On Providing Survivable QoS Services in the Next Generation Internet

Anotai Srikitja and David Tipper
Dept. of Information Science and Telecommunications
University of Pittsburgh
Pittsburgh, PA 15260 USA

Deep Medhi
Dept. of Computer Networking
University of Missouri - Kansas City
Kansas City, MO 64110 USA

Abstract— **In this paper we present a comparative study of two schemes to provide survivability for guaranteed QoS connections in a possible Next Generation Internet network architecture. In the first scheme a QoS connection is provided standby backup resources on a disjoint path by reserving resources on both the working and backup path. In order to reduce the amount of backup resources required a method for sharing backup resources when the working connections have disjoint routes has been included. In the second scheme a dynamic search for restoration resources is conducted over a preplanned set of alternate paths upon notification of a failure. A simulation based performance study shows that the first scheme results in much higher connection blocking under normal operations, slightly faster restoration times, and longer transient congestion times after fault recovery due to non-optimal backup routing.**

I. Introduction

The Next Generation Internet (NGI) will provide QoS-based services in addition to traditional best effort service. A certain set of users (e.g., military) can be expected to demand a cost effective level of fault tolerance for QoS-based services. Hence there is a need for techniques to ensure the survivability of certain services in NGI architectures. Note that some QoS services will likely follow fixed routes (e.g., Guaranteed Service class in IntServ model [17]) and thereby be subject to single point failures (e.g., link failure) as in circuit switched networks.

The subject of providing survivability in the face of failures has been extensively studied for circuit switched networks and recently for ATM networks [3], [9], [18]. This includes work on network design and capacity allocation, as well as work on traffic restoration/network management algorithms for fault recovery. A variety of survivability techniques have been proposed for circuit switched and ATM networks at the physical layer, logical layer and the traffic layer. At the physical layer most of the work focuses on SONET rings or automatic protection switching both of which involve provisioning idle spare capacity. At the logical and traffic layer several approaches for provisioning spare network capacity in mesh type topologies have been proposed along with traffic restoration techniques. It is well known that a mesh type topology together with traffic restoration techniques is more capacity efficient than physical layer approaches. The traffic restoration techniques typically differ in specification of the location of rerouting, the rerouting algorithm and the reservation or nonreservation of resources in the event of a failure. From the literature it is clear that no one approach is cost effective or optimal for all networks and a multi-layer survivability approach with a combination of techniques is suggested. Such survivability issues have received little attention in the context of NGI architectures as the focus thus far has been to develop a QoS based architecture.

In this paper we investigate two schemes for survivability applicable at both the traffic or logical layers in any packet switched network that supports explicit path establishment. In the first scheme a guaranteed QoS service is provided reserved standby backup resources on a path which is disjoint with the working path. In order to reduce the amount of backup resources required, the method for sharing backup resources proposed in [4] is adopted. In the second approach a dynamic search for restoration resources is conducted over a preplanned set of alternate paths upon notification of a failure. A comparative simulation based study of the performance of the two survivability schemes is presented. The study considered both steady state and transient network behavior and shows that the first scheme results in much higher connection blocking under normal operations, slightly faster restoration times and longer transient congestion times.

In the next section, we discuss NGI architectures and the two survivability schemes studied in detail. In Section III, we present the results of our simulation based performance evaluation. Lastly in Section IV we summarize our findings.

II. Survivability Schemes for NGI

Curently two different QoS-based services frameworks are being considered for the NGI namely: (1) the Integrated Service (IntServ) model and (2) the Differentiated Service (DiffServ) model. In the IntServ model, per connection QoS-based services are classified into three types: Guaranteed Service, Controlled-Load Service and Best-effort Service [17]. The Guaranteed Service class is intended to provide a guarantee of bandwidth, a end-to-end delay bound and no packet loss, to applications with a stringent real-time delivery requirement. The Controlled-Load service is aimed to be used for those classes of applications that can tolerate some amount of loss or delay, i.e. adaptive real-time applications. Controlled load service provides a loose guarantee of service on delay and packet loss. Best-effort Service corresponds to current Internet service in that no guarantees are made.

supported in part by NSF grant NCR 9506652 and DARPA under agreement No. F30602-97-1-0257

0-7803-5538-5/99/$10.00 © 1999 IEEE

The Differentiated Service approach provides QoS service to aggregrates of traffic instead of individual flows [11]. All packets entering the network are sorted into a different QoS service class and treated/forwarded differently. Three QoS service classes are proposed: Premium Service, Assured Service and Best-effort service. Premium Service will provide a guaranteed peak bandwidth service with a end-to-end delay bound. Assured Service offers an expected level of bandwidth with a statistical delay bound. Lastly, Best-effort service corresponds to current Internet service.

Both frameworks are being considered in the Multiprotocol Label Switching (MPLS) architecture [8] where the explicit routing of Label Switched Paths is supported. In the backbone of a NGI network, one would expect a large portion of the traffic to be an aggregation of intransit traffic which can be of IntServ or DiffServ type carried by MPLS virtual networks. The backbone network is expected to be engineered using explicit path routing where an aggregration of traffic flows within the same class, refered to as traffic trunk, is routed manually or dynamically through a specified path [10].

Here, we consider a NGI network architecture where explicit path routing is utilized, this is consistent with both the MPLS architecture and the IntServ model. For the sake of clarity we specify the model in terms of the IntServ model along with the Resource ReserVation Protocol (RSVP) [1] for connection set-up and tear-down signaling of QoS-services. We assume that the RSVP functionalities are extended to support explicit path reservation as discussed in [7]. This extension facilitates the management of QoS paths in setting up the path that meets the QoS of individual flows and maintaining or adjusting them in response to failures and changes in the network. Note that, this RSVP extension can be used with IP layer source-based routing [2] for QoS services.

We consider two approaches to support *survivable* QoS-based services in this architecture. In the first approach, any time a connection request arrives, a disjoint backup route is set up with reserved standby resources, along with the primary route (working path). The backup route is used only if the primary route fails. This approach was proposed in [4] along with a method for reducing the amount of backup resources reserved in the network by aggregating backup resources requirements at a router port for connections that have disjoint primary routes. At each router port a sharing bandwidth table is required which keeps track of the backup resources needed. Specifically at a port, for the set of connections whose primary routes are disjoint, one reserves standby resources equivalent to those needed by the connection in the set with the maximum resource requirement. In contrast, for the set of connections at a port whose primary routes are not disjoint, one reserves standby resources equivalent to the sum of the resource requirements of all connections in the set. The drawback of this approach is that while resources for recovery from a single failure are ensured, the network load that can be supported under normal operation is reduced considerably due to the reservation of standby resources. Also the connection setup time is increased as a connection/flow cannot begin transmitting data until both the primary and backup route resources have been reserved.

The second approach is to setup a working connection and attempt traffic restoration only after a failure occurs. Specifically, the source node of the failed connection conducts a dynamic search for the selection of a fault recovery route from a set of predefined possible routes. The required QoS resources are then requested along one backup route just as if a new connection were being setup. Note that in this case one is not guaranteed recovery from a failure as the necessary resources may not be available. However the load supported under normal operations can be considerably higher than the first scheme and the connection setup time is less. One point often made against such a dynamic search approach is that the speed of restoration will be longer than a reserved backup method. However, the outage time seen by a connection after a failure is made up of the components: fault detection time, alarm dissemination time, and traffic restoration time. Considerng realistic numbers (e.g.; 30-45 secs to detect link failure in current routers[6]) for the time components one sees that the fault detection time is the largest component and the outage time for a dynamic search will be on the same order of magnitude as that for a reserved backup approach.

For both approaches an important underlining component is that a candidate set of paths for each source-destination pair must be known. The candidate path set used here is the set of shortest routes subject to a hop count limit, augmented with a set of link disjoint routes. The hop count limit was imposed in order to limit the size of the path set. Given the path set, we adopt IBM's NBBS (Networking BroadBand Services) source based QoS path selection algorithm [14]. The algorithm selects the minimum cost path where the cost of a path is defined as the sum of the link cost on the path. The link cost function $w(l)$ for link l is defined as

$$w(l) = \frac{C_l}{(C_l - B'_l)(C_l - B_l)} \quad (1)$$

where C_l is the link capacity, B_l is the bandwidth occupied by existing connections and B'_l is the bandwidth used if the new connection were to be added on this link. Each node in the network is assumed to have knowledge about the capacity available on each directional link or port. This can approximately be achieved through the periodic distribution of the required information from every node [14].

To support the implementation of the two survivability mechanisms, we extended RSVP. The explicit route object is used to specify a path selection made at the source node. This object is carried in the RSVP `Path` message and will be delivered to necessary agents (e.g., routing agent, admission control agent). In the reserved backup-path restoration method, primary path information will be included in the signalling message during backup path reservation and used to determine if backup resources can be shared. In both survivability

techniques, when adjacent nodes detect a failure, the RSVP `ResvTear` message will be sent in the upstream direction of a failed connection to notify the traffic source to redirect its traffic to the reserved backup path in the first scheme or to discover a new route with the necessary resources to restore the connection in the second method. The `PathTear` message will be sent in the downstream direction to tear down the connection and notify the destination node to start receiving traffic from the alternate path. Thus, resources along paths of failed connections are released for further use.

III. Performance Evaluation

A simulation model of the two schemes has been developed by extending the Network Simulator-NS (version 2) [12]. NS supports the simulation of TCP, IP, routing, and multicast protocols. In order to simulate the survivability schemes new modules were added to NS including a admission-control agent, RSVP agent, flow-routing agent, resource agent and fault-tolerant agent. The admission-control agent determines if a connection requesting QoS will be accepted based on the available resources. The RSVP agent will send reservation messages to setup or tear down the flow along the path given by a flow-routing agent. The flow-routing agent at each node maintains path information (set of candidate paths to other nodes) and routing information once the flow is set up. This agent also runs the path selection algorithm to find the path that gives the minimum cost route. A resource agent at each node keeps track of resource levels at all ports. The simulation model is constructed so that all nodes share global information of resource levels. A fault-tolerant agent at each node incorporates the two different restoration recovery schemes. For the reserved backup-path restoration method, this agent determines the amount of backup resouces shared at each link.

The 11-node backbone network [13] shown in Figure 1 was studied. There are 19 bidirectional links all with the same capacity of 1.544 Mbps. A long haul network is assumed with a propagation delay of 100 msec at each link. The initial candidate path set was the eight shortest paths for each node pair. Since this set may not provide disjoint paths, the path set was augmented (if necessary) to include additional paths until at least three disjoint paths are in the set for each node pair.

Traffic demand was generated between randomly selected source and destination nodes. The bandwidth needed by each traffic flow was uniformly distributed between 1.5 to 2.5 percent of the minimum network link bandwidth. At the packet level, variable-size data packets were generated using the empirical distribution based on measurement data from the MCI commercial Internet backbone [15]. Packet interarrival times were assumed exponentially distributed. The RSVP signalling packets were assumed to be of equal size of 64 bytes. This is smaller than actually needed for the reserved backup-path restoration scheme since primary path information has to be included in the RSVP packet. TCP window-based flow control was used at the transport layer.

After a failure occurs there is a time delay in detecting the failure, disseminating the failure information to the affected source nodes and restoring the failed connections. During this time delay packets in transit on the failed connections may be lost, requiring retransmission by the traffic source. Thus creating a backlog of dropped packets at the sources of the various failed connections. This backlog may create transient congestion in the network after traffic restoration occurs. In order to observe the transient congestion due to backloging a infinite buffer size was used at each network node. Along with the maximum advertized TCP window size (TCPWnd) and congestion window size set large enough so that after a failure backlogged packets are quickly retransmitted with little waiting on a slot in the window (TCPWnd = 97 packets was used). This corresponds to greedy TCP.

Experiments were conducted by running the simulation until a given mean network load was reached and steady-state was attained, then a link was failed and the transient fault recovery period was observed. Statistics were collected for percentage of calls rejected, network load, offered load, number of packets dropped (backlog size), call rerouting time, percent restoration call blocking, and percent demand restored. Experiments were repeated 10 times in each case and 95 % confidence intervals were computed. In the transient analysis the ensemble average behavior at various time instances was determined for the instantaneous end-to-end packet delay and instantaneous queue-length. The instantaneous delay is the average delay experienced by the packets entering the network computed over 500 msec periods.

In the results reported here (additonal results in [16]), link 7-9 was failed and the failure was detected by the adjacent nodes after 3 seconds. Note that in current routers, the detection time ranges from 30 to 45 seconds using the standard Hello protocol. From [6], a three second failure detection time is possible at T1 speed or higher when Hello packets are forwarded with high priority, a 1 second Hello-update interval is adopted and the failure is assumed after 3 losses (all are modifications to normal router operation).

Here we summarize some of the simulation results, to show the trade off between the two different restoration schemes. Table 1 shows the basic results for the two schemes. As shown in Table 1 the reserved backup path scheme results in a high call blocking rate under normal operation. Notice that the % call blocking increases dramatically with increasing load. This is due to the fact that the total reserved bandwidth (working + backup) on some links is near link capacity. For example a average network load of 0.6 for the working traffic can drive the mean effective load (working + reserved backup) to as high as 96.95 % of the link bandwidth. A detailed analysis of the simulation results shows that the reserved backup path scheme unfairly penalizes connection requests from node pairs that are a far distance apart. That is once the network load is sufficiently high to result in significant call blocking, the connection request that tend to be accepted are for node

pairs seperated by a single hop. Only a small capacity gain was observed by sharing the backup resources among flows, whose primary paths are disjoint. The major benefit of the reserved bandwidth recovery scheme is 100 % restoration under a single link or node failure, as verified by the simulation results shown in Table 1. Also, notice that the mean restoration time is small ($\approx$ 0.5 sec $>$ the detection time) and the range of restoration times is small.

In the dynamic search restoration scheme, more calls can be admitted during normal operation. As shown in Table 1, at 0.6 load, the mean call blocking is 0.2 % of calls and no call blocking was observed at lower loads. Notice that the % restoration call blocking is **zero** at mean network loads of 0.6 and below. Thus, while fault recovery is not guaranteed one can expect that at moderate network load *most connections will be recovered.* Furthermore, the mean restoration time of this scheme is only *slightly* longer than the reserved backup path approach. However, the range of restoration times is larger. The results shown in Table 1 are in a sense the best case scenario for the dynamic restoration scheme since we have assumed that the information of resource availability at every node is consistent and accurate. This is not always true since the exchange of resource information is typically done on a periodic basis and may not be consistent, especially after a failure.

As shown in Table 1 for TCPWnd = 97, the mean backlog size created by all failed connections for the dynamic restoration scheme is less than the backlog for the reserved backup path scheme. The dynamic restoration approach yields a smaller number of connections that needed to be restored after the failure. This is due to the fact that the reserved backup approach results in longer routes for traffic from the same node pairs for the working flows since each connection uses greater resources. The longer the working path, the more likely it will be affected by a failure.

Figures 2-6 show the typical transient network behavior of the two schemes. Figure 2 shows the instantaneous end-to-end packet delay of the traffic between nodes 4 and 9 that is affected by the failure when the mean network load is 0.6 and the reserved backup path scheme is used. Notice the mean delay is in steady state until time 1950 when the failure occurs, the delay then becomes very small as almost no packets reach the destination. After traffic restoration around time 1954 the connections have been switched to their backup paths and one can observe an increase in the delay due to a combination of the backlog retransmission and the backup paths being longer than the orginal path. The corresponding behavior for the dynamic rerouting scheme is shown in Figure 3. From Figure 2, one can clearly see that the delay for the reserved backup scheme increases dramatically. In contrast, the dynamic rerouting scheme better balances the network load after the failure with a short transient of around 25 seconds.

The queue length versus time was also measured in the simulation at all network queues. Here we show samples results at one of the links used (the link connecting node 9 to 5) to route around the failure. The instantaneous queue size at link 9-5 using the reserved backup path scheme and the dynamic rerouting scheme are compared in Figures 4 and 5 for 0.6 load. One can see a large difference in the transient behavior of the two schemes. Specifically, at a load of 0.6, the reserved backup scheme requires a buffer size of 24 Kbytes for no packet loss, whereas the dynamic restoration scheme requires a buffer size of less than 2 Kbytes. Notice at the 0.6 load, for the reserved backup scheme the queue will build up rapidly and is not stable within 800 sec, requiring a large maximum buffer size even though the steady state load on the link is stable.

To study the extent of network congestion, we compared the number of links used in traffic restoration and the number of links congested by restoration. Table II shows the results of our simulations, notice that on average the reserved backup path scheme requires the use of more links in restoring the traffic. To determine which links are congested, we follow [5] and note that typical buffer size for at a router port for this network would be 20 Kbytes, with a congestion threshold (e.g., for RED) of 16 Kbytes. By examining the queue length plots of all links used for restoration and applying the congestion threshold we get the results shown in Table II, which show the dynamic restoration scheme congests fewer links than the reserved backup scheme.

In general, it was found that the transient congestion was more severe for the reserved backup path approach than the dynamic restoration scheme. This is consistent with the reserved backup scheme, not utilizing spare resources as efficiently as the dynamic scheme. Note that for sources, that implement TCP transport protocol, adjustment to the maximum advertised congestion window can reduce the magnitude of the congestion but will lengthen its duration. Figure 6 shows the queue length versus time of the reserved backup path scheme when a maximum TCP window of 48 is used along with a finite buffer of 20 Kbytes at each router port. Comparing Figure 6 to Figure 4, one can see the congestion in the network is reduced. However, when one looks at the number of dropped packets (both the backlog due to failure and dropping due to congestion of finite buffers) as shown in the bottom of Table 1, one sees that the number of packets needing retransmission increases considerably. Determining how to set the maximum window size to make the tradeoff between transient congestion magnitude and duration is currently under study.

IV. Summary

In this paper we investigated two schemes for improving the survivability of connections in a possible NGI network architecture. In the first scheme a QoS service is provided reserved standby backup resources on a path which is disjoint with the working path. In order to reduce the amount of backup resources required, the backup resources are shared among con-

	Mean Offered Load	Reserved Backup Path		Dynamic Restoration Backup Path	
		Mean	95% C.I.	Mean	95% C.I.
% Call Blocking	0.40	7.79	(5.06, 10.51)	0	(0, 0.45)
	0.50	15.24	(13.24, 17.25)	0	(0, 0.36)
	0.60	52.86	(49.99, 55.72)	0.20	(0, 0.49)
% Restoration Call Blocking	0.50	0	(0, 4.63)	0	(0, 8.43)
	0.60	0	(0, 2.63)	0	(0, 6.20)
Restoration Time (sec) Mean, (Min,Max)	0.50	3.55 , (3.20, 4.41)	(3.54, 3.55)	3.56 , (3.40, 4.47)	(3.56, 3.57)
	0.60	3.50 , (3.20, 4.44)	(3.50, 3.50)	3.96 , (3.20, 11.27)	(3.93, 3.99)
Backlog Size (Kbytes) TCPWnd = 97	0.50	23.46	(20.84, 26.07)	15.21	(13.39, 17.02)
	0.60	40.24	(37.23, 43.24)	30.01	(21.06, 38.97)
Backlog Size (Kbytes) TCPWnd = 48	0.50	231.86	(180.01, 283.62)	102.88	(66.17, 139.60)
	0.60	7008.14	(4760.60, 9255.68)	246.33	(246.33, 591.69)

TABLE I

COMPARISON OF 2 DIFFERENT RESTORATION MECHANISMS

	Mean Offered Load	Reserved Backup Path		Dynamic Restoration Backup Path	
		Mean	(Min, Max)	Mean	(Min, Max)
No. of links used in restoraton	0.5	24.3	(17, 31)	19.1	(14, 22)
	0.6	26.7	(21, 31)	21.3	(14, 26)
No. of links congested	0.5	5	-	3	-
	0.6	10	-	6	-

TABLE II

NUMBER OF LINKS USED AND CONGESTED DURING TRAFFIC RESTORATION

nections whose working connections traverse disjoint paths. In the second approach a dynamic search for restoration resources is conducted over a preplanned set of alternate paths upon notification of a failure. A comparative simulation based study of the performance of the two survivability schemes was presented. The study considered both steady state and transient network behavior and shows that the reserved backup scheme results in much higher connection blocking under normal operations, slightly faster restoration times, a larger backlog of packets needing retransmission after a failure and more severe transient congestion after traffic restoration. Since the reserved backup path approach leads to such inefficient use of resources under normal operations and the possibility of long transient congestions after a failure we recommend that it be used only for a small portion of network traffic. Specifically for connections that require a guaranteed level of fault tolerance. Depending on the network loading is may be possible to assign the two survivability approaches to different service classes. For example, the Guaranteed Service class in the Integrated Services NGI model could use the reserved backup restoration scheme, where as the Controlled Load class should implement the dynamic rerouting scheme to discover the alternate path when needed.

REFERENCES

[1] R. Braden, et al., "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification" , RFC 2205, Sept. 1997. see www.ietf.org

[2] E. Crawley, et al., "A Framework for QoS-based Routing in the Internet," RFC 2386, August 1998. available from www.ietf.org.

[3] R. Doverspike, "Trends in Layered Network Management of ATM, SONET and WDM Technologies for Network Survivability and Fault Management," *Journal of Network and Systems Management*, Vol. 5, pp. 215-220, 1997.

[4] K. Dovrolis and P. Ramanathan, "Resource Aggregation for Fault Tolerance in Integrated Services Packet Networks," *Computer Communication Review*, Vol. 28, No. 2, April 1998.

[5] S. Keshav and R. Sharma, "Issues and Trends in Router Design," *IEEE Communications Magazine*, Vol. 36, No. 5, May 1998.

[6] P. L. Higginson and M. C. Shand, "Development of Router Clusters to Provide Fast Failover in IP networks," *Digital Technical Journal*, Vol. 9, No. 3, pp. 32-41, 1997.

[7] D. H. Gan, et al., "Setting up Reservations on Explicit Paths using RSVP," *Internet Draft, draft-guerin-expl-path-rsvp-01.txt*, Nov. 1997.

[8] R. Callon, et al., "A Framework for Multiprotocol Label Switching," *Internet Draft, draft-ietf-mpls-framework-02.txt*, November, 1997.

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

[10] D. O. Awduche, et al., "Requirement for Traffic Engineeering over MPLS," *Internet Draft, draft-ietf-mpls-traffic-eng-00.txt*, October 1998

[11] Y.Bernet, et al., "A Framework for Differentiated Services," *Internet Draft, draft-ietf-diffserv-framework-02.txt*, Febuary 1999

[12] UCB/LBNL/VINT Network Simulator-NS (version 2), available from http://www-mash.cs.berkerley.edu/ns

[13] D. Medhi and S. Shah, "Performance under a Failure of Wide-Area Datagram Networks with Unicast and Multicast Traffic Routing," *Proc. of IEEE MILCOM'98*, Boston MA., October 1998.

[14] L. Gun, et. al., "NBBS path selection framework," *IBM Systems Journal*, vol. 34, no. 4, 1995.

[15] G. J. Miller, K. Thompson and R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics", *IEEE Network*, Nov/Dec. 1997.

[16] A. Srikitja, et. al., "On Providing Survivable QoS Services in the NGI," *Technical Report, University of Pittsburgh*, see www.tele.pitt.edu/tipper.html

[17] P. White and J. Crowcroft, "The Integrated Service in the Internet: State of the Art," *Proceedings of The IEEE*, Vol. 85, No. 12, Dec. 1997

[18] T.H.Wu, *ATM Transport and Network Integrity*, Academic Press, New York, NY., 1997.

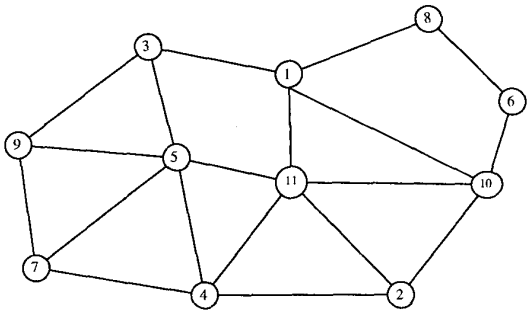

Fig. 1. Topology of the 11-node test network

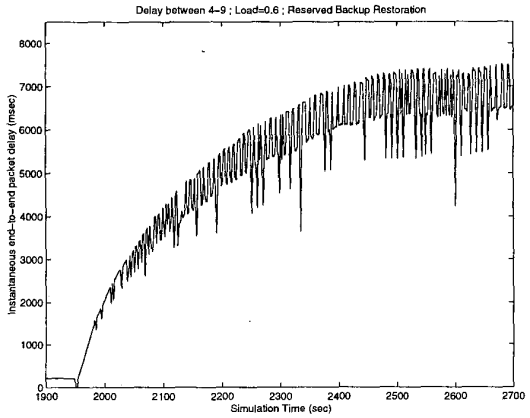

Fig. 2. Inst. End-to-End Delay between 4–9 (load=0.6)

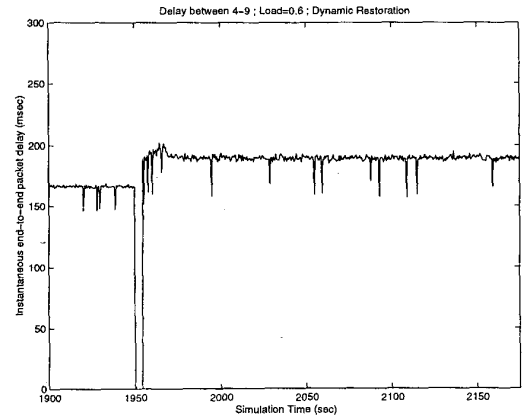

Fig. 3. Inst. End-to-End Delay between 4–9 (load=0.6)

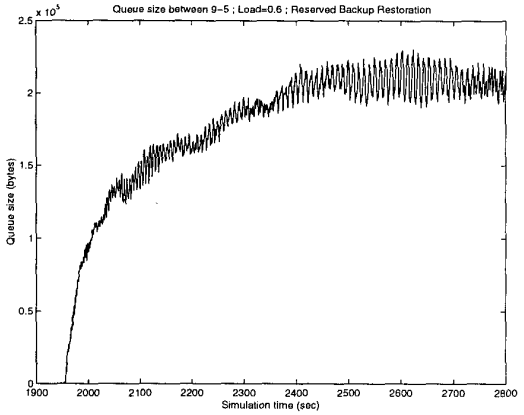

Fig. 4. Inst. Queue Size between 9–5 (load=0.6), TCPwnd size=97

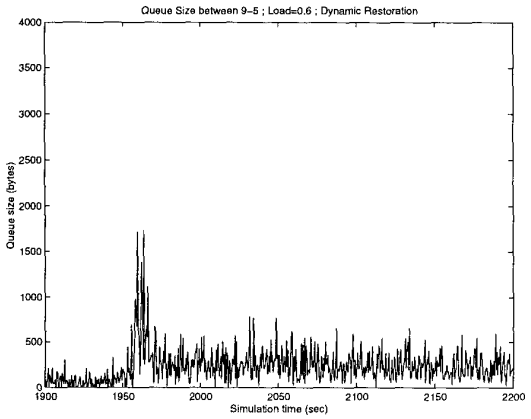

Fig. 5. Inst. Queue Size between 9–5 (load=0.6), TCPwnd size=97

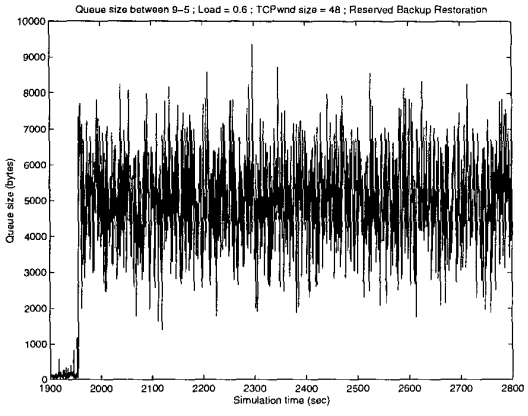

Fig. 6. Inst. Queue Size between 9–5 (load=0.6), TCPwnd size=48