

# Performance Analysis of Multicast and Priority-Based Routing under a Failure in Differentiated-Services Internet\*

Gerald Rogers<sup>‡</sup>, Deep Medhi<sup>‡</sup>, Wen-Jung Hsin, Suresh Muppala  
Department of Computer Networking  
University of Missouri–Kansas City  
Kansas City, MO 64110, USA

David Tipper  
Department of Information Science and Telecommunications  
University of Pittsburgh  
Pittsburgh, PA 15260, USA

## Abstract

*In this paper, we study the transient network behavior of multicast and unicast connections in a differentiated services capable Internet architecture under a major link failure. This is done using the MoMaRS 2.5 simulation tool that we have developed where we have implemented a multicast routing component and priority based scheduling/routing to emulate a differentiated services environment. In our simulation results, we consider the multicast workload to be of higher priority compared to unicast services. Our results show that the performance of the multicast routing protocol is essential in reducing the overall network utilization, thereby reducing the overall delay to all packets through efficient network utilization.*

## 1 Introduction

In recent years, the Internet has seen an increasing demand for transporting "point-to-multipoint" (such as Telecast) and "multipoint-to-multipoint" (such as audio/video conferencing) multicast traffic [5], besides the popular reliable "point-to-point" unicast traffic (such as telnet, ftp, email, and web.) As any network gains wide acceptability, an important issue to address is network survivability under a major link or a node failure. A failure within a network changes the network topology and hence changes the paths over which packets are routed. Thus, the transient behavior in the network and the adaptability of the network to a failure is a critical issue. Unicast based applications such as Telnet, FTP, HTTP, etc. running over a reliable transport protocol (e.g.,

TCP) compensate for failures by retransmitting data; however, TCP is designed to compensate only for very minimal loss of packets (less than 1%, see [7]). Real time delay sensitive applications such as video conferencing and IP Telephony running over a unreliable transport protocol (e.g., UDP) avoid acknowledgements and retransmission, because it is better to lose a packet than to retransmit a packet due to delay sensitivity. Nevertheless, it is important to minimize the packet loss by providing efficient routing algorithms, that quickly react to the failure within the network and reroute packets to their destination. Fault recovery routing is particularly important for multicast applications like video conferencing.

In [6], Shah and Medhi have reported the development of the MoMaRS (Missouri - Maryland Routing Simulator) simulation tool where multicast workloads are implemented alongside unicast components in a best effort services environment. This tool was designed to study a major link failure for a best effort TCP/IP network and simulation results showing the effects of link failures were presented in the same paper. We have now extended this tool to include the multi-cast routing developed by Kompella *et. al* [4] (hence to be referenced as MKOMPELLA). We add fault tolerant multicast to MKOMPELLA to reduce the impact of a link failure on the multicast flow. Furthermore, priority-based routing algorithms have been incorporated, thus making this tool capable of considering differentiated-services Internet [1] environment. For ease of reference, we will refer to this updated simulation tool as MoMaRS 2.5. In this work, we highlight the new components of MoMaRS 2.5 as well as present some results on the transient behavior of the network within the scope of differentiated services under a link failure. The rest of the paper is organized as follows. In Section II, we describe the extensions to

\*Research supported by DARPA and Air Force Research Lab, Air Force Materiel Command, USAF, under agreement No. F30602-97-1-0257.

<sup>‡</sup>Current affiliation: LarsCom, North Carolina

<sup>‡</sup>Author for Correspondence

the MoMaRS 2.5 simulation testbed. Our simulation scenarios are discussed in Section III while results and observations are presented in Section IV.

## 2 MoMaRS 2.5 Simulation Testbed

In this section, we briefly describe the multicast routing algorithm MKOMPELLA, and priority-based scheduling/routing in the context of Differentiated Services, as they have been implemented in MoMaRS 2.5.

### 2.1 Multicast Routing Component: MKOMPELLA

MKOMPELLA routing algorithm is a heuristic approach to solving the constrained Steiner Tree problem [3, 4]. The algorithm maintains two optimization goals. The first goal is to minimize the average path delay, which is the average of the minimum path delays from the source to each of the destinations in the multicast group, and the second goal is in terms of the cost of the multicast tree, which is the sum of the costs on the edges in the multicast tree. The general steps of the algorithm are as follows: First, given graph  $G$ , construct a closure graph  $G'$  which is the minimum cost path from the multicast source to all multicast destinations. Next, from the closure graph  $G'$ , construct the minimum cost multicast tree  $T$ . Finally, given the minimum cost multicast tree  $T$ , remap the multicast tree onto graph  $G$ , removing any loops to obtain the final multicast tree  $T'$ .

We have modified the original algorithm to fit the MoMaRS 2.5 environment. First, we modified the algorithm to use the MoMaRS 2.5 data structures. Second, we did not use the modified Floyd Warshall [4] algorithm to compute the minimum cost path from any source  $S$  to any destination  $D$ . Instead we used the straight Floyd Warshall algorithm [2] that was already implemented in the MoMaRS 2.5 environment for unicast minimum path calculation. Third, we modified the algorithm to compute the minimum cost multicast tree for more than a single group. Fourth, we used a depth first recursive tree traversal algorithm to remap the minimum cost multicast tree  $T$  to the final multicast tree  $T'$ . Last, we modified the algorithm to map the resulting multicast tree into the multicast routing table structure of MoMaRS 2.5.

We have also modified the multicast tree algorithm to make it fault tolerant. The multicast tree is reconstructed based upon the link failure. This process is initiated when a topology update is broadcast to the network. When the link fails, that change in topology is broadcast throughout the network. When a node receives the topology change, a recalculation of the multicast tree is initiated. Therefore it is possible that a multicast flow can be disrupted when a multicast packet reaches a node that has not recalculated the multicast

tree. In other words, it will take time for the multicast tree to stabilize. The signaling portion of this is accomplished by the node. When the node detects a link failure, it will initiate a route update message to all adjacent nodes.

### 2.2 Priority-Based Routing: Differentiated Services

The priority-based routing algorithm we have chosen is a non-preemptive scheme with five levels of priority and starvation. In the context of differentiated services, the MoMaRS 2.5 traffic component can specify the traffic priority for packet processing at the routers. We have modified all traffic generating components to place a priority level on each packet they generate. We then modified the node and link components to implement the priority-based routing algorithm. Specifically, each node component has a classifier and a packet scheduler. We place the priority queues at each outgoing link, commonly known as output queuing mechanism, to avoid Head of Line blocking. The classifier forwards incoming packets to an appropriate outgoing priority queue. The scheduler uses a simple priority-based scheduling scheme, i.e., it transmits packets from higher priority queues before sending packets from lower priority queues.

## 3 Simulation Network and Scenarios

For our simulation we have designed an eleven node network with eighteen links, as shown in Figure 1. The links have different bandwidths as shown in Figure 1 and a propagation delay of 100  $\mu sec$ . The data packet processing delay at a node is 1000  $\mu sec$ , and the routing packet processing delay is 6  $msec$ . Each priority queue has infinite buffer space, unless specified otherwise.

We define a unit of traffic  $U$  to be a workload with an average packet size of 512 bytes, and a packet interarrival time of 100  $msec$ . In our network we have a total of 28 connections consisting of both unicast and multicast traffic. Each traffic type has its own characteristics based upon  $U$ , as defined in Table 1. Table 2 details the source-destination node pairs for each traffic type. A source/sink pair or multicast group is delineated by  $\{ \}$ . For conference traffic any member of the group can be a multicast source. For Telecast the first entry in the list is the multicast source. For unicast (i.e., FTP and Telnet) the first entry in the list is the source. We set the inter-packet distribution times for all traffic sources exponential throughout our study. Within MoMaRS 2.5 environment, TCP is not implemented. Instead a simple sliding window protocol is provided with a maximum send window size of 20 for FTP and 30 for Telnet. The packets are produced based upon the interarrival distribution of the traffic, but are limited to the window sizes.

The multicast sessions use UDP with no retransmission of dropped packets.

To evaluate our network, we have established eight simulation scenarios as shown in Table 3. In all simulation scenarios, the routing packets have the highest priority (level 1). All simulation scenarios were executed for 1000 seconds; of which, the first 100 seconds were used for the warm up time. To study a link failure, we have a link fail at 300 seconds, with full recovery of the link at 600 seconds. This allows for the observation of network dynamics during and after failure. There is a 2000  $\mu$ sec delay between the failure, and the detection of the failure and recovery. We performed multiple replications, and the graphs discussed in the next section depict the average; the 95% confidence-interval is found to be around 4% of the average. We consider two link cost calculation methods: delay-based and utilization-based. In delay-based, the routing calculation uses instantaneous (or average) delay on a link to determine the unicast and multicast paths. In utilization-based, the routing calculation uses the instantaneous (or average) utilization of a link to determine the unicast and multicast paths. The average delay is based upon the overall delay of the network since the simulation begins. The instantaneous delay is calculated based upon an average over a small interval.

## 4 Results and Observations

We summarize some key results (due to the paper length limitation) for a specific set of scenarios. For the results reported here, we consider the effect due to failure of the link 5-11. This failure is chosen because it is a high capacity link within the middle of the network carrying a substantial flow of data through the network. The effect due to other link failures is similar to that of link 5-11. We first present the results of our new multicast routing component MKOMPELLA, then we present the results for priority-based scheduling/routing.

### 4.1 Multicast Routing

We first discuss our results for the infinite buffer case. To study the performance of the new multicast component, we first take a look at the network performance with no link failures. Figure 2 shows the plot of both instantaneous and average overall unicast traffic delays for both utilization and delay based calculation methods. The overall unicast traffic delay is higher for the utilization-based approach. However, under the utilization based method, we see that the traffic is more evenly distributed across the network. Note that the network stabilizes around 300 seconds, which allows us to perform a link failure at that time.

Next, we fail link 5-11 at 300 seconds, and recover it at 600 seconds. Figure 3 shows the instantaneous delay for

a selected multicast traffic stream flowing over link 5-11. In figure 3, Telecast Sink 114 represents node number 11 and connection number 4. Figure 4 provides a view of the instantaneous and average delays for the overall unicast traffic. Comparing Figure 2 and Figure 4 shows that there is a sudden increase in delay at the time of the failure. This is because, in our study, we delay the triggered routing calculation for 10 seconds, therefore some of the receivers downstream from link 5-11 may not get any data during this time. As to be expected, these figures show that the delay increases during failure. In this link failure, we notice that MKOMPELLA routing algorithm recovers in approximately 30 seconds after the link failure. Figure 5 shows how the traffic is rerouted around the failed link 5-11. Specifically, this figure shows that adjacent link 4-5's utilization increases, whereas adjacent link 5-9's utilization decreases because the affected data take different paths after failure.

To study the network with finite buffer spaces, we set node 3 and node 4 equal to 80% of their maximum buffer sizes in the above failure simulation model. In our study, we observe that finite buffer increases the overall unicast traffic delay within the network during the failure under utilization-based calculation. The delay is most evident for the traffic sources that flow over those nodes with finite buffer capacity. We also notice an increase in the packet loss rate, see table 4. The reason for the packet loss in the infinite buffer case is due to some packets being in the transit upstream of link 5-11 at the time of failure.

### 4.2 Priority-Based Routing

For the priority-based routing study, we include the packet priority as follows (in order of priority): Routing, Multicast, and Unicast. We first discuss our results for the infinite buffer case. To study the performance of the priority-based routing, we first take a look at the network performance with no link failures. We observe a dramatic decrease in the instantaneous delay and jitter for multicast traffic when priority-based routing is employed, see figure 6. However, the delay for unicast traffic increases over the non-priority simulations. We see that with no failures, the priority-based routing algorithm provides better QOS to those services with a higher priority.

When subjected to link 5-11 failure, the instantaneous delay for multicast traffic increases slightly, but there is a dramatic increase in the delay for unicast traffic. Figures 7 and 8 show the instantaneous delay for selected multicast and unicast traffic streams flowing over link 5-11.

In general, in the finite buffer case, restricting the buffer space at node 3 and node 4 to 80% of their maximum values obtained from the previous failure study,

dramatically increases the delay of unicast traffic and decreases the delay of multicast traffic when using priority-based routing.

## 5 Summary

In this paper, we have studied the performance of the multicast routing algorithm MKOMPELLA, and provided QoS priority for differential services within a network. We particularly emphasized on the transient behavior of the network when it is subjected to physical link failures. From the studies carried out thus far, it can be concluded that MKOMPELLA multicast routing algorithm successfully accomplishes the rerouting of traffic during a link failure. Although delays are higher during the failure duration (due to less capacity in the network), they do stabilize. When subjecting MKOMPELLA to finite buffers, we saw a significant increase in packet loss. We showed that the priority-based routing algorithm that we have chosen to implement Differentiated Services, does provide some QoS assurance. Although the priority-based routing algorithm may result in starvation of lower priority traffic, if the network is designed with ample rollover capacity, the starvation of lower priority packets can be minimized.

## References

- [1] Bernet, Y. et. al "A Framework for Differentiated Services," Internet Draft draft-ietf-diffserv-framework-02.txt, February 1999.
- [2] Bertsekas, D. and Gallager, R. *Data Networks*, 2<sup>nd</sup> ed, Prentice Hall, Englewood Cliffs, N.J. 1992.
- [3] Goemans, M. X. and Myung, Y. "A catalog of Steiner Tree Formulations." *Networks* 23, 1993. pp. 19-28.
- [4] Kompella, V.P., Pasquale, J.C., and Polyzos, G. C. "Multicast-Routing for multimedia communication" *IEEE/ACM Transaction on Networking* 1 (1993), pp. 286-291.
- [5] *Networking Research Group at Lawrence Berkeley National Lab web site.* <http://ee.lbl.gov/>
- [6] Shah, S. and Medhi, D. "Performance under a failure of wide-area datagram networks with unicast and multicast traffic routing." *Proceeding of IEEE Military Communications Conference (MILCOM'98)*, Boston, MA, October 1998.
- [7] Stevens, W. R. "TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms," Internet Request for Comment RFC 2001, January 1997. (available from [www.ietf.org](http://www.ietf.org))

Table 1: Traffic Definitions

| Traffic Type  | Packet Size (bytes) | Packet Interarrival Time (ms) | Basis Value |
|---------------|---------------------|-------------------------------|-------------|
| FTP Source    | 512                 | 100                           | 1           |
| FTP Sink      | 512                 | 100                           | 1           |
| Telnet Source | 128                 | 100                           | .25         |
| Telnet Sink   | 512                 | 100                           | 1           |
| Conference    | 512                 | 100                           | .25         |
| Telecast      | 80                  | 80                            | .194        |

Table 2: Network Traffic

| Workload Type | Members                              |
|---------------|--------------------------------------|
| Conference    | {6,5,7,4} {1,10,6,8} {4,2,10,6}      |
| Telecast      | {2,11,5,3,1,8} {3,5,9,7,11}          |
| FTP           | {1,3,9,11,4,7,5} {11,5,4,9,7,10,6,8} |
| Telnet        | {2,9} {3,6} {3,8} {4,8} {5,8}        |
|               | {6,7} {6,11} {7,9} {8,9} {9,11}      |
|               | {2,7} {3,4} {3,7} {5,6} {3,10}       |
|               | {6,9} {7,8} {7,10} {9,10}            |

Table 3: Eight simulation scenarios

| Buffer Size | Packet Priority Level       | Link Failure    |
|-------------|-----------------------------|-----------------|
| Infinite    | All data packets (Level 2)  | 5-11, 7-9, 6-10 |
|             | Multicast packets (Level 2) | 5-11, 7-9,      |
|             | Unicast packets (Level 3)   | 6-10            |
| Finite      | All data packets (Level 2)  | 5-11            |
|             | Multicast packets (Level 2) | 5-11            |
|             | Unicast packets (Level 3)   |                 |

Table 4: Packet Loss Comparison

|        | Infinite Buffer | 80% Buffer |
|--------|-----------------|------------|
| Node 3 | 10              | 904        |
| Node 4 | 2               | 1284       |

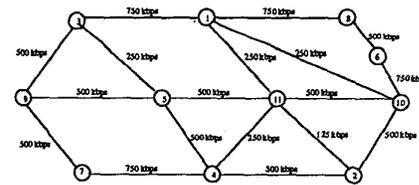


Figure 1: The 11-node test network

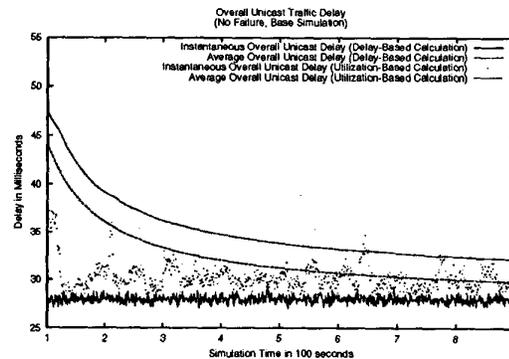


Figure 2: Average and Instantaneous Unicast Traffic Delays (No Failure)

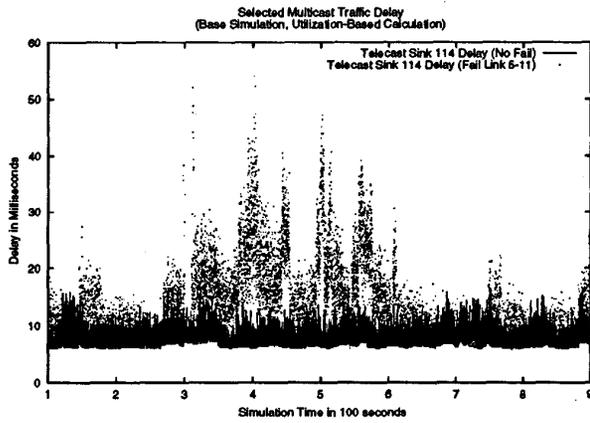


Figure 3: Telecast Sink 114 Instantaneous Delay

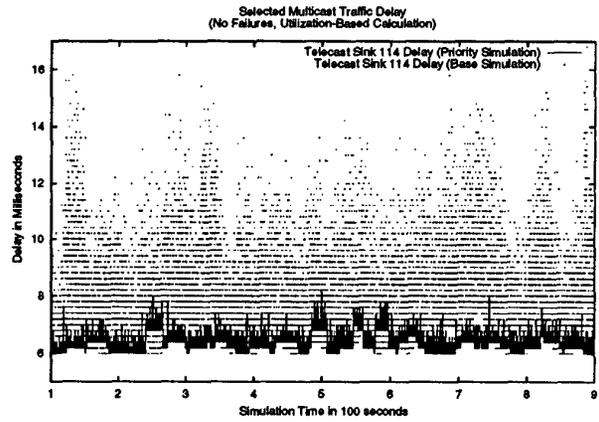


Figure 6: Telecast Sink 114 Instantaneous Delay (No Failure)

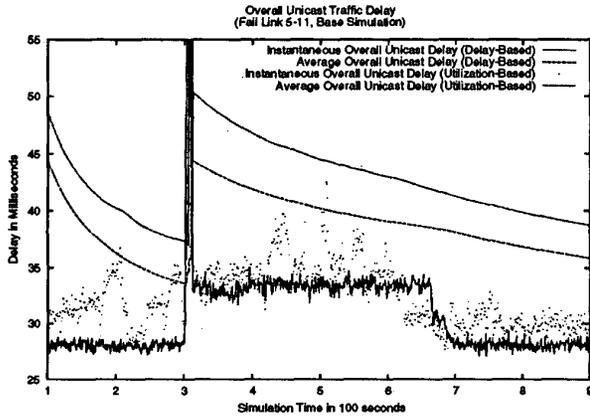


Figure 4: Average and Instantaneous Unicast Traffic Delays (Fail Link 5-11)

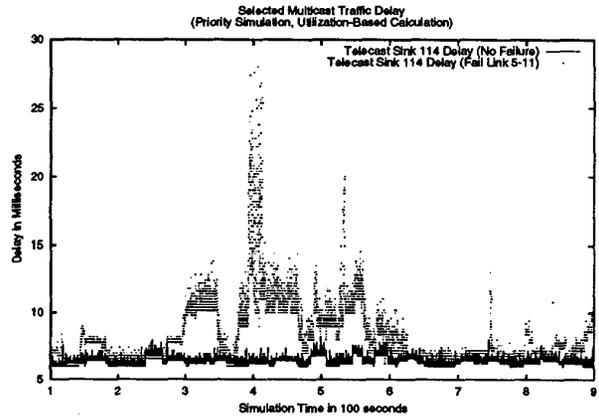


Figure 7: Telecast Sink 114 Instantaneous Delay (Priority-Based Routing)

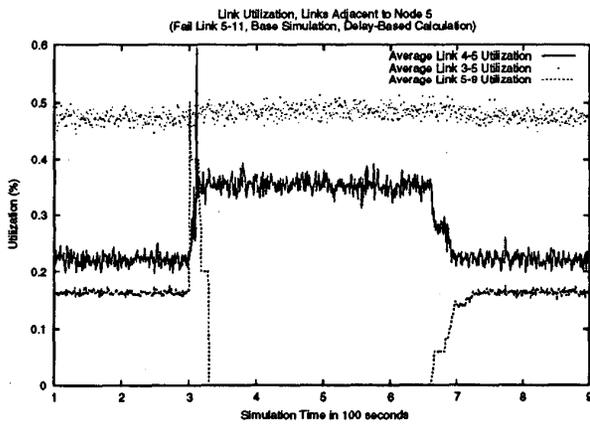


Figure 5: Link Utilization for Links Adjacent to Node 5 (Fail Link 5-11)

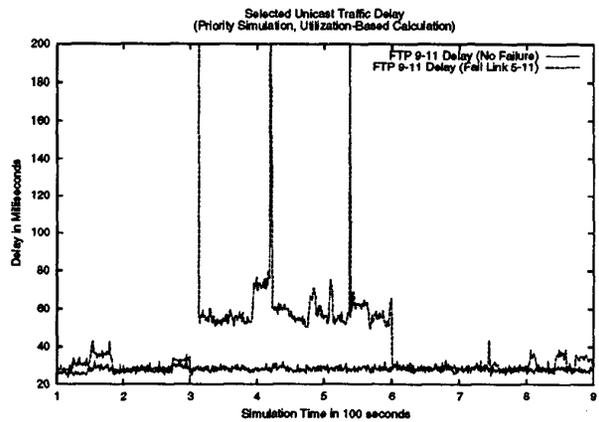


Figure 8: FTP9-11 Instantaneous Delay (Priority-Based Routing)