# Part 2

# Simulation Analysis

In Part 1 of this dissertation, we have established that the relationship between packet loss events and increases in RTT samples is not conducive to DCA. We conjecture that the fundamental problem is that an RTT does not provide enough information for an endpoint to reliably predict packet loss events. In order to prove this we must monitor the queueing points within the network where loss occurs and show the reasons why an RTT sampling process is not able to reliably detect the queue buildup that precedes loss events. This was not possible only by analyzing the end-to-end traffic.

Over the next several chapters, we use simulation to provide analysis into the problems associated with DCA in a manner that was not possible using end-to-end measurements. We find the following limits the effectiveness of DCA.

- The granularity of a TCP constrained RTT sampling process prevents an endpoint from accurately tracking the bursty congestion associated with packet loss. Factors such as the bursty traffic arrival processes at high speed switches along with the dynamics of TCP's congestion control algorithms make it difficult for an endpoint to reliably avoid packet loss.
- An end-to-end sampling process is not able to accurately assess the congestion level of a router. A DCA algorithm cannot differentiate between the significant number of increases in RTT that are not associated with packet loss from those increases that do lead to loss.

We also confirm the following assumptions that our measurement work has made:

- DCA does not significantly reduce the congestion level over the path. This assumption is necessary to justify that there are no benefits to incrementally deploying DCA. Similarly, we also show that DCA does not significantly affect the congestion processes that are active over the path. The throughput model analysis from the previous chapter requires this assumption.

- our thesis holds for variations of the TCP/DCA algorithm as well as for significantly different algorithms (e.g., TCP/Vegas).

The simulation analysis is organized as follows. In chapter 7 we develop and present simulation models based on two of the Internet paths that we examined in the previous chapters. We select the paths between NCSU and Emory University (we refer to this as the Emory path) and between NCSU and Arizona State University (i.e., the ASU path). By emulating the dynamics observed over these two paths, we create realistic simulation models allowing us to further explore the problems associated with DCA. We run end-to-end TCP/Reno connections over the simulated paths and extract the *tcpRTT* time series as we did in the measurement analysis. We first show that the dynamics associated with the simulated connections are similar as those that we observed over the measured paths. By comparing the *tcpRTT* time series with the internal router queues levels, we provide a deeper understanding of the problems associated with DCA in a high speed Internet environment.

In Chapter 8. we develop a simulation model that implements the hypothetical DCA algorithm (i.e., the TCP/DCA algorithm) that was the basis of our earlier measurement analysis. We perform simulation experiments based on the Internet path models developed in Chapter 7 and show the following.

- Compared to a TCP/Reno connection, a TCP/DCA flow is able to avoid packet loss (by up to 8%), however the flow will experience throughput degradation on the order of 30%.
- Replacing a TCP/Reno flow with a TCP/DCA flow will not reduce the loss rates at the bottleneck link nor will it significantly lower the average queue level. Intuitively, this makes sense since during periods of packet loss, buffers that are not consumed by a DCA flow will be consumed by the more aggressive competing traffic (e.g., TCP/Reno). Depending on the composition of the competing traffic, queue oscillations and future RTT values are not significantly impacted by the behavior of a single DCA flow. This analysis confirms our conjecture that the reactions by DCA to RTT will not significantly impact the congestion process over the path.

- As the size of the DCA send rate reduction decreases, the amount of throughput reduction decreases because the algorithm becomes less reactive and the algorithm's ability to avoid loss becomes less effective.

In Chapter 9, we further validate our thesis by showing that the DCA algorithms used in TCP/Vegas and TCP/Dual, in addition to not being able to reduce the congestion level within the simulated network, are also not able to improve TCP throughput.

# 7 Validation of the Measurement Results Over Simulated Internet Paths

In this chapter, we develop "realistic" simulation models based on the measured data that was used in the analysis from the preceding chapters. By developing simulation models that accurately emulate the observed dynamics of Internet paths we can extend the DCA analysis by examining congestion dynamics at internal nodes (routers) within the simulation model. The objectives of this chapter are twofold. First, we present simulation models of two of the paths that we used in the measurement analysis and to validate that they reflect congestion dynamics similar to those observed over the measured paths. Second, we provide additional analysis of DCA that was not possible using end-to-end measurement analysis. We show that:

- A TCP constrained RTT congestion probe is too coarse to accurately track the bursty congestion associated with packet loss over high speed paths.
- A DCA algorithm cannot differentiate between the significant number of increases in RTT that are not associated with packet loss from those increases that do lead to packet loss.

Using a method similar to that used in the previous measurement analysis, we obtain the *tcpRTT* time series from an end-to-end TCP/Reno connection over the simulated path. By looking at the *tcpRTT* time series curve along with the queue levels at the bottleneck links during a simulation run, we illustrate that the effectiveness of the abilities of a DCA algorithm to avoid packet loss are limited by:

- Rapid queue growth prior to packet loss with respect to the RTT. The simulation data confirms that the rate of queue increase at a congested router can be less than a RTT making it impossible for a DCA algorithm to react in time to prevent loss.
- For cases when the rate of queue increase prior to loss is one the order of 1-2 RTT's, the coarseness of a TCP constrained RTT probe impacts the ability of a DCA to reliably detect the increase in RTT in time to prevent the loss. The magnitude of the RTT and the packet loss rate impact the coarseness of the TCP RTT probe.

The measurement results suggest that DCA is not able to accurately assess the congestion level of a router. The problems described above contribute to this. The obvious impact to DCA is that a DCA algorithm will frequently react "unnecessarily" to RTT increases that would not lead to packet loss even if DCA did not react. We conjecture that another aspect of this problem that further reduces the effectiveness of DCA is if there are multiple congested routers over the path. If the link capacities and buffer sizes associated with the congested routers are different, the dynamics associated with the congestion will be different. We show through simulation that this is another partial explanation for the large number of loss events that are observed (in both the measured and simulation data) which are not preceded by a significant increase in RTT.

This chapter is organized as follows. The first section explains the methodology used to construct the models. The next section analyzes TCP/Reno flows over the simulated paths demonstrating that the congestion dynamics are similar to the dynamics observed over the real paths. The final section provides the additional analysis stated above.

We preface our discussion of the simulation methodology with a disclaimer. While the models that we present in this chapter are based on measured data, they also rely on a set of assumptions and conjectures. While the end-to-end characteristics of the models are similar to the measured data, we do not know how closely the internal congestion dynamics of the model matches the dynamics of high speed Internet paths. However, even if the model is only marginally accurate, we believe that the results complement and reinforce our measurement analysis of DCA.

## 7.1 Simulation Methodology

To develop realistic models of Internet paths, the following information is necessary:

- Static attributes of the path (number of hops, link capacities, propagation delays)
- Time scales and dynamics of congestion (via the *tcpRTT* time series)

- Throughput behavior
- Packet loss dynamics
- Location of congestion

Our methodology is straightforward. We select two paths from the set of Internet paths that we analyzed in the previous chapters (we selected the Emory and ASU paths). We use *pathchar* and *traceroute* to obtain an estimate of the static attributes of the path. For each path, we select a single run and calibrate the simulation parameters such that the *tcpRTT* time series, throughput and loss dynamics are similar to the measured results. The location of congestion is of interest as it can help us set the background traffic levels that are necessary to emulate the end-to-end dynamics observed over the measured path.

### 7.1.2 Location of Congestion

Figure 7-1 illustrates the *traceroute* output associated with the Emory path (Figure 4-1 shows the *pathchar* output). To help gain understanding of the location of congestion, we traced an *echoping* transfer (i.e., a Unix measurement tool that sends any amount of TCP data to a discard server) over the Emory path [BORT99]. While the TCP transfer is active we also ran three concurrent "fast pings" (i.e., a normal ping without a timeout between transmissions so that 1 ping packet is sent every round trip time) between the NCSU host and three hosts along the path. Figure 7-2 illustrates the results. The upper curve plots the sequence/acknowledgements of the traced TCP connections. The lower curve plots three curves representing the RTT time series from each of the three concurrent pings. Of the three ping plots, the dark curve with the lowest RTT values is the RTT time series between NCSU and router 5. The other dark ping curve represents the RTT time series between NCSU and router 8. The lightest curve is the ping output between NCSU and the destination host located at emory.edu.

We observe from the ping plots that a moderate level of short time scale congestion exists somewhere between NCSU and router 5. Furthermore, congestion that is higher in intensity and larger in time scale exists somewhere between hop 8 and the destination at emory.edu. While other runs show different

congestion dynamics, the key conclusion that we draw from our measurement data associated with the Emory path is that there is more than one location of congestion. Based on Figure 7-2, it seems that the primary location of congestion occurs at the two ISP exchange points (i.e., between hops 4 –5 and hops 10-11).

```
traceroute to electron.mathcs.emory.edu (170.140.150.48): 1-30 hops, 38 byte packets
 1  cmdfhub-5513rsm-1.egrc.ncsu.edu (152.1.191.1)  0.750/0.808/1.15 (0.55) ms  50/50 pkts (0% loss)
 2  ccgw3.ncsu.edu (152.1.1.7)  0.935/1.7/1.76 (0.137) ms  50/50 pkts (0% loss)
 3  ncsu-gw.ncren.net (198.86.71.1)  1.8/1.29/1.79 (0.163) ms  50/50 pkts (0% loss)
 4  rtp7-gw.ncren.net (128.109.243.1)  1.61/2.32/4.46 (0.642) ms  50/50 pkts (0% loss)
 5  Serial0-1-0.GW2.RDU1.ALTER.NET (157.130.36.157)  2.68/8.36/23.1 (4.98) ms  50/50 pkts (0% loss)
 6  142.ATM2-0.XR1.TCO1.ALTER.NET (146.188.163.162)  *  *  11.6/16.3/39.5 (4.65) ms  48/50 pkts (4% loss)
 7  193.ATM2-0.TR1.DCA1.ALTER.NET (146.188.161.162)  *  10.0/13.6/18.1 (2.17) ms  49/50 pkts (2% loss)
 8  101.ATM6-0.TR1.ATL1.ALTER.NET (146.188.136.13)  23.0/26.6/32.7 (2.3) ms  50/50 pkts (0% loss)
 9  299.ATM6-0.XR1.ATL1.ALTER.NET (146.188.232.89)  22.8/26.9/35.1 (2.36) ms  50/50 pkts (0% loss)
10  195.ATM9-0-0.GW2.ATL1.ALTER.NET (146.188.232.65)  23.4/26.8/32.7 (2.24) ms  50/50 pkts (0% loss)
11  emory-gw.customer.alter.net (157.130.65.46)  24.4/29.8/52.7 (4.1) ms  50/50 pkts (0% loss)
12  ndg-bbatm.netops.emory.edu (170.140.2.190)  *  26.2/30.3/37.7 (2.53) ms  49/50 pkts (2% loss)
13  electron.mathcs.emory.edu (170.140.150.48)  * * * * * * * * * * * * * * * * * * * * * * * * * * * * 26.6/39.9/71.8 (11.8)
ms  25/50 pkts (50% loss)
```

Figure 7-1. Traceroute between NCSU and emory.edu

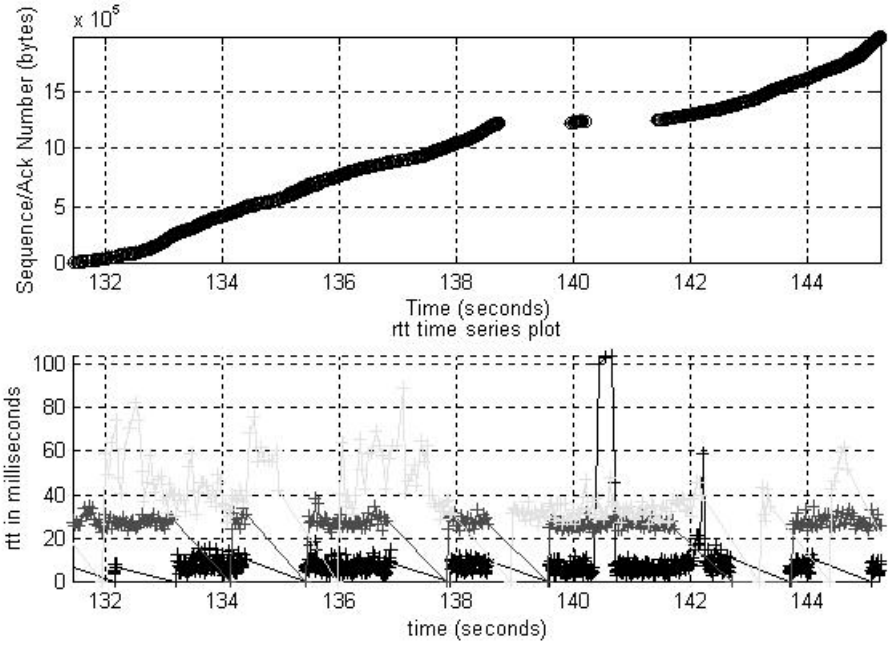


Figure 7-2. Concurrent ping between NCSU and Emory

Figure 7-3 illustrates the *traceroute* output for the ASU path. The minimum RTT time that we observed over the ASU path was 68ms. Figure 7-4 illustrates the concurrent ping measurement results. Comparing

the top curve with the sequence/ack curve of the Emory run shows the extreme difference in dynamics. The ASU path suffers from sustained congestion and high loss rates. Looking at the lower curve of Figure 7-4, the dark RTT time series curve that has the lowest RTT values represents the ping output between NCSU and router 8 (i.e., the Sprintlink.net connection to the MAE). The other dark time series curve represents the ping between NCSU and router 10. The lightest time series plot is the ping output between NCSU and the ASU destination. We see a fairly high level of congestion somewhere between NCSU and router 8. There is a significant level of sustained congestion between router 8 and router 10. There appears to be minimal additional congestion between router 10 and the ASU host. Based on this data as well as other measurement data, we conclude that (as with the Emory path) there are multiple locations of congestion and loss. For the ASU model (which we present in the next section), we assume that links 4-5 and 10-11 are moderately congested, link 8-9 experiences sustained congestion and link 7-8 experiences bursty congestion (designed to cause loss).

```
1  cmdfhub-5513rsm-1.egrc.ncsu.edu (152.1.191.1)  1.167 ms  0.847 ms  0.908 ms
2  ccgw3.ncsu.edu (152.1.1.7)  2.324 ms  2.955 ms  1.923 ms
3  ncsu-gw.ncren.net (198.86.71.1)  2.370 ms  8.981 ms  3.613 ms
4  rtp7-gw.ncren.net (128.109.243.1)  13.821 ms  13.084 ms  19.338 ms
5  sl-gw9-dc-12-0-T3.sprintlink.net (144.228.128.9)  11.970 ms  10.027 ms  9.569 ms
6  sl-bb10-rly-3-0.sprintlink.net (144.232.7.185)  9.794 ms  9.716 ms  11.271 ms
7  sl-bb4-dc-0-0-0.sprintlink.net (144.232.7.154)  12.229 ms  14.938 ms  10.474 ms
8  sl-e2-mae-0-1-0.sprintlink.net (144.228.10.42)  12.649 ms  13.917 ms  17.548 ms
9  mae-east.good.net (192.41.177.101)  45.117 ms  104.443 ms  168.903 ms
10  phx-vienna.phoenix.good.net (209.141.210.61)  85.238 ms  69.190 ms  81.990 ms
11  phx-co-ds3-2.phoenix.good.net (209.141.210.37)  103.253 ms  82.448 ms  89.526 ms
12     asu-oc3.phoenix.good.net (209.54.110.46)  77.604 ms  80.159 ms *
13     gold-fddi.inre.asu.edu (129.219.152.2)  70.338 ms  71.377 ms  86.959 ms
14     www.eas.asu.edu (129.219.30.21)  82.887 ms *  85.562 ms
```

Figure 7-3. Traceroute between NCSU and ASU

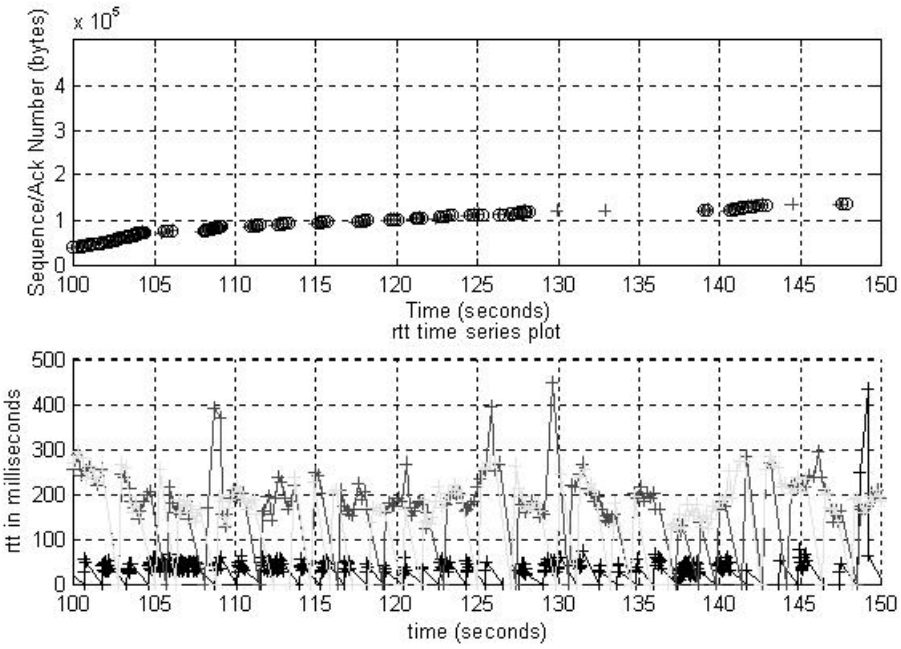


Figure 7-4. Concurrent ping between NCSU and ASU

### 7.1.2 Developing the Models

Both the Emory and ASU paths have roughly the same number of hops and traverse about the same number of ISP's (the ASU path includes 3 ISP's while the Emory path includes only 2). Based on the IP addresses shown in the *traceroute* output, we are able to locate the ISP exchange points (referred to as Network Access Points or NAPs). The starting point for both paths is the generic network model illustrated in Figure 7-5. Attached to each router are a number of sources and sinks. Figure 7-6 illustrates 3 routers, each configured with 16 sources and 16 sinks. Using router 6 as an example, the sources are referred to as xs6-1 through xs6-16 and the sinks are referred to as xd6-1 through xd6-16.

We will show in the next section the details of the Emory and the ASU models. In general, the TCP connection that is under observation flows between the hosts hs1 and hd1 as illustrated in Figure 7-5. The sources and sinks are connected to the routers with either a 10mbps or a 100mbps link. The background traffic consists of a mix of TCP and UDP traffic. Both TCP and UDP data streams are

generated by pareto traffic sources which are able to emulate realistic traffic patterns [PAXS95]. The pareto sources attached to the TCP cross traffic connections are configured to emulate HTTP traffic. As an example, a typical setting for a TCP cross traffic source that is connected via a 10 mbps link to a router is:

*Burst rate: 10mbps*

*Burst time: 15 ms*

*Idle time: 60 ms*

The pareto shape parameter (which sets the level of burstiness associated with the source) is typically 1.2. A burst at 10mbps for 15 ms corresponds to roughly 12 packets (of size 1448 bytes) which is reasonable for an HTTP GET command. The pareto distribution, by definition, will make the size of the burst vary widely. As we will see in the next section, the purpose of the pareto/UDP flows is to produce large bursts of packets in an effort to emulate the short time scale, high intensity bursts of RTT that were observed in the measured data. Additionally, high bandwidth pareto/UDP flows are used to generate varying levels of packet loss.

```
Tcp    ------------------------------------------------------------------------------------------------------->>> Tcp
Sender                                                                                                            Receiver
hs1 - r1  ----   r2 ---- r3 ----(access)---- r4 ---- r5 ---- r6 ---- r7 ----(NAP)---r8 ---- r9 ---- (access)--- r10 - hd1

          Source Network                    (ISP 1)                          (ISP 2)                Dest network
```

Figure 7-5. Generic network model

```
xs6-1 …xs6-16                    xs7-1 …. xs7-16                xs8-1 … xs8-16
          |                              |                              |
     ---  r6  ---------------------------- r7 ---------------------------- r8 ---
          |                              |                              |
xd6-1 … xd6-16                   xd7-1  …. xd7-16               xd8-1 … xd8-16
```

Figure 7-6. Example router nodes and background traffic sources and sinks

## 7.2 Validation of the Models

Focusing on one simulation run from each model, we duplicate the analysis that was used in the measurement analysis. We trace a TCP/Reno connection over the path and extract the *tcpRTT* time series. We run the correlation metrics defined in the Part 1 of the dissertation and show that the results are similar to the measured data results. Rather than trying to exactly duplicate the dynamics of the measured connection, we design the model such that the end-to-end dynamics are *feasible* and could possibly be observed over the real path.

In this section, we present the Emory and ASU simulation models. We show how well the dynamics associated with the simulation model match the measured data. By examining internal router queues, we see quite clearly the problems faced by a DCA algorithm.

### 7.2.1 Emory Path Model

Figure 7-7 illustrates the Emory simulation model. Based on the *traceroute* output, we learn that there are 13 hops along a path that contains 4 domains and 2 ISP's. The top line labeled *bs* indicates the buffer size (in packets) of each router. The *pd* line indicates the propagation delay associated with the link and the *lc* line indicates the link capacity (in mbps). In the figure, h1 represents the location of the TCP sender that is under observation and hd1 is the location of the corresponding TCP sink. The uncongested RTT is roughly 27 milliseconds. The prior measurement work has shown that the loss rates are typically low (in the .5% to 1% range) although the path exhibits a significant level of packet variation.

Based on the concurrent ping results we assume that the location of significant queue buildup occurs at links 4-5 and 10-11. We also assume that low levels of loss occur at one of the 155 mbps links. While the link capacity estimates obtained from *pathchar* are inconclusive, we assume all interior ISP hops are 155

mbps and the ISP access points and exchange points are 45mbps. Router buffer sizes are set to 200 packets with the exception of link 4-5 which is set to 500 packets.

Roughly 150 pareto/TCP traffic generators are located throughout the network. Several high bandwidth pareto/UDP flows are established to produce bursty congestion at links 4-5, 7-8 and 10-11. The intensity of the pareto/UDP flows (i.e., the burst rate and burst time) is a major factor in the packet loss dynamics at these two links. We found it difficult to reproduce the observed RTT variations using a relatively small number of TCP flows (i.e., hundreds). Thousands of low bandwidth flows are necessary however large scale simulation requires significant processing capabilities. Pareto/UDP traffic generators allow us to model realistic traffic levels while not demanding excessive processing requirements.

```
bs  200     200    200    200     500    200    200      200  200     200    200      200      200
 pd .1ms   .1m     1ms    1ms    .2ms   2ms    2ms      2ms  2ms    .4ms   2ms     .5ms     .1ms
 lc 10m   155m    155m   155m    45m   155m   155m     155m 155m   155m    45m    155m      10m
h1 ----- r1 ---- r2 ----- r3 ---- r4  ----- r5 ----- r6 ----- r7 ----- r8 ----- r9 ----- r10 ----- r11 ----- r12 ----- dh1
h2 --|            |                   |                                                |                        |-- dh2
h3 --|            |                   |                                                |                        |-- dh3
       | ncsu.net  |      ncren        |                          alter.net             |      emory.net    |
```

Figure 7-7. Simulation model of NCSU to Emory path

Figure 7-8 illustrates the measured Emory run that we want to model. This run, which was described in Chapter 4, is typical of the path (although increases in RTT samples and loss events appear less correlated in this run that in other runs over the path). The measured loss rate is about 1%. Figure 7-9 illustrates the corresponding simulation run. The end-to-end TCP/Reno connection under observation is configured similarly to the TCP stack used in the measurements. The maximum window size is limited to 12 packets with a segment size of 1448 bytes. The flow consists of one-way ftp data. The sink is configured to perform delayed acknowledgements. The cross traffic is a combination of TCP and UDP sources as described above. The loss rate at link 4-5 is 1.5% and the loss rate at the second lossy router (i.e., link 7-8) is .86%. The loss rate experienced by the TCP connection under observation is .52% which explains why the throughput exhibited in Figure 7-9 is higher than the measured results shown in Figure 7-8.

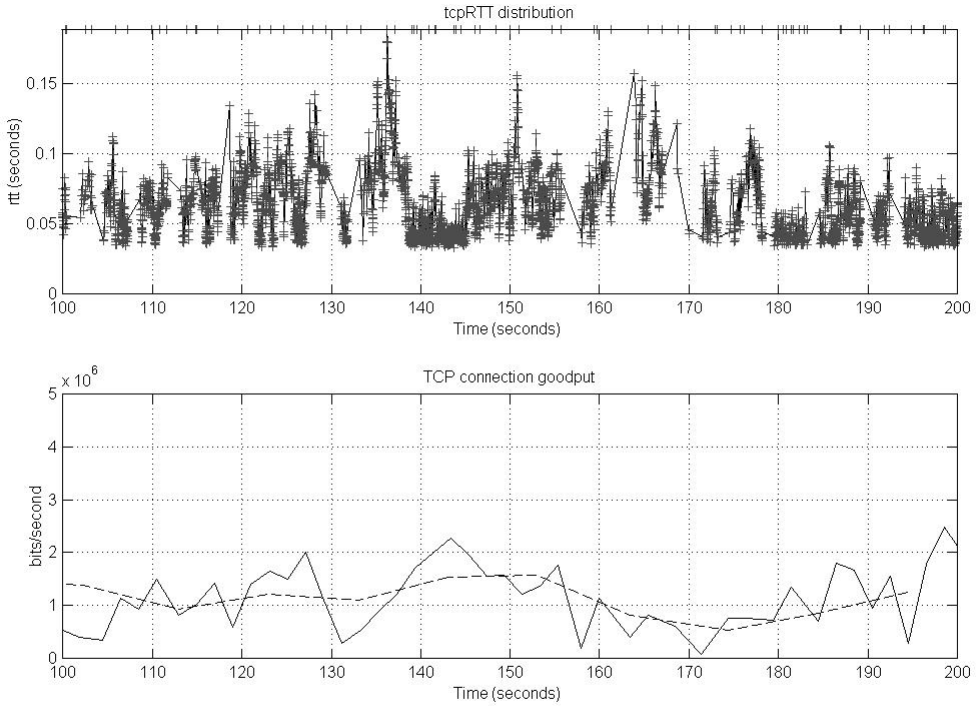


Figure 7-8. Measured Emory path results

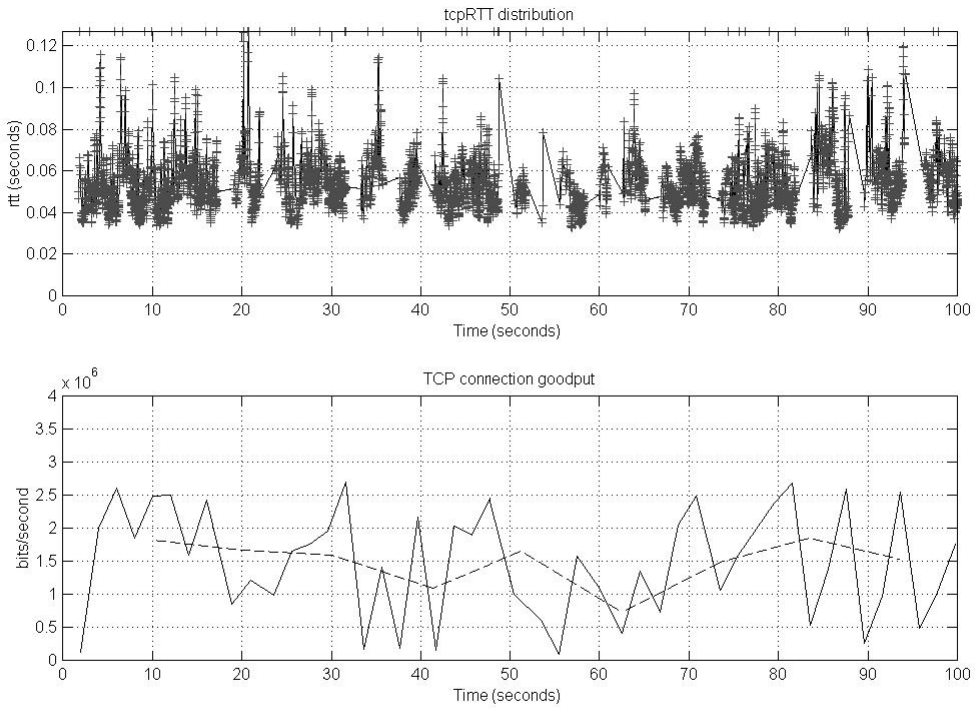


Figure 7-9. Simulated Emory path results

Comparing Figures 7-8 with 7-9 we see that the magnitude and time scale of the congestion experienced by the simulated connection is roughly similar to that of the measured connection. The results of the correlation metrics on the simulated data confirms that the congestion dynamics exhibited by the model is similar to the measured results. Clearly the observed behavior of an end-to-end connection over the simulated path will not match the measured results exactly. Our goal is simply to develop a simulation model which is capable of generating dynamics that might possibly be observes over the real path.

Figures 7-10 and 7-11 illustrate the loss conditioned delay correlation metric and loss conditioned CDF metric respectively. The loss conditioned delay correlation metric results reflect a higher level of correlation between loss and increases in the *tcpRTT* samples as compared to the measured run (Figures 5-3 and 5-4). The metric also indicates that the *tcpRTT* samples associated with packets that are sent after the transmission of the packet that is dropped reflects a larger increase in delay. The simulation results in

Figure 7-10 actually more closely match the metric as applied to the aggregate Emory data illustrated in Figure 5-11.

The loss conditioned CDF metric applied to the simulation data (Figure 7-11) is also similar to the aggregate data (Figure 5-18) although the simulation results show a tendency for loss to occur at slightly larger RTT values. The correlation indication metric values applied to the simulation data are also similar to the aggregate results shown in Table 5-4:

*P[sampledRTT(2) > windowAVG(5)] = .43*

*P[sampledRTT(2) > windowAVG(20)] = .41*

*P[sampledRTT(2) > windowAVG(20) + std] = .13*

*P[sampledRTT(2) > rttAVG] = .50*

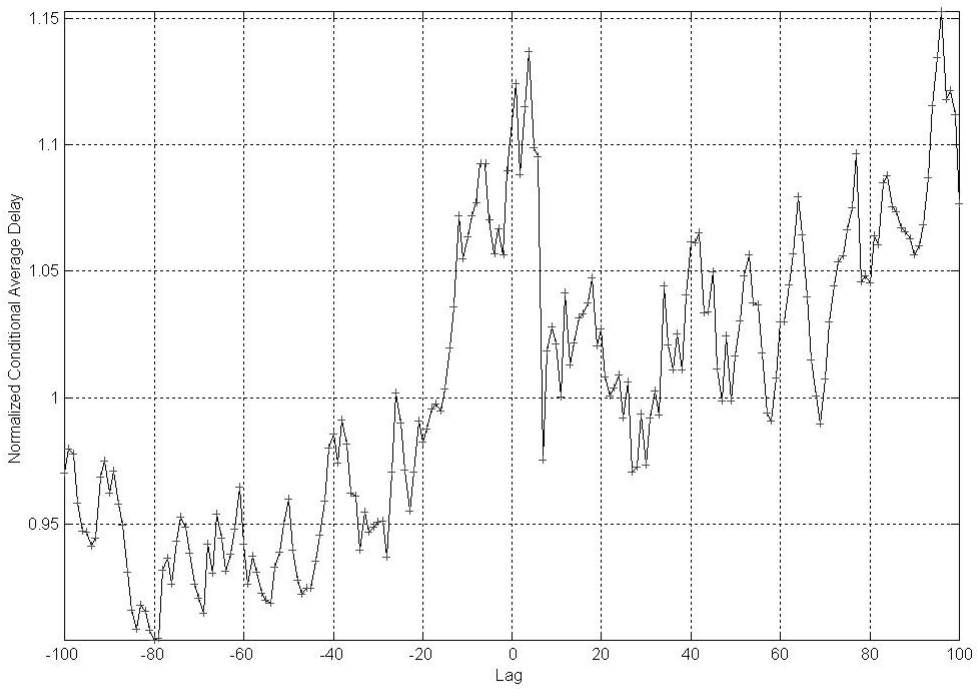


Figure 7-10. Loss conditioned delay correlation metric for Emory simulation

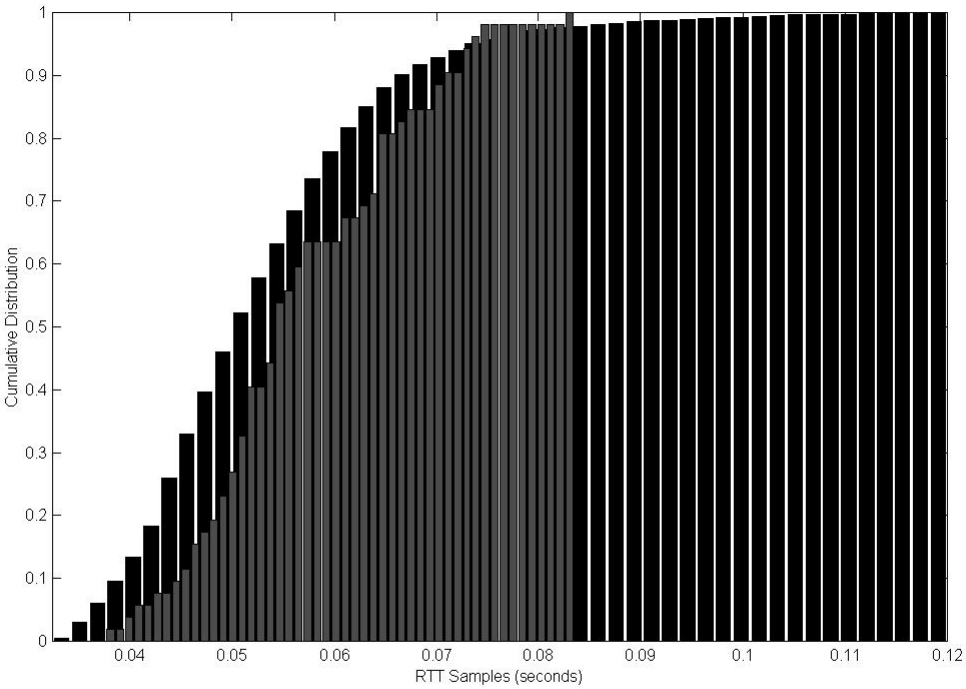


Figure 7-11. Loss correlated CDF for Emory simulation

### 7.2.2 ASU Path Model

Figure 7-12 illustrates the ASU network simulation model. Based on the *traceroute* output, we learn that there are 15 hops along a path that contains 5 domains and 3 ISP's. The uncongested RTT is roughly 67 milliseconds. The path exhibits very high loss rates. The figure illustrates the router buffer size (the *bs* row), the propagation delay (the *pd* row) and the link capacity (the *lc* row). We set the background traffic levels such that there is moderate congestion at links 4-5 and 10-11, sustained congestion at link 8-9 and bursty congestion at link 7-8.

```
bs          200    200     200     900     200     200     200     600     200     900     200     200    200     200
pd  .1     .5ms    .1m    .25ms   3.5ms   .07ms   .5ms    .5ms    .9ms   25ms    .5ms    .4ms   .25ms .5ms    .1ms
lc  10m   155m    155m   155m    45m     155m    155m    155m   100m   155m    45m     155m   100m  100m    10m
h1 ----- r1 ---- r2 ----- r3 ---- r4  ----- r5 ----- r6 ----- r7 ----- r8 ----- r9 ----- r10 ----- r11 ---  r12 --- r13 --- r14 ---- dh1
h2 --|            |                    |                                |                                |              |-- dh2
h3 --|            |                    |                                |                                |              |-- dh3
     | ncsu.net   |      ncren         |          sprintlink.net        |            good.net            |  asu.edu     |
```

Figure 7-12. Network model of NCSU to ASU path

Figure 7-13 illustrates the measured run over the ASU path that we attempt to model (this is the same run discussed in Figures 5-6 through 5-8). The loss rate is roughly 7%. Figure 7-14 illustrates the equivalent results for the simulation run. The loss level experienced by the TCP connection (which flows between h1 and dh1) is also roughly 7%. Loss occurs at link 7-8 and at link 8-9. To generate bursty congestion at link 7-8, we use a similar approach as we did in the Emory model where the background traffic is a combination of TCP flows along with a high bandwidth, bursty pareto/UDP flow. We used several hundred low bandwidth TCP connections to create sustained congestion over the 100mbps bottleneck link between routers 8 and 9.

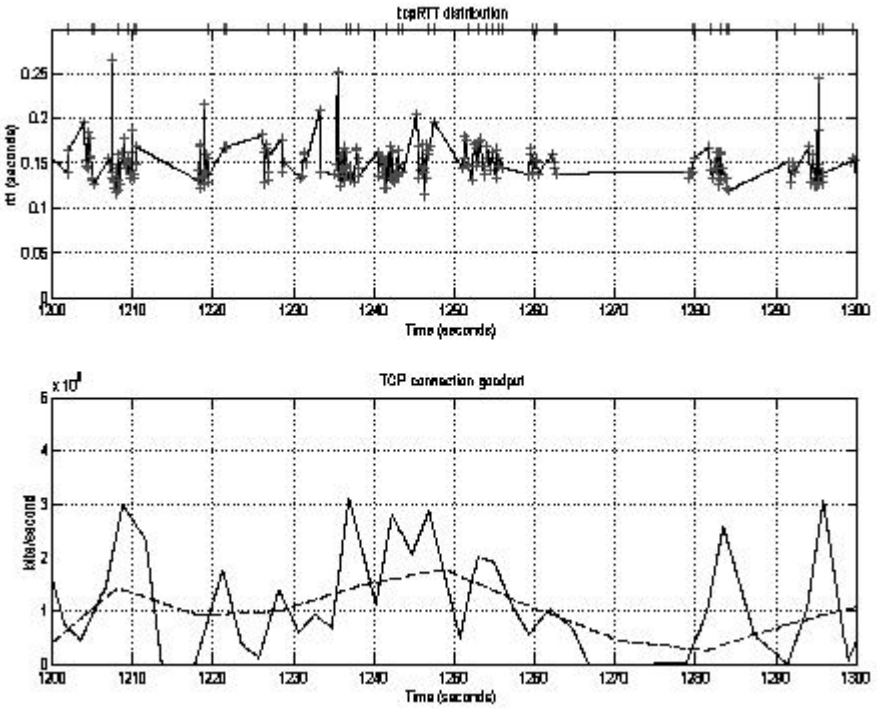


Figure 7-13. Measured dynamics over the ASU path

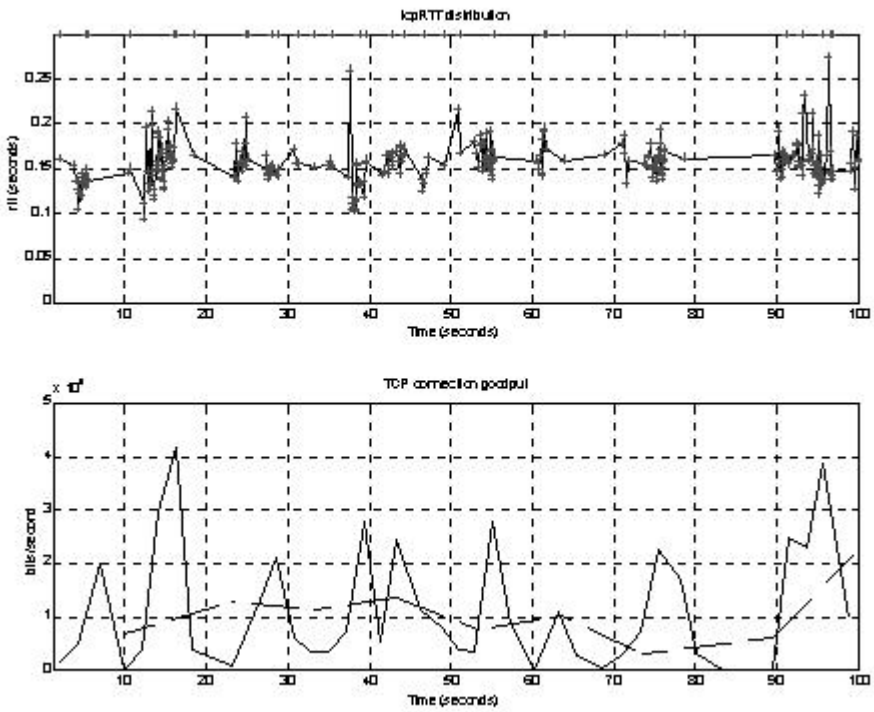


Figure 7-14. Simulated dynamics over the ASU path

The correlation indication metric results for the simulation run are:

$$P[sampledRTT(2) > windowAVG(5)] = .30$$

$$P[sampledRTT(2) > windowAVG(20)] = .37$$

$$P[sampledRTT(2) > windowAVG(20)+std] = .10$$

$$P[sampledRTT(2) > rttAVG] = .36$$

The metric results shown above are similar to the results from the measured run associated with Figures 5-6 through 5-8. Figure 7-15 shows the loss conditioned correlation delay metric results for the simulation run. Comparing Figure 7-15 with the metric results from the measured run (Figure 5-7), we see a similar level of "random" behavior indicating weak correlation. The measured run does exhibit a slightly stronger level of correlation after lag 0. Figure 7-16 is very similar to the measured CDF results. Both show a similar *tcpRTT* distribution, although the simulation had a greater number of low *tcpRTT* values. In both cases, we see that effectively loss is as likely to occur as any *tcpRTT* value.

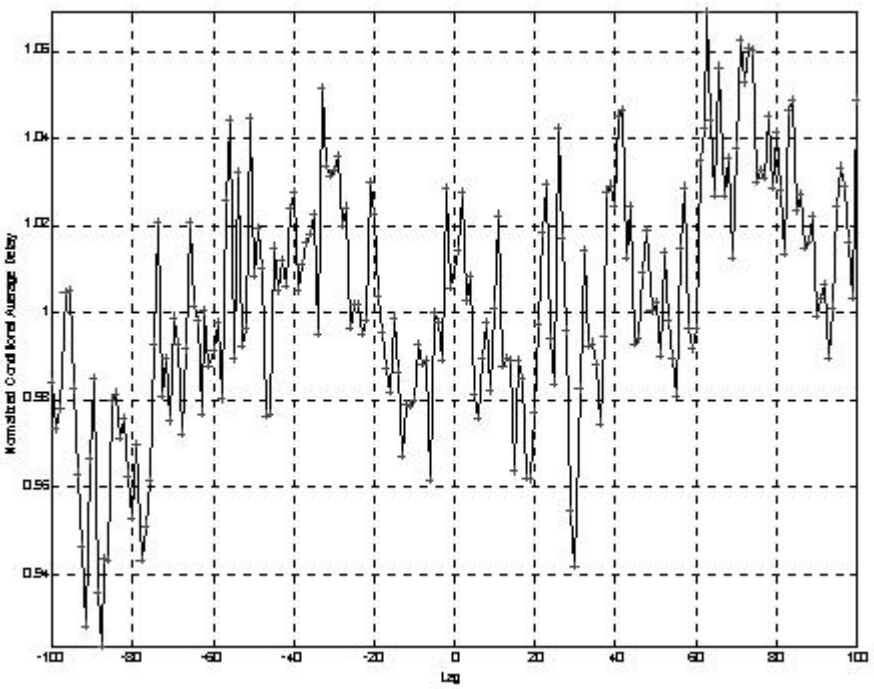


Figure 7-15. Loss conditioned correlation delay metric for the simulated ASU run

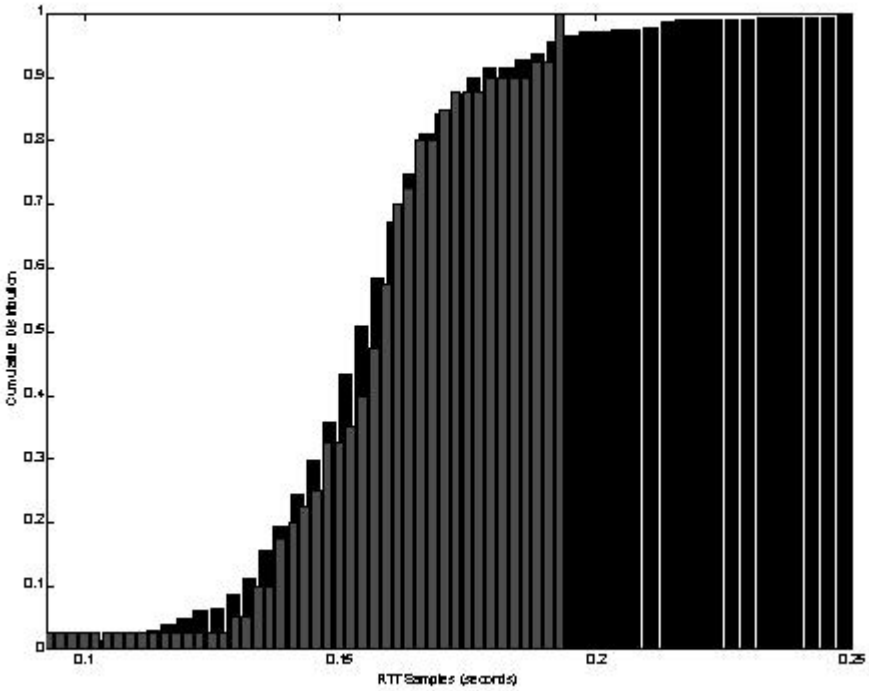


Figure 7-16. Loss conditioned CDF metric for the simulated ASU run

## 7.3 Potential Difficulties for DCA

We examine in detail one simulation run from the Emory model and one run from the ASU model. We show that when the time scale associated with the queue increase is on the order of 1-2 RTT periods, the bursty dynamics of TCP degrades the sampling abilities of the TCP RTT probe. When the time scale of the queue increase that precedes a loss event is on the order of one RTT, we show that DCA is not able to react in time to prevent the loss. Finally, we show the impact to DCA when there are multiple routers along the path experiencing loss.

### 7.3.1 Problems over the Emory Path

Figure 7-17 plots a portion of the *tcpRTT* time series from the Emory simulation run (illustrated in Figure 7-9) along with the queue levels from the two lossy links (links 4-5 and 7-8 respectively). In the queue level plots, the solid line represents the maximum queue level observed during a sample period (.1 seconds) and the dashed line is the minimum queue level during the period. The congestion that occurs at both routers is bursty as evidenced by how seldom the minimum queue level curve detects congestion (i.e., congestion with a time scale greater than .1 second). The queue delay associated with congestion at link 4-5 dominates (compared to the delay associated with the faster link 7-8). The time scale associated with congestion at link 4-5 is between .05 and .1 seconds (which is roughly 1-2 RTT's) which allows the end-to-end probe to detect the queueing. The queue delays associated with link 7-8 are significantly lower in magnitude than those associated with link 4-5 because the link capacity is larger (155mbps versus 45 mbps) and because the maximum queue level is smaller (200 packets versus 500 packets). There are two loss events captured in the top curve of Figure 7-17. The first loss (at time 25.7 seconds) occurs at link 4-5 and the second loss (at time 26 seconds) occurs at link 7-8.

The first loss reflects a scenario that we believe occurred frequently in our traced connections. It demonstrates that a TCP constrained RTT probe is unable to detect the queue buildup that precedes packet loss in time to be able to react and avoid the loss. The *tcpRTT* sample at the time of the transmission of the segment that is eventually dropped reflects only a small increase in RTT (at time 25.52 seconds). The next sample (at time 25.55 seconds) reflects the queue buildup associated with the congestion that will eventually lead to loss. However, this congestion indication arrives at the sender 1 RTT too late. This example shows a loss event that is preceded by a significant increase in RTT however several factors increase the coarseness of the TCP constrained sampling process making it difficult to detect the increase in RTT in time to avoid the loss.

- The number of probe packets is limited by TCP's congestion control algorithms. So, during times of congestion, the number of probes tends to reduce.
- Because of TCP's bursty behavior, probes tend to be sent "clumped" together every RTT.
- As the queue size increases, the congestion indication feedback time grows with the increase in RTT further affecting the sampling.

In this case, even if the TCP RTT probes were more evenly spaced, the congestion decision would most likely still have occurred to late to avoid the loss because of the rate of the queue buildup. However, the point is that aspects of TCP's congestion control algorithm will tend to increase the coarseness of the RTT congestion probes reducing the effectiveness of DCA.

The second loss event shown in Figure 7-12 demonstrates that a TCP RTT congestion probe is not able to accurately assess the congestion level of a router. The loss occurs at the 155 mbps link 7-8. The queue delay at this link is essentially undetectable compared to the queue delay associated with the 45mbps link 4-5. For example, the waiting time experienced by a packet at router 4 when there are 400 packets in the queue is roughly .1 seconds while the waiting time experienced by a packet at router 7 when there are 150 packets waiting is an order of magnitude lower (.01 seconds). Clearly this presents a problem for an end-to-end probe algorithm trying to predict future packet loss events based on RTT fluctuation. The problem

becomes worse as the loss rate increases at a router that has higher link capacities and/or lower buffer sizes relative to other congested links.

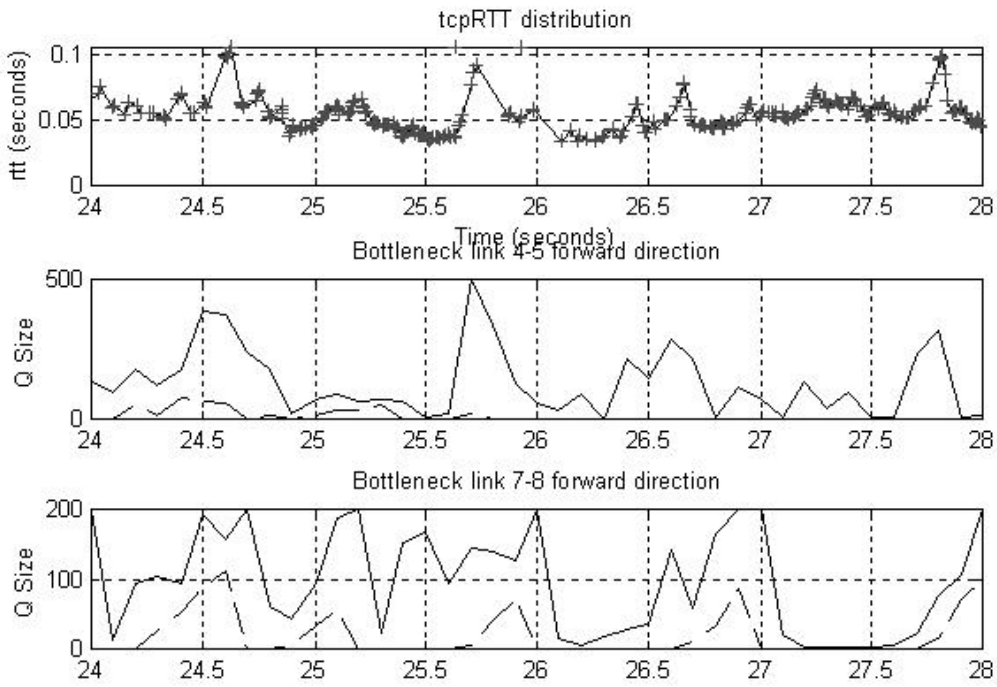


Figure 7-17. *tcpRTT* time series and bottleneck link queue for Emory simulation run

### 7.3.2 Problems over the ASU Path

Figure 7-18 plots the *tcpRTT* series from the ASU simulation run (the same run from the preceding section) along with the queue levels from the two lossy links (link 7-8 and link 8-9 respectively). The top curve illustrates that 4 loss events occur (i.e., indicated by the hash mark at the top of the curve). At least 3 of the 4 loss events occur at link 8-9 (it is unclear if the third loss at time 46.95 seconds occurs at link 7-8 or link 8-9). Due to the high loss rates, the window size is typically 1-3 packets (regardless of the value of the maximum window size). As can be seen in the *tcpRTT* time series curve, the RTT is quite large. Consequently, the probe frequency is infrequent (i.e., every .12 seconds).

It is extremely difficult for DCA to operate effectively in this environment for two reasons. First, because of the coarseness the probe due to the low TCP throughput and the very large RTT. A DCA algorithm would have a difficult time reacting to the congestion indication in time to prevent loss. Second, the increase in the queue level that accompanies loss (especially for loss that occurs at link 7-8) is not large enough to allow a DCA algorithm to reliably predict future loss events. The next example further illustrates these points.

Figure 7-19 expands Figure 7-18 between the time 46 and 48 seconds. The router queue level curves (the second and third plot) plots the maximum and minimum queue level during an interval of .01 seconds. The burstiness associated with the congestion at link 7-8 is more evident. The extreme coarseness of the DCA probe is also evident. The dropped packet is originally transmitted at time 46.9 seconds. The *tcpRTT* sample that is generated at this time does not reflect a significant increase in queue growth (as compared to the last several samples). The packet is dropped at either link 7-8 or link 8-9 (we cannot tell from the plot). In either case, the time scale associated with the queue increase that precedes loss is less than one RTT. The *tcpRTT* sample at time 47.25 seconds reflects a significant increase in queueing. This sample is generated from a packet that was transmitted close to the time that the dropped packet was originally sent. The probe therefore measures the congestion associated with the loss, however it clearly arrives 1 RTT too late. It is unclear if the probe experienced congestion at both links (i.e., the congestion epoch at link 7-8 at time 47.1 and the congestion at link 8-9 at time 47.15). Even if the probe does wait in the queue at link 7-8, the additional delay is minor compared to the delays associated with link 8-9. The point is that it is not possible to avoid packet loss when the time scale associated with the queue buildup preceding loss is less than 1 RTT period.

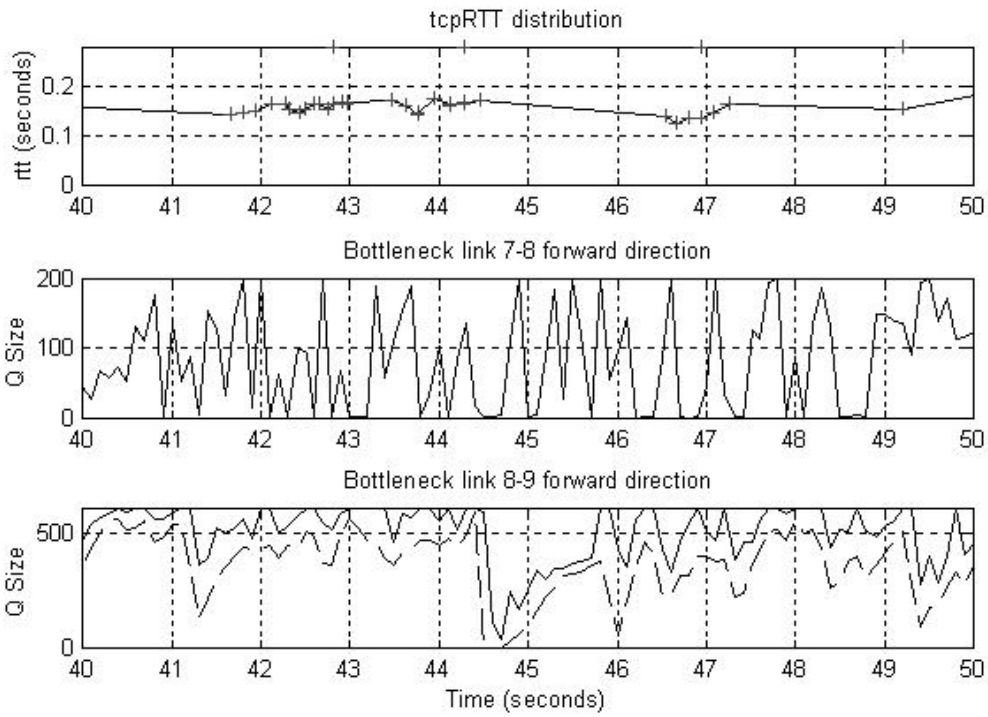


Figure 7-18. *tcpRTT* time series plotted with 2 bottleneck queues from the ASU simulation run

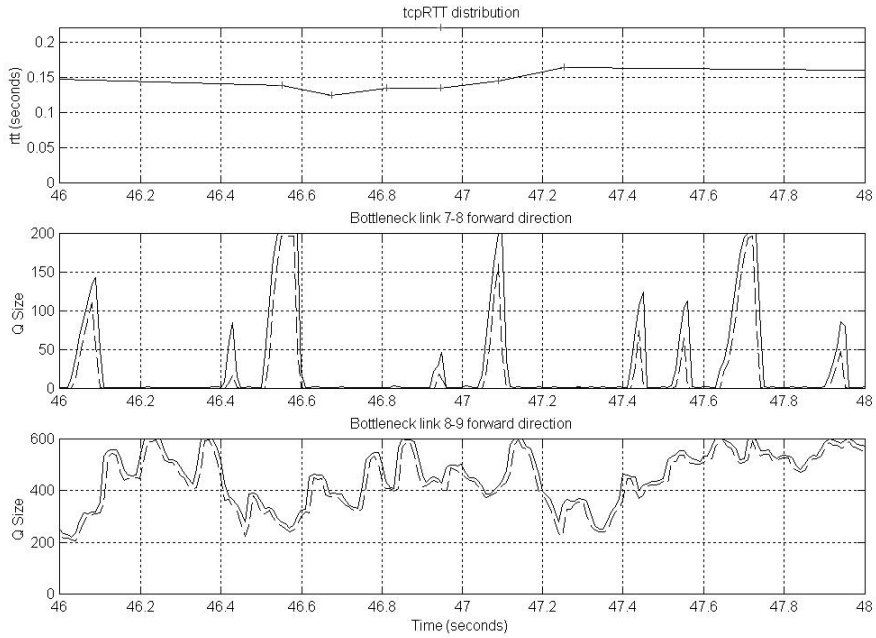


Figure 7-19 Bottleneck queues plotted using more frequent samples