1995 IEEE Military Communications Conference (MILCOM'95)*, San Diego, CA, pp. 1139-1143, November 1995.

[3] M. Ball, A. Vakhutinsky, P. Chimento, L. Gun and T. Tedijanto, "Distributed Call Rerouting in Multiclass Broadband Networks," *Journal of Network and System Management*, Vol. 3, No. 5, 1995

[4] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1992.

[5] Robert Doverspike and Brain Wilson, "Comparison of Capacity Efficiency of DCS Network Restoration Routing Techniques" *Journal of Network and System Management*, Vol. 2, No. 2, 1994

[6] W. E. Falconer, "Service Assurance in Modern Telecommunications Networks," *IEEE Comm. Mag.*, vol. 28(6), pp:32-39, June 1990.

[7] H. Fuji and N. Yoshikai, "Restoration Message Transfer Mechanism and Restoration Characteristics of Double Search Self-Healing ATM Network,"*IEEE Journal of Selected Areas in Communications*, Vol. 12, January, 1994.

[8] G. Marin, C. Immanuel, P. Chimento and I. Gopal, "Overview of the NBBS Architecture," *IBM Systems Journal*, Vol. 34, No. 4, 1995.

[9] J. Kobza, *"The Significance of Transients Following Failures and Repairs in Packet-Switched Networks"*. PhD thesis, Virginia Polytechnic Institute and State University, February 1993.

[10] D. Logothetis and K. Trivedi,"The Effect of Detection and Restoration Times on Error Recovery in Communications Networks," *Proceedings of IEEE Milcom 95*, San Diego, CA, Nov. 1995.

[11] M. Herzberg, S. J. Bye, and A. Utano, "The Hop- Limit Approach for Spare-Capacity Assignment in Survivable Networks",*IEEE/ACM Transactions on Networking*, Dec, 1995.

[12] Special Issue, "Integrity of Public Telecommunications Networks," *IEEE Journal of Selected Areas in Communications*, Vol. 12, January, 1994.

[13] K.R. Krishnan, R. Doverspike, and C. Pack, "Improved Survivability with Multi-Layer Dynamic Routing" *IEEE Communications Magazine*, July, 1995.

[14] W. Lovegrove, J. Hammond, and D. Tipper, "Simulation Methods for Studying Nonstationary Behavior of Computer Networks," *IEEE Jrnl. Sel. Areas Comm.*, vol. 8:1696-1708, 1990.

[15] D. Medhi and S. Sankarappan, "Impact of a Transmission Facility Link Failure on Dynamic Call Routing Circuit-Switched Networks under Various Circuit Layout Policies," *Jrnl. of Net. and Sys. Mgmt.*, Vol. 1, pp. 143-169, 1993.

[16] D. Medhi and R. Khurana, "Optimization and Performance of Network Restoration Schemes for Wide-Area Teletraffic Networks," *Journal of Network and Systems Management*, Vol. 3, No. 3, pp. 265-294, September 1995.

[17] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations* John Wiley, New York, 1990.

[18] D. Mitra and J. B. Seery, "Comparative Evaluations of Randomized and Dynamic Routing Strategies for Circuit-Switched Networks," *IEEE Trans. Comm.*, Vol. 39, pp. 102-115, 1991.

[19] D.Tipper, J.Hammond, S.Sharma, A.Khetan, K.Balakrishnan, and S. Menon, "An Analysis of the Congestion Effects of Link Failures in Wide Area Networks," *IEEE Jrnl. Sel. Areas Comm.*, vol. 12:179-192, 1994.

[20] W. Wang, D. Tipper, B Jaeger, and D. Medhi, "Fault Recovery Routing in Wide Area Packet Networks," technical report, University of Pittsburgh, August, 1996. available by anonymous ftp to *violet.tele.pitt.edu*.

[21] T.H.Wu, *Fiber Network Service Survivability*. Artech House, Boston, Mass., 1992.

[22] The ATM Forum Technical Committee*Private Network-Network Interface Specification version 1.0 (PNNI 1.0)*. ATM Forum Document af-pnni-0055.000, March 1996.

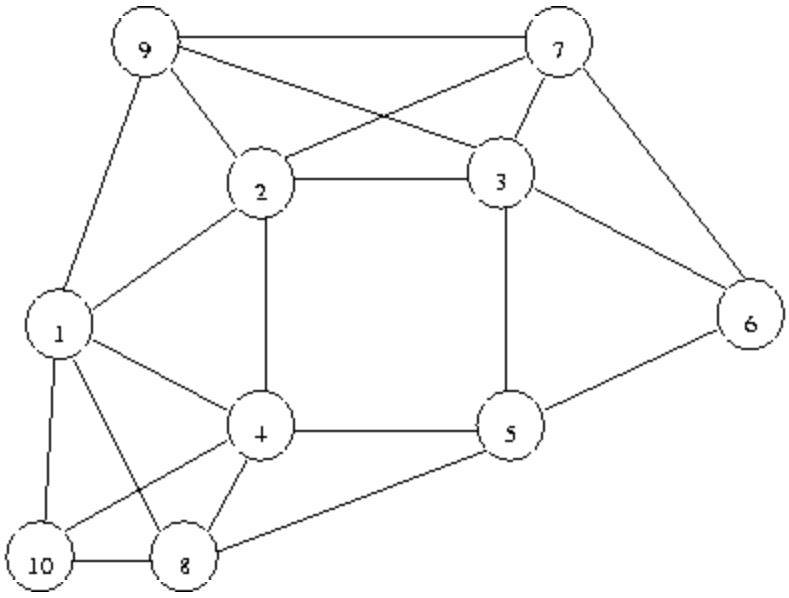

Figure 1: Ten node network topology

Table 1: Comparison of paths selected in the LITE load case

d_k	P_k	MD	MH	LDAP	LDPL
0.15	12	9–1–10	9–1–10	9–3–5–8–10	9–1–10
0.14	17	9–1–8	9–1–8	9–1–10–4–8	9–1–8
0.12	17	9–1–4	9–1–4	9–2–3–5–4	9–1–4
0.11	21	9–3–5	9–3–5	9–2–1–8–5	9–3–5
0.09	17	9–3–5–4	9–1–4	9–7–6–5–4	9–1–4
0.08	8	9–2–1	9–1	9–7–3–2–1	9–2–1
0.06	21	9–3–5	9–3–5	9–1–3–5	9–3–5
0.05	17	9–7–6–5–8	9–1–8	9–1–4–5–8	9–2–1–8
0.04	17	9–7–6	9–3–6	9–1–2–7–6	9–7–6

Table 2: Comparison of paths selected in the HEAVY-INCREASING case

d_k	P_k	MD	MH	LDAP	LDPL
0.04	17	9–2–7–6	9–3–6	9–2–7–3–6	9–3–6
0.05	17	9–1–8	9–1–8	9–2–1–4–8	9–2–1–8
0.06	21	9–3–5	9–3–5	9–2–7–3–5	9–3–5
0.08	8	9–1	9–1	9–3–7–2–1	9–2–1
0.09	17	9–2–1–4	9–2–1–4	9–2–3–5–4	9–3–5–4
0.11	21	9–3–5	9–3–5	9–2–7–3–5	9–2–3–5
0.12	17	Rejected	9–7–6–5–4	9–1–10–4	9–1–4
0.14	17	Rejected	Rejected	Rejected	Rejected
0.15	12	Rejected	Rejected	Rejected	Rejected

Table 3: Comparison of paths selected in the HEAVY-DECREASING case

d_k	P_k	MD	MH	LDAP	LDPL
0.15	12	9–1–10	9–1–10	9–3–5–4–10	9–1–10
0.14	17	9–2–1–8	9–2–1–8	9–2–1–4–8	9–2–1–8
0.12	17	9–3–5–4	9–3–5–4	9–1–10–4	9–3–5–4
0.11	21	9–3–5	9–3–5	9–2–7–3–5	9–3–5
0.09	17	Rejected	Rejected	Rejected	Rejected
0.08	8	Rejected	Rejected	Rejected	Rejected
0.06	21	9–2–7–6–5	9–3–6–5	9–2–7–6–5	9–2–3–6–5
0.05	17	Rejected	Rejected	9–1–10–4–8	Rejected
0.04	17	9–3–6	9–7–6	9–2–7–3–6	9–2–7–6

Table 4: Comparison of paths selected in the HEAVY-OPTIMIZATION case

d_k	P_k	MD	MH	LDAP	LDPL
0.15	12	Rejected	9–2–1–10	Rejected	Rejected
0.14	17	9–3–5–4–8	Rejected	9–2–1–4–8	9–3–5–4–8
0.12	17	9–1–4	9–1–4	9–1–10–4	9–1–4
0.11	21	9–3–5	9–7–6–5	9–2–7–6–5	9–3–5
0.09	17	9–2–1–4	9–3–5–4	9–2–3–5–4	9–2–1–4
0.08	8	9–1	9–1	9–1	9–1
0.06	21	9–2–7–6–5	9–3–5	9–2–7–3–5	9–3–6–5
0.05	17	9–2–1–8	9–3–5–4–8	9–3–5–4–8	9–2–1–8
0.04	17	9–2–7–6	9–3–6	9–2–7–3–6	9–3–6

Table 5: Comparison of %call blocking and %demand restored

Network Load - connection order	% Call Blocking MD/MH/LDAP/LDPL	% Demand Restored MD/MH/LDAP/LDPL
LITE	0.0/0.0/0.0/0.0	100/100/100/100
HEAVY-INCREASING	33.3/22.2/22.2/22.2	51/66/66/66
HEAVY-DECREASING	33.3/33.3/22.2/33.3	74/74/80/74
HEAVY-OPTIMIZATION	11.1/11.1/11.1/11.1	82/83/82/82

Table 6: Congestion metrics (Congested: number in system ≥ 20, congestion over: number in system ≤ 5)

Network Load - connection order	Avg Time Congested MD/MH/LDAP/LDPL	Max Time Congested MD/MH/LDAP/LDPL	Links used (Congested) MD/MH/LDAP/LDPL
LITE	483/1300/518/507	850/2250/600/1150	13(3)/7(2)/23(5)/10(3)
HEAVY-INCREASING	630/2988/1890/2600	700/10000/3000/4000	9(5)/12(4)/16(5)/10(4)
HEAVY-DECREASING	2520/3533/80000+/3100	4700/4000/80000+/4000	12(5)/12(3)/16(5)/13(5)
HEAVY-OPTIMIZATION	7880/3960/47500+/18900	12700/8400/80000+/39000	12(5)/13(5)/16(2)/11(5)

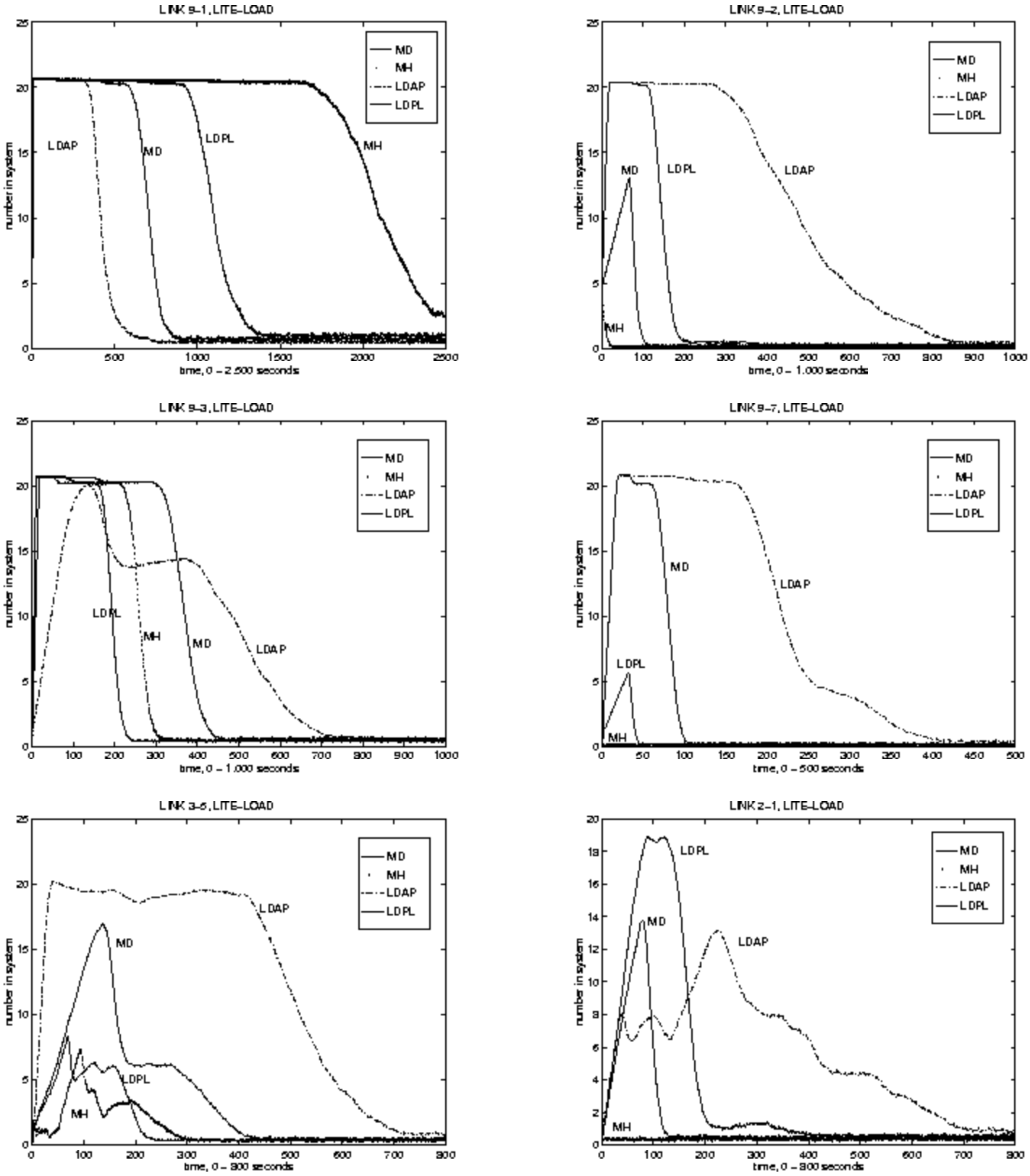

Figure 2: Number in system versus time for selected links in LITE-LOAD case