Effect of Unreliable Nodes on QoS Routing

Swapna S. Gokhale and Satish K. Tripathi
Bourns College of Engineering
University of California
Riverside, CA 92521
{swapna,tripathi}@cs.ucr.edu

Abstract

A number of QoS routing algorithms have been proposed to address the dual objective of selecting feasible paths through the network with enough resources to satisfy a connections' QoS request, while simultaneously utilizing network resources efficiently. However, these routing algorithms and the guarantees they provide do not consider the possibility of node and link failures. The failure of a node or a link along a path can disrupt the continuity of an on-going session and potentially terminate the session. Hence the problem of QoS routing should be extended to incorporate reliability and fault-tolerance requirements. In this paper we study the impact of unreliable nodes on QoS routing. We describe a scheme to restore the flows that are disrupted due to node failures to alternate paths. Two prioritized restoration policies, one to maximize the number of disrupted flows that can be restored, and the other to maximize the disrupted demand that can be restored are also presented. We conduct extensive simulations to evaluate routing performance in the presence of node failures and repairs, as well as the performance of the restoration scheme. Our results indicate: i) when the availability of the nodes is beyond a certain threshold, the routing as well as the restoration performance is comparable where the number of failures in one is twice the number of failures in the other, ii) the percentage of the disrupted flows and the percentage of the disrupted demand that can be restored successfully increase with decreasing network load, and iii) prioritized restoration policies are effective under heavy network load conditions, and their effectiveness decreases with decreasing network load.

1 Introduction

One of the most important factors contributing to the success of the existing Internet is the robustness offered by its connectionless datagram delivery, that is every packet is treated separately, and routing decisions are made on a per packet basis, based on the destination address in the packet header (may be coupled with the type-of-service field in the packet header) [9], and taking into account the state of the network at the time of the decision. Various intra-domain routing protocols such as distance-vector RIP [8] and link-state OSPF [10], and inter-domain routing protocols such as BGP [12] are presently employed for this purpose. The connectionless datagram delivery in the current Internet facilitates rerouting along a failed network element by making a local decision at a node which is adjacent to the failed element. The source and/or the destination of the packet need not be involved in making this rerouting decision.

However, the inherent limitation of the present best-effort Internet architecture to support applications requiring real-time guarantees has triggered a number of activities along different dimensions. Network protocols and infrastructure are undergoing fundamental changes driven by the need to support performance sensitive distributed and multimedia applications. An essential component of QoS architecture is QoS based routing. QoS routing attempts to satisfy the dual objective of selecting feasible paths through the network which have the highest potential of satisfying the requested QoS while utilizing network resources efficiently. A number of routing algorithms [5, 6, 14, 16, 17] have been developed to satisfy this dual objective. Typically these routing algorithms and the guarantees they provide do not consider the possibility of node and link failures. The failure of a node or a link along a path can disrupt the continuity of an on-going session and potentially terminate the session. As a result, the problem of QoS routing should be extended to incorporate reliability and fault-tolerance requirements.

The effect of unreliable nodes and links on QoS routing can be addressed using a two-pronged strategy: a proactive approach in which an additional "reliability" constraint can be provided to the path selection algorithm. The path selection algorithm can be extended to select paths using a constraint of reliability in addition to the other constraints such as bandwidth, delay, jitter etc. This approach is feasible only if a reliability metric is available or can be obtained for the nodes and the links under consideration. Though proactive approaches can help minimize the impact of unreliable nodes and links by selecting reliable paths based on the requirements of the flow, they are by themselves not sufficient to guarantee perfect reliability through the life-time of a call. Reactive approaches, which essentially consist of rapid and efficient restoration of the flows disrupted by the failure of a network element to alternate paths then become crucial. A number of issues such as the scope, granularity and the duration of restoration, and the effect of the topology and the load on the network, and the spare capacity necessary to achieve a certain degree of restoration then merit investigation.

In this paper, we assess the impact of unreliable nodes on QoS routing. We outline the various design choices that can be employed to restore the flows that are disrupted due to the node failures, and study a combination of some of these choices. We define performance metrics to assess the efficiency of the restoration scheme in terms of the percentage of the disrupted flows that can be restored successfully, and the percentage of the disrupted demand that can be restored successfully. We evaluate the efficiency of the restoration scheme by conducting extensive simulations, under different network topologies as well as load. Two prioritized restoration policies, one to increase the percentage of the disrupted calls restored successfully, and the other to increase the percentage of the demand restored successfully are also outlined, and their effectiveness is evaluated. Our results indicate: i) when the availability of the nodes is beyond a certain threshold, the routing performance as well as the performance of the restoration scheme is comparable where the number of failures in one is twice the number of failures in the other, ii) the percentage of disrupted calls that can be restored successfully as well as the percentage of the disrupted demand that can be restored successfully increase as the network load decreases, and iii) prioritized restoration policies are effective under heavy network load conditions, and their effectiveness decreases with decreasing network load.

The balance of the paper is organized as follows: Section 2 describes the network model used in the experiments. Section 3 outlines the failure and repair model for the nodes, and the procedure followed to restore the flows disrupted in the event of a node failure. Section 4 defines the performance measures used to assess the efficiency of the flow restoration scheme described in Section 3. Section 5 presents the simulation environment to obtain the performance measures defined in Section 4 and provides some discussion. Section 6 concludes the paper.

2 Network model

The network model used in the analysis consists of the following aspects:

- *Network topologies:* Figure 1 shows the topologies used. The topology on the left hand side in the figure is the well known `isp` topology, used in various simulation based studies of QoS routing [2], and reflects the topology of a nationwide ISP. The `isp` topology is not a very well connected topology. The one on the right hand side is an artificial `mesh` like structure, and is used to study the effect of a well connected topology with multiple equal hop paths. The links are assumed to be symmetric, that is, have equal bandwidths in both the directions. The propagation delay of the links is assumed to be 1 $msecs$.
- *Traffic model:* All traffic in the network is assumed to require bandwidth guarantees, that is, there is no best effort traffic in the network. A single request is characterized by the 3-tuple, that is, $(source, destination, demand)$. Requests are assumed to arrive independently at each node according to a Poisson distribution, and the destination node is chosen uniformly among all the remaining nodes in the network. Request duration is assumed to be exponentially distributed. Requested bandwidth is uniformly distributed in the range of $[L, U]$. This is the homogeneous or uniform traffic model where the average traffic between any two pairs of nodes is the same.
- *Path selection:* Recent simulation studies indicate that from among the many heuristics proposed for routing requests with bandwidth requirements, shortest (with respect to the number of links on the path) path heuristics [6, 7] perform better than the widest path heuristics [17]. The width of a path, also called bottleneck capacity, is defined as the minimum available bandwidth over all the links in the path. A widest-shortest path selection algorithm is employed, where paths are computed as follows: Links that have insufficient available bandwidth for the request that is being routed are pruned from the network topology before the path is computed. Then the minimum hop count paths between the source and the destination are discovered, and the widest one is used to route the request. If there are more than one widest-shortest paths, a path is chosen at random with a probability that is weighed according to the bandwidth available on the first hop of the path. An explicit route to the destination is determined by the source node when a request is received. Path selection is done on-demand, that is, a path is computed for every incoming request.

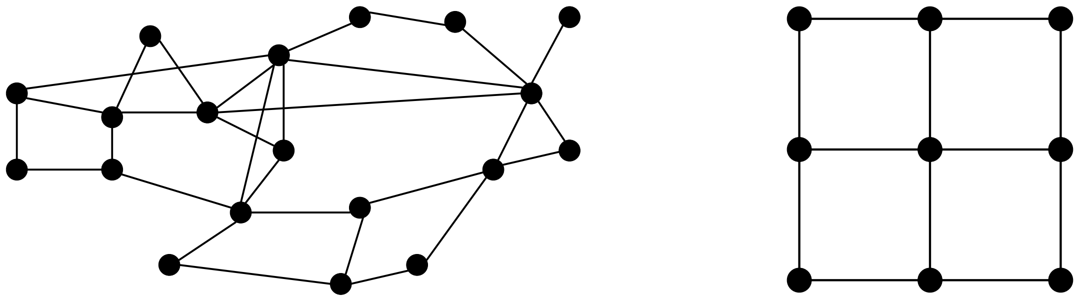

Figure 1. Network topologies

- *Reservation and admission control:* Once a route is determined by the routing algorithm at the source of the request, a control packet is sent along the route determined, to reserve the bandwidth along the route. The admission control is a simple check if the available bandwidth on the link, is sufficient to support the new request without causing a detrimental effect on the already accepted flows. If the admission control succeeds at all the intermediate nodes, then the source is notified of the successful reservation, else, the node at which admission control fails notifies the source of reservation failure. High level admission control [2] which aims at preventing routing of flows over alternate longer paths if such routing results in an inefficient utilization of network resources, is not employed in this study. This is mainly because restoration of disrupted flows may require routing over longer paths, especially in sparsely connected topologies where multiple equal hop paths are not available, and a higher level admission control scheme could potentially preclude from the disrupted flows from being restored. An alternative approach could be to prohibit a new connection request from reserving resources over longer paths resulting in an inefficient utilization of network resources, while relaxing this policy when a disrupted flow is being restored. We will explore the benefits and cost tradeoffs of such a scheme in a future study.

- *Routing updates:* We assume that each node generates periodic updates of its link states, with the period randomized by a small amount to avoid synchronization of routing updates [3], and these are flooded throughout the network.

3 Node failures and flow restoration

In this section we describe the failure model adopted for the nodes, and the semantics of the flow restoration that is performed when a node fails. We assume that the nodes failures follow an exponential distribution with rate λ, and repairs also follow an exponential distribution with rate μ. Thus, the steady state availability A of the node is given by:

$$A = \frac{\mu}{\lambda + \mu} \tag{1}$$

Exponential distribution to describe failure and repair behavior is chosen here because of its common and widespread use in modeling these type of phenomenon. To the best of our knowledge there is no data available regarding the failure as well as repair processes of the nodes, based on which a guided inference about the distribution as well as the parameters of the distribution can be drawn.

We assume that the failure of a node implies that it is operating in a degraded mode, where it is incapable of creating and maintaining reservations for flows requiring guarantees, but continues to process routing and other control messages. Upon failure, the node asynchronously generates an update message, which announces zero available bandwidth on all of its links. This update is then flooded through the network in a similar fashion as the routine periodic update messages, so that eventually all the nodes are informed about the failure. Similarly, upon repair, the node generates an update message announcing for each of its links, the available bandwidth equal to the capacity of the link. This update is then propagated through the network, so all the other nodes eventually learn about the repair of the node.

When a node fails, the requests currently routed through the node will have to be restored to alternate paths with sufficient capacity to satisfy their demand. Towards this end, we first classify the flows routed through a given node X into three categories: i) A *primary flow* is a flow that originates at node X. In other words, node X is the source of the flow. ii) A *secondary flow* is a flow that terminates at node X. In other words, node X is the destination of the flow. iii) A *tertiary flow* a flow for which node X serves as a transit node, that is the flow neither originates or terminates at the node. In other words, node X is neither the source nor the destination of the flow.

Figure 2 shows the primary, secondary, and tertiary flows with respect to node 3. It is clear that upon the failure of a node, only the tertiary flows through the node can be attempted to be restored, whereas the disrupted primary and

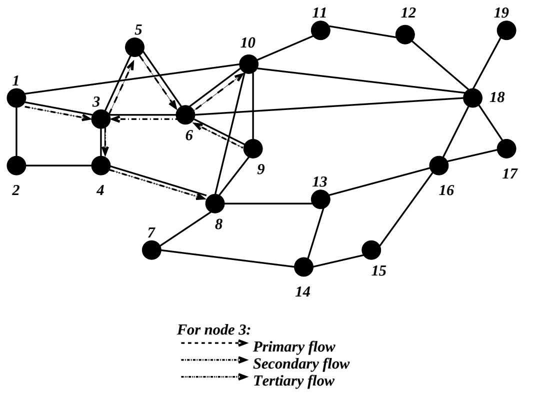

Figure 2. Classification of flows into primary, secondary and tertiary

secondary flows have to be terminated, and the resources held by them on the other links released. Also, all the primary flow requests arriving at a failed node, as well as the secondary flow requests choosing the failed node as the destination, will be blocked. For a particular node, the number of primary flow requests blocked depends on the arrival rate of the calls at the node, whereas the number of secondary flows blocked depends on the arrival rate at the other nodes, and the probability of choosing the particular node as a destination.

There are various restoration strategies that can be employed based on the scope, granularity, and duration of the restoration phase, some of which we outline below:

- *Scope of restoration:* The disrupted path can be restored around the failed node to the destination, (local restoration), or can be completely restored from the source to the destination (source restoration) [11]. An intermediate scheme such as restoring the path from a node which is upstream to the failed node, but not necessarily the source are also possible [4]. The distance of the upstream node (using some metric such as the number of hops) from where the flow is restored to the failed node, would determine the scope of restoration. One of the primary advantages of local restoration is that it effectively contains the impact of the failure within a small local area surrounding the element. This is not the case with source restoration, where nodes distant from the failed network element are also affected. Also, local restoration can be performed faster than source restoration, which can be a crucial factor for some real-time applications. However, local restoration typically requires higher capacity than source restoration (for the same level of restoration) [11].

- *Granularity of restoration:* The disrupted paths could be restored one at a time or in bundles, that is, an aggregate path could be restored. Clearly, the latter approach has the advantage of minimizing the restoration overhead, since the procedure needs to be invoked only once for all the paths constituting the bundle. However, this approach may lead to inefficient use of network resources and may require higher bandwidth than the former approach, where the demand is being restored in smaller units of bandwidth.

- *Duration of the restoration phase:* Recovering from a network failure involves restoring a large number of disrupted paths at almost the same time; a situation which does not occur very commonly in normal operation [13]. The path restoration process for the disrupted flows can proceed either sequentially or in a parallel fashion [15]. In case of sequential restoration, a disrupted path is either restored or the request is blocked before attempting to restore the next one. This increases the duration of the restoration phase, but may improve the percentage of the disrupted flows restored successfully, because of the routing updates that may occur during the restoration interval. Various policies intermediate to either a completely sequential or a completely parallel restoration are possible, which tradeoff between the duration of the restoration phase, and the percentage of the disrupted calls that can be restored.

From the design space generated by the above alternatives, we choose the following:

- The source of the flow attempts to restore the disrupted flow.

- The disrupted flows are restored on a per-flow basis.

- A semi-sequential restoration is studied, which involves the following two steps: the routing algorithm determines an alternate path for the disrupted flow, and initiates the reservation process over this alternate path. The restoration of the next flow begins right after. Thus, the path selection, and resource reservation proceeds in a parallel fashion. Once, the restoration of a flow terminates successfully or unsuccessfully (due to reservation or routing failure), the resources held by the flow on the unfailed portions of its original path are released. We note that some of the disrupted flows will be blocked because of the resources that are currently held by the flow itself, and efficient mechanisms to account for these resources either in the routing phase or resource reservation phase need to be developed, to avoid the problem of "stepping on one's shadow". One of the possibilities could be to release the resources held by the disrupted flows prior to selecting the alternate paths. However, this could increase the duration of the restoration phase to an extent which could be intolerable by flows with real-time constraints.

The other design alternatives and their combinations will be explored in a future study.

The disrupted and new flows, are treated in exactly the same fashion in the path selection phase. No resources are reserved for restoring the disrupted flows. Although crucial, we don't attempt to assess detection and restoration times in this experiment, and is a subject of further study. We assume a fixed detection time of $10\ msec$, that is, an attempt is made to restore the disrupted tertiary flows $10\ msec$ after the node fails. The value of $10\ msecs$ was chosen so that there is enough time for the information about the failure of the node to propagate to the remaining nodes in the network.

4 Performance measures

In general, the routing performance depends on the percentage of incoming requests and the percentage of the offered demand that can be routed successfully, whereas the performance of the restoration scheme depends on the percentage of the disrupted calls and the percentage of the disrupted demand that can be restored successfully. To assess the routing performance in the event of node failures and repairs, as well as the performance of the restoration scheme, we define the following four metrics:

- Total acceptance ratio (TA): This is the ratio of the total number of flows routed successfully (original and disrupted) to the total number of flow requests.

$$TA = \frac{\#\ routed\ (orig.\ +\ disrup.)}{\#\ of\ requests\ (orig.\ +\ disrup.)} \qquad (2)$$

  Since the original and the disrupted requests are treated in an identical manner during the path selection process, the total acceptance ratio is defined to account for both.

- Total bandwidth acceptance ratio (TBA): This is the ratio of the bandwidth routed successfully (original and disrupted) to the total bandwidth requested (original and disrupted).

$$TBA = \frac{Demand\ routed\ (orig.\ +\ disrup.)}{Demand\ requested\ (orig.\ +\ disrup.)} \qquad (3)$$

- Restoration acceptance ratio (RA): This is the ratio of the number of disrupted flows routed successfully to the total number of disrupted flows.

$$RA = \frac{\#\ disrup.\ flows\ restored}{\#\ disrup.\ flows} \qquad (4)$$

- Restoration bandwidth acceptance ratio (RBA): This is the ratio of the demand restored successfully to the total demand to be restored.

$$RBA = \frac{Demand\ restored}{Total\ demand\ to\ be\ restored} \qquad (5)$$

The first two performance metrics, namely, total acceptance ratio (TA), and the total bandwidth acceptance ratio (TBA) are geared towards evaluating the routing performance in the presence of node failures and repairs. The third and the fourth metric, namely, restoration acceptance ratio (RA), and restoration bandwidth acceptance ratio (RBA) are directed towards assessing the effectiveness of the restoration scheme.

Initially, we assume that the disrupted flows are restored in random order, typically the order is determined by the entries in the flow table of the failed node. The random order policy also serves as the baseline for comparing the following two policies, which restore the disrupted flows in a prioritized fashion:

- The first policy ranks all the disrupted flows in an increasing order of demand. Intuitively, this should increase the number of disrupted flows that can be restored successfully, and hence the restoration acceptance ratio (RA).

- The second policy ranks all the disrupted flows in a decreasing order of demand. Intuitively, this should increase the the demand that can be restored successfully and hence the routing bandwidth acceptance ratio (RBA).

5 Simulation experiments

The performance measures for the restoration scheme outlined in Section 3 are obtained using simulation. The simulations are carried out using MaRS [1], which has been enhanced to include the widest-shortest path QoS routing algorithm [2]. In this section, we describe the evaluation environment which was used to obtain the performance measures defined in Section 4.

The link capacities in case of both the topologies are determined in such a manner that the call blocking probability when the call interarrival time at each node is $10\ secs$ for `isp` topology, and $7.5\ secs$ for the `mesh` topology, is in the range of $2-5\%$. These blocking probabilities are determined in the absence of failures. This dimensioning method results in the link capacities of $45-300$ Mbit/sec for `isp` topology, and 100 Mbit/sec for `mesh` topology. The mean request duration is set to 3 minutes. This value is chosen to represent the average duration of a phone call; it does not necessarily reflect the duration of new types of multimedia calls, such as video conferencing, but to the best of our knowledge there are no data available for that type of calls. Request sizes are uniformly distributed between 64 kbit/sec (typical bandwidth requirement of a voice call), up to a maximum of 6 Mbit/sec (typical bandwidth requirements of a high quality MPEG-2 video). Each node generates an update of the available bandwidth on its interfaces periodically every 10 $secs$. The value of 10 $secs$ is randomized by a small value to avoid the synchronization of periodic routing messages [3]. Each simulation was carried out for 100 $hours$ and the performance measures were computed at the end of 100 $hours$. The simulations were repeated 20 times for each scenario, and the average performance metrics for these 20 runs are reported here.

Initially we study the effect of availability (Equation (1)) on the total acceptance ratio (TA) and the total bandwidth acceptance ratio (TBA) for both `isp` and `mesh` topologies. Towards this end, we varied the availability of all the nodes from 99.5% to 95.0%, using two different configurations of failure and repair rates for the nodes, as shown in Table 1. These values were chosen so that they are significantly higher than the values of the other parameters such as interflow spacing, and propagation delay of links. To the best of our knowledge there is no documented data available, which either provides the failure and the repair rates of the nodes, or from which this information can be inferred. The interflow spacing is assumed to be 10 $secs$. The failure and the repair rates in case of the second configuration are twice the respective failure and repair rates of the first configuration. This implies that on an average there will be twice as many failures observed in case of the second configuration, as compared to the number observed in case of the first configuration. However, the average total down time or the average time a node is unavailable will be the same in both configurations. Thus, the number of times the restoration procedure is invoked in the second configuration is twice the number of times the restoration procedure needs to be invoked in the first configuration. Assuming that the network reaches a quasi-steady state between two successive failures, this implies that twice the number of flows have to be restored in the second configuration than the first.

Figure 3 shows the effect of availability on total acceptance ratio (TA) for both the `isp` and `mesh` topologies. In case of `isp` topology the total acceptance ratio (TA) is quite close in case of both the configurations, for availabilities of 99.5%, 98.5% and 96.5%. However, the total acceptance ratio (TA), in case of configuration #2 is lower than in case of configuration #1, when the availability of the nodes is 95%. A similar phenomenon is observed in case of the `mesh` topology, in fact, the total acceptance ratio (TA) is higher in case of configuration #2 than in configuration #1, when the availability of the nodes is 98.0%. Thus, for higher availabilities of the nodes the network seems to completely oblivious to the fact that there are twice as many number of failures, and hence approximately twice as many number of flows that need to be restored and effectively provides the same routing performance. This is perhaps important, since increasing the reliability of a node beyond a certain threshold (or equivalently reducing the failure rate of the node beyond a threshold) can turn out to be prohibitively expensive. A better and may be a price effective alternative without sacrificing the routing performance would then be to reduce the down time or the repair time of the node. Similar results are observed in case of total bandwidth acceptance ratio (TBA), and are not shown here due to space limitations.

Figure 4 shows the effect of availability on the restoration acceptance ratio (RA)) for `isp` topology. The restoration acceptance ratio (RA) is slightly worse for higher availabilities in configuration #2, but is better than configuration #1 for lower availability. One of the reasons for this phenomenon is that as the availability reduces, the down time of the nodes increases, due to which the number of primary and secondary flows blocked increases. Subsequently, the number of flows that are disrupted and hence need to be restored in the event of future failures are lower in the lower availability situation. Also, since the number of primary and secondary flows blocked increases when the availability is lower (for the same failure rate), the residual capacity in the network which is available for restoration increases. As a result, a higher number of disrupted flows can be restored, which manifests itself as a slight increase in the restoration acceptance ratio (RA). Similar observations are recorded for the restoration bandwidth acceptance ratio(RBA) for `isp` topology, and restoration acceptance ratio (RA) and restoration bandwidth acceptance ratio (RBA) for the `mesh` topology and are not shown here.

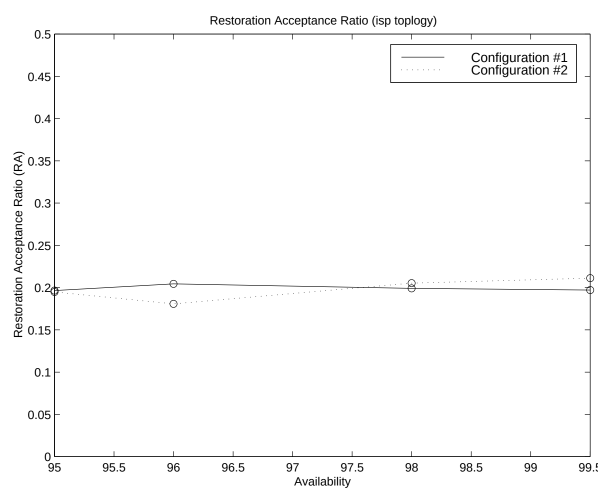

Figure 4. Effect of availability on RA for `isp` **topology**

Table 1. Failure and repair rates of nodes

Availability	Configuration #1		C onfiguration #2	
	Failure rate/hr	Repair rate/hr	Failure rate/hr	Repair rate/hr
99.5	0.01	2.00	0.02	4.00
98.0	0.01	0.50	0.02	1.00
96.5	0.01	0.24	0.02	0.48
95.0	0.01	0.19	0.02	0.38

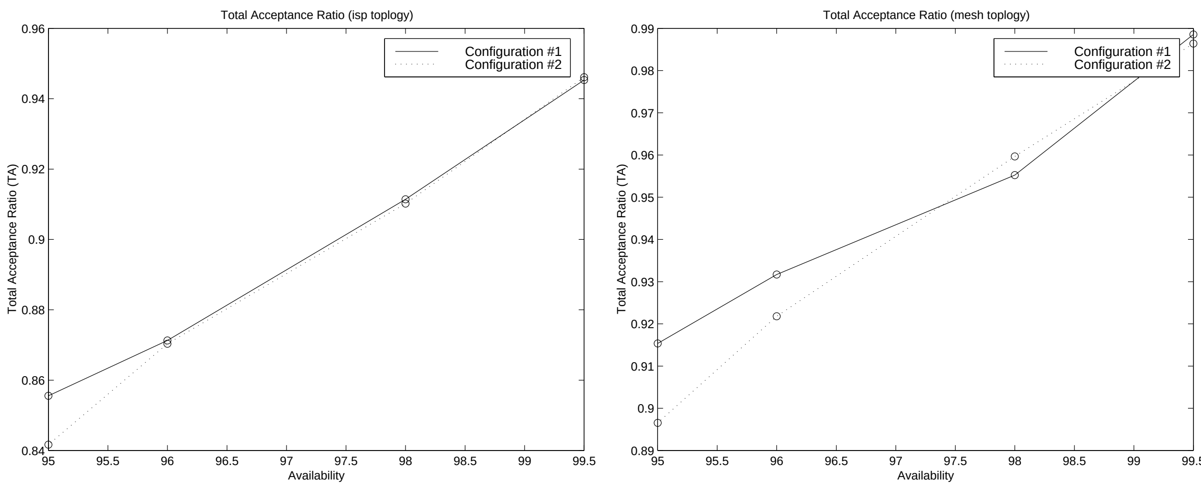

Figure 3. Effect of availability on TA for `isp` and `mesh` topologies

We then study the performance of the restoration scheme under varying conditions of network load. Decreasing the network load should mitigate the impact of node failures because of two reasons: i) A decrease in the network load causes an increase in the residual capacity in the network. Since the disrupted flows are restored using this residual capacity, an increase in the residual capacity has a positive influence on the percentage of the disrupted flows as well as the percentage of the demand that can be restored successfully. ii) The number of flows that are disrupted due to the failure of a node decreases as the network load reduces. As a result, the number of disrupted flows competing for the residual capacity upon a failure decreases, leading to a higher percentage of flows as well as demand that can be restored successfully.

The network load was controlled by changing the interflow spacing at each node. We varied the interflow spacing from 7.5 $secs$ to 80 $secs$ for the `isp` topology, and 4 $secs$ to 10 $secs$ for the `mesh` topology. These values were chosen so that the network is operated in a relatively high blocking region (about $10-15\%$) in the heavily loaded case, and almost zero or no blocking in the lightly loaded scenario. These base blocking probabilities were obtained in the absence of failures. To study the effect of network load in isolation, the availability of all the nodes is assumed to be 99.5%, using the failure and repair rates of configuration #1.

Figure 5 shows the effect of workload on the restoration acceptance ratio (RA) for `isp` topology. The restoration acceptance ratio (RA) increases as the load on the network decreases, and under extremely light load, there is enough residual capacity to restore all the disrupted tertiary flows, and the only flows that cannot be restored are primary and secondary. As a result, the restoration acceptance ratio (RA) does not increase significantly with a decrease in the network load, beyond a certain threshold of network load. Similar observations were recorded for the restoration bandwidth acceptance ratio (RBA), for both `mesh` and `isp` topologies, and restoration acceptance ratio (RA) for `mesh` topology and are not included here.

We then study the effect of prioritized restoration on the restoration acceptance ratio (RA), and restoration bandwidth acceptance ratio (RBA). Towards this end, we initially restore the disrupted flows as per increasing order of their demand, and later as per decreasing order of their demand. This is carried out for different network loads. To ensure similar conditions in order to facilitate fair comparison between the random and prioritized policies, we replayed the failure and repair events recorded during the experiments which restore the flows in random order, for pri-

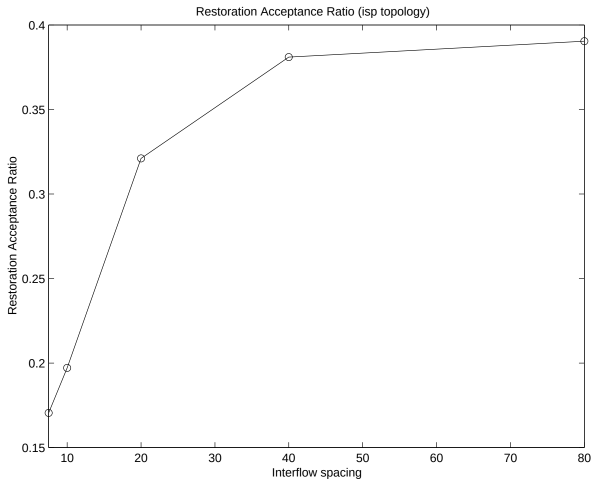

Figure 5. Effect of workload on RA for `isp` topology

oritized restoration. This was done for each of the 20 runs. The availability of all the nodes was set to 99.5%, using the combination of failure and repair rates as in configuration # 1. The results for the `isp` topology are shown in Figure 6. The restoration acceptance ratio (RA) is higher than the random restoration case when the disrupted flows are restored in an increasing order of demand, and the restoration bandwidth acceptance ratio (RBA) is higher than the random case, when the disrupted flows are restored in a decreasing order of demand. This effect is seen when the network is heavily loaded (lower interflow spacing). The performance of both random restoration and prioritized restoration is almost the same, in case of light network loads (higher interflow spacing). This is due to the fact that under light load conditions there is sufficient residual capacity to restore all the disrupted tertiary flows, and the priority of restoration does not impact the restoration acceptance ratio (RA), as well as the restoration bandwidth acceptance ratio (RBA). Also, the effect of prioritization on the restoration acceptance ratio (RA), and the restoration bandwidth acceptance ratio (RBA) is slightly more pronounced in the case of `mesh` topology, than in the case of `isp` topology. This could be because of the highly structured nature of the `mesh` topology, as opposed to the sparse connectivity of the `isp` topology, which offers multiple equivalent paths between almost every source-destination pair. The results of the `mesh` topology are not included here due to space constraints.

6 Conclusions

In this paper we discuss the effect of failures and repair dynamics of nodes on QoS routing, and outline the various design choices to enable an efficient restoration of the flows that are disrupted due to such failures. We define four metrics, two to assess routing performance in the presence of node failures and repairs, and two to assess the performance of the restoration scheme. We conduct extensive simulations to obtain these performance measures under various network conditions as well as failure and repair configurations. Initially, we study the effect of availability on the various performance measures, and our results indicate that beyond a certain level of availability, the percentage of the flows routed successfully, is quite comparable, even though the number of failures in one scenario are twice as much the number of failures in the other. We then study the effect of network load on the percentage of disrupted flows and the demand that can be restored successfully, and as expected observe that as the network load decreases, the percentage of the disrupted flows that can be restored as well as the percentage of the demand restored increases. We then study two prioritized restoration policies, namely, restoring the disrupted flows in an increasing order of their demand, and restoring them in a decreasing order of their demand. The former is expected to increase the percentage of the disrupted flows that can be restored, whereas in the latter case, the percentage of the disrupted demand that can be restored is higher. Our results indicate, that the benefits of the prioritized restoration policies are higher under heavy network loads, since under light loads, there is enough residual capacity to restore all the disrupted flows, and prioritization does not yield any benefits.

The future research will include an analysis and comparison among other possible restoration schemes, as well as devising techniques to determine the amount of spare capacity that must be provisioned in the network to achieve a certain degree of restoration. The time required for the detection of a failure and subsequent restoration is also a crucial factor in determining the efficiency of a restoration scheme, since this will determine the amount of time the service of a flow will be disrupted. Realistic assessment of the detection as well as the restoration time, and policies to minimize them will also be explored.

References

[1] C. Alattinoglu, A. U. Shankar, K. Dussa-Zieger, and I. Matta. "Design and implementation of MaRS: A routing testbed". *J. of Internetworking Research and Experience*, 5(1):17–41, 1994.

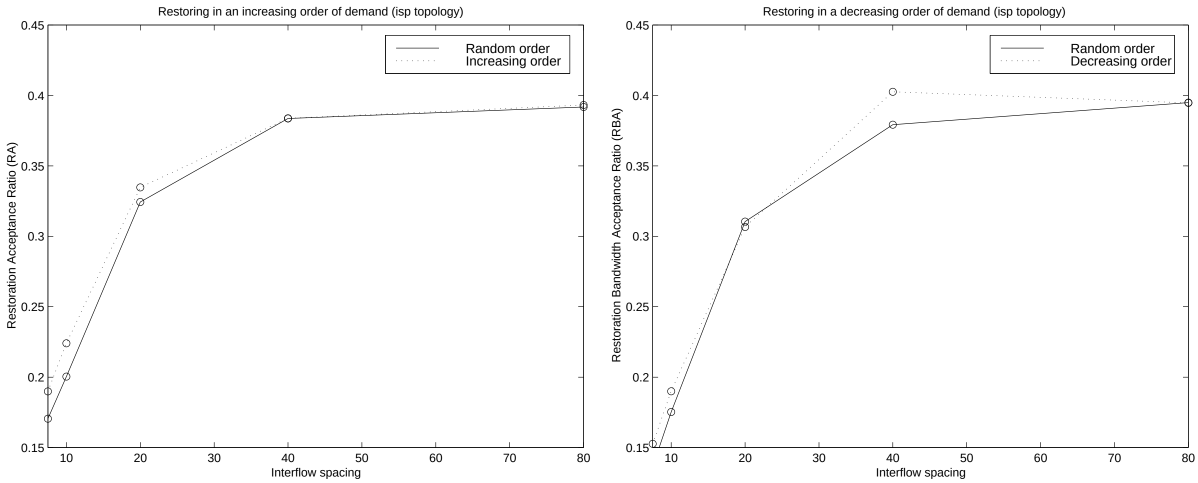

Figure 6. Effect of prioritized restoration policies on RA and RBA for `isp` topology

[2] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi. "Quality of Service routing: A performance perspective". In *Proc. of SIGCOMM*, pages 3–16, Vancouver, Canada, September 1998.

[3] S. Floyd and V. Jacobson. "The synchronization of periodic routing messages". *IEEE/ACM Transactions on Networking*, 2(2):122–136, April 1994.

[4] D. Gan, T. Li, R. Guerin, E. Rosen, and S. Kamat. "QoS path management with RSVP". In *Proc. of the 2nd IEEE Global Internet Mini-Conference*, Phoenix, AZ, November 1997.

[5] W. C. Lee, M. G. Hluchyj, and P. A. Humblet. "Routing subject to Quality of Service constraints in integrated communication networks". *IEEE Network*, pages 46–55, July/August 1995.

[6] Q. Ma and P. Steenkiste. "Quality of Service routing for traffic with performance guarantees". In *Proc. of IFIP Fifth International Workshop on QoS*, pages 115–126, NY, NY, May 1997.

[7] Q. Ma, P. Steenskiste, and H. Zang. "Routing high-bandwidth traffic in max-min fair share networks". In *Proc. of SIGCOMM*, pages 206–217, Stanford, CA, August 1996.

[8] G. Malkin. "RIP version 2". Request for Comments (Standards Track) RFC 2453, Internet Engineering Task Force, November 1998.

[9] I. Matta and A. U. Shankar. "Type-of-service routing in dynamic datagram networks". *IEEE Journal on Selected Areas in Communications*, 13(8):1411–1428, October 1995.

[10] J. Moy. "OSPF version 2". Request for Comments (Standards Track) RFC 2328, Internet Engineering Task Force, April 1998.

[11] K. Murakami and H. S. Kim. "Comparitive study on restoration schemes of survivable ATM networks". In *Proc. of INFOCOM '97*, Kobe, Japan, April 1997.

[12] Y. Rekhter and T. Li. "A Border Gateway Protocol 4 (BGP-4)". Request for Comments (Standards Track) RFC 1771, Internet Engineering Task Force, March 1995.

[13] M. P. Rumsewicz and D. E. Smith. "A comparison of SS7 congestion control options during mass call-in situations". *IEEE/ACM Trans. on Networking*, 3(1):1–9, February 1995.

[14] A. Shaikh, J. Rexford, and K. G. Shin. "Dynamics of Quality-of-Service routing with inaccurate link-state information". In *Proc. of IEEE Intl. Conf. on Network Protocols (ICNP) '98*, October 1998.

[15] W. Wang, D. Tipper, B. Jaeger, and D. Medhi. "Fault recovery routing in wide area packet networks". In *Proc. of 15th Intl. Teletraffic Conference*, Washington, DC, June 1997.

[16] Z. Wang and J. Crowcroft. "Bandwidth-delay based routing algorithms". In *Proc. of GLOBECOM*, Singapore, December 1995.

[17] Z. Wang and J. Crowcroft. "Quality of Service routing for supporting multimedia applications". *IEEE Journal on Selected Areas in Communications*, 14(7):1228–1234, September 1996.