Estimating Traffic Parameters in Internet via Active Measurements for QoS and Congestion Control

Vinod Sharma Suma M.B
Electrical Communication Engineering Dept.
Indian Institute of Science, Bangalore-560012, India

Abstract - **In this paper we develop a scheme which can help provide certain QoS via active measurement techniques. We send a probing stream, using the TCP or the UDP protocol, on a path with one bottleneck router. Several TCP and UDP connections (cross traffic) may be sharing the link on the router. We make measurements on the RTT and the throughput of the probing stream and do not assume any knowledge of the cross traffic. Then we obtain estimates of the mean delay and the throughput a new TCP or UDP stream will obtain through that router. We also obtain the total traffic intensity of the UDP streams and the total work load in the queue by the TCP connections in the cross traffic. The router may or may not employ RED control. We find that using TCP for a probing stream provides better estimates than using the UDP.**

Key words: Network monitoring, estimating traffic parameters, local information, providing QoS, Internet.

1. Introduction

Internet was designed to provide only best effort service. However recently efforts are being made to guarantee certain quality of service (QoS) for some applications (see Black, et al. [1] and Braden et al. [2]). For example, some data applications (e.g. web browsing, file transfer, interactive computation) require certain upper bounds on the delays which can be guaranteed by providing a minimum throughput. In addition real time applications, e.g. IP telephony, are being envisaged which require upper bounds on end-to-end delay and probability of packet loss. Therefore, Diffserv and Intserv architecture ([1], [2]) have been designed to provide various levels of QoS. An essential element in any such architecture will be to search for an appropriate route for a new connection request. The route should be such that the routers on it have the resources to guarantee the QoS requested by the new connection (QoS routing). If no such route exists, the connection request may be denied. Since the congestion at various links in the network varies with time, one feasible approach is to make measurements at the time of a connection request to check for the suitability of a particular route.

Monitoring the network and measurements on it are being used for various other purposes: congestion control, network management and fault detection. Therefore, a large number of protocols and measurement techniques are being used to serve different purposes (see Adams et al. [3], Caceres et al. [4], CAIDA [5] Paxson et al. [6]). The measurement techniques are classified broadly into active measurements and passive measurements. In passive measurements a device on a link or a router measures passively the different packets passing through that link/router. Using these measurements, one can obtain the amount of traffic passing through that link (router), the classification of traffic into different categories and whether the link is live or not etc. The active measurements are made via sending a probing stream through a part of the network. Examples of it in the Internet are the PING and the TRACEROUTE tools. These have been used to check for network connectivity, probability of packet loss and round-trip time (RTT). However using the analytical tools developed in Adams et al. [3], Sharma and Mazumdar [7], Sharma [8] and Alouf et al.[9] more detailed information can be obtained (e.g. traffic intensity at different nodes). This work is another attempt to provide more analytical tools in this endeavor.

In this paper we continue the work in [8]. In [7],[8] and [9], finite and infinite buffer networks were considered where the traffic streams arriving were Poisson, renewal or general stationary and they were passing through various queues. However in the Internet there are feedback controlled TCP connections along with UDP connections (which can be represented by Poisson, renewal etc streams). The feedback controlled systems can behave very differently than the other systems. Therefore the techniques, developed in [7],[8] and [9], do not work in the TCP/IP based internet. (However it is shown via simulations in [9] that these techniques may work to some extent when a large number of TCP connections share a queue). We develop new methods in this paper which enable us to decide whether sending a new TCP or a UDP connection through a queue on a bottleneck router will provide the required mean delay and/or throughput. We also determine the W_{max} (the maximum window size) the new TCP connection will need to obtain the throughput it requires. In addition, we estimate the mean delay the *existing* connections on the bottleneck queue will face if the new connection will pass through that queue (router). Our results also provide the total mean workload in the queue by the TCP connections and the total traffic intensity of the UDP streams in the cross traffic. These provide important information about the congestion at the router. To obtain this information, we do not need to know anything about the number or the parameters of the cross traffic on the bottleneck queue. We will use the new formulae developed in Sharma and Punyaslok [10] for delay and throughput of a traffic stream passing through the routers shared by various TCP and UDP connections.

Theoretically we can use a TCP or a UDP connection for a probing stream. But we have found through simulations that a TCP probing stream provides more accurate estimates of quantities of interest. Also, because of feedback control, a TCP probing stream may cause less congestion than a UDP

0-7803-7206-9/01/$17.00 © 2001 IEEE

probing stream. We have also extended our method to the case where the bottleneck queue is controlled by the Random Early Detection (RED) algorithm (Floyd et al. [11]).

Although in this paper we report our results on a single queue, our results can be generalized to a tandem of queues. The accuracy of estimates reduces as the number of queues increases. But, we have verified via simulations, that the estimates are good for two queues with small propagation delays. Due to lack of space the multiple queue results are not reported here.

The paper is organized as follows. In section 2 we develop the method for the system with the bottleneck queue having an infinite buffer and it is shared by various TCP and UDP connections. Section 3 extends the results to the router with RED control. Section 4 provides simulation results to verify the accuracy of our estimators.

2. Queue With Multiple Tcp And Udp Connections

In section 2.1, we describe the basic model and the underlying theory for our approach. Section 2.2 describes the methodology to estimate the parameters of interest. In section 3 we extend the results to the RED controlled routers.

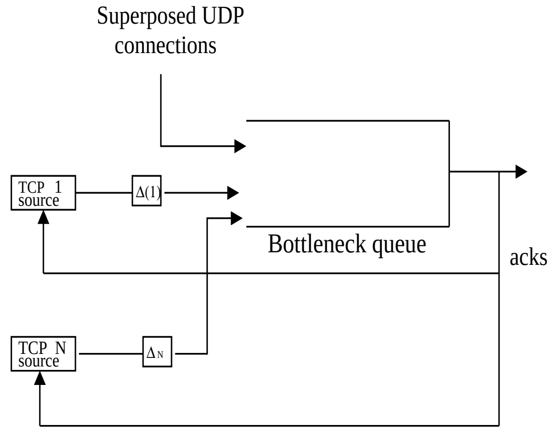

Figure 1: System under consideration

2.1 *The model and basic theory*

We consider a queue at the output link of a router (Fig 1). The buffer length at the queue is assumed infinite in this section. In the next section, this assumption will be augmented with an active queue management strategy. The queue is shared by several TCP and UDP connections. We do not assume any knowledge of the number of these connections or their parameters (e.g. traffic intensity or packet length distribution etc.) In the paper we will concentrate only at the above mentioned queue of the router, even when we use the word router. This router represents the bottleneck router on the route of interest for a new TCP or UDP connection request. To check for the suitability of this route, a probing stream is sent. The probing stream itself can use the TCP protocol or the UDP protocol. In case it uses UDP, we will take it to be a Poisson stream. This causes no loss of generality because the probing stream is in the controller's control. The throughput (goodput) and the round trip time (RTT) of the packets of the probing stream are measured. In the following, we describe our method to estimate the goodput and the mean delay the new TCP request will obtain for any given W_{max} Poisson or a Markov Modulated Poisson Process (MMPP), we will estimate the mean delay it will experience. We will also obtain an estimate of the extra delay the new stream will cause to the *existing* connections on the bottleneck router. By sending that information to the appropriate controllers, it can be decided if the existing connections will continue to receive their QoS. This way it can be decided if the new connection should be admitted on the route under consideration.Our results also provide estimate of the workload being brought in by the TCP and the UDP cross traffic at the router. This information can be used for congestion control in the network.

To estimate the parameters of interest mentioned above, we use the formulae obtained in Sharma and Punyaslok [10]. Let there be N TCP connections passing through the queue. The ith connection has maximum window size $W_{max}(i)$, total propagation delay (which may include the processing delays at the routers) $\Delta(i)$ and the packet lengths iid with a generic length $s(i)$. The TCP protocol used may be Tahoe or Reno (our simulation results later are only for Reno). There can be several UDP streams passing through the queue, each modeled as an MMPP. Then the superposition of all these streams is also an MMPP and hence we consider only one such stream. Let the mean arrival rate of the MMPP stream be $E[\lambda]$. If it is a Poisson stream, we just write λ. The packet lengths of the UDP stream are also assumed iid with a general distribution, and with a generic length s.

It is shown in Sharma and Punyaslok [10] that this system has a unique stationary distribution if $E[\lambda]E[s] < 1$, which we assume in this paper. We denote the average throughput of the ith TCP stream by $\lambda_T(i)$, its stationary mean sojourn time in the queue $E[V_T(i)]$ and the mean queue length $E[q_T(i)]$. The following closed form expressions are obtained in Sharma and Punyaslok [10] : for each i,

$$E[V_T(i)] = \left(\frac{\sum_{j=1}^{N} E[s(j)]E[q_T(j)])}{1 - E[\lambda]E[s]} \right), \qquad (1)$$

$$\lambda_T(i)E[V_T(i)] = E[q_T(i)], \qquad (2)$$

$$\lambda_T(i)\Delta(i) = W_{max}(i) - E[q_T(i)], \qquad (3)$$

$$E[V_T(i)] = E[D(i)] - \Delta(i) \qquad (4)$$

where $E[D(i)]$ is the mean RTT for TCP connection i. Of course, in the above equations (2), (3) are simply the Little's law and (4) is also obvious. Only (1) requires some thought and is in-fact an approximation. Now the mean sojourn time for the UDP stream can also be obtained, using Little's law on the queue for the overall traffic (because the mean queue length for the UDP stream is also available in [10]). But, we will simply use (1) which provides a good approximation.

The formulae in (1) - (4) will be used in the next section to provide various estimates using the active measurements we suggest there.

In [10] these formulae have been obtained for a tandem of queues also (each queue with its own cross traffic). We have used them to obtain measurement based estimates for tandem queues also. These are not reported for lack of space.

2.2 *Measurement based estimation of QoS parameters*

Suppose we want to consider a new TCP request which may have a minimum bandwidth (throughput) requirement and also an upper bound on mean delay. We consider a candidate route for this connection. Let there be a bottleneck router on this route. Most of the delays and throughput constraints will be at this router. The other routers on the way can then be ignored to some extent. The delays caused by them can be included in the propagation delays.

To explore the suitability of the router and the W_{max} needed for the required throughput, we send a TCP probing stream through the route. The probing stream has $W_{max}(p)$ as its maximum window size and $s(p)$ a generic packet length (assumed iid). We assume the knowledge of $W_{max}(p)$ and $E[s(p)]$. We estimate its mean round trip time $E[D(p)]$ from the average of the RTTs obtained by the probing stream and the mean throughput $\lambda(p)$ by the number of probing stream packets sent in time T (suitably large) divided by T. It is shown in Sharma and Punyaslok [10], that if the superposed UDP streams form an MMPP process (a reasonably realistic assumption), then this system (including the probing stream) is ergodic and asymptotically stationary and regenerative. Therefore, the estimates of $E[D(p)]$ and $\lambda(p)$ so obtained are consistent and in fact under quite realistic assumptions also asymptotically normal.

Now consider the equations (1)-(4) for this stream. We rewrite (1) as

$$E[V_T(p)] = (\alpha + E[s(p)]E[(q_T(p))])\beta \tag{5}$$

where $\beta = [1 - E[\lambda]E[s]]^{-1}$ and

$$\alpha = \sum_{j=1}^{N} E[s(j)]E[q_T(j)].$$

From measurements we know $E[D(p)]$ and $\lambda_T(p)$ and by design we can know $E[s(p)]$ and $W_{max}(p)$. Also, $\Delta(p)$ can be estimated accurately from the minimum of all RTTs (in fact this is also a consistent estimator). We do not know anything else about other streams passing through the bottleneck router. Consider (2)-(5) with i replaced by p in (2)-(4). Since we have estimated $\Delta(p)$ and $E[D(p)]$, from (4) we obtain an estimate of $E[V_T(p)]$. Then from (2) we obtain $E[q_T(p)]$. Substituting these in (5) provides an equation in α and β. To obtain one more equation in α and β, we run the above system with another TCP probing stream (with different $W_{max}(p)$ and $s(p)$ parameters) after terminating the first probing stream, through the same route. Now the parameters α and β remain same as for the first probing stream (because these depend upon only the cross traffic on the router). Thus we obtain two equations in α and β to obtain estimates $\hat{\alpha}$ and $\hat{\beta}$. We use $\hat{\alpha}$ and $\hat{\beta}$ below to obtain QoS parameters for the new TCP and/or UDP connection. However $\hat{\alpha}$ and $\hat{\beta}$ themselves also directly provide the congestion information at the router. For example, $\hat{\alpha}$ is the estimate of the total workload at the queue due to all the existing TCP connections and $(\hat{\beta} - 1)/\hat{\beta} = E[\lambda]E[s]$ is the total traffic intensity due to the UDP connections in the cross traffic at the router link.

Now suppose we want to send a new TCP stream through the above route. Let W'_{max} be its maximum window size. The propagation delay Δ', α and β remain same as during the experiment with the probing stream. Therefore, knowing $E[s\,']$ for the new stream, using (2),(3) and (5) we obtain estimates of $E[V'_T]$, $E[q'_T]$ and λ' from equations

$$E[V'_T] = (\hat{\alpha} + E[s']E[q'_T])\hat{\beta}, \tag{6}$$

$$\lambda'_T E[V'_T] = E[q'_T], \tag{7}$$

$$\lambda'_T \Delta' = W'_{max} - E[q'_T]. \tag{8}$$

If the λ'_T so obtained is not sufficient for the new TCP, one can increase W'_{max} appropriately. Furthermore, to see the effect of the new TCP on the mean sojourn time $E[V_T]$ of existing TCP connections, from (1), we realize that it also equals (6).

Next consider the case when we want to send a new UDP stream. Let it be an MMPP stream with mean arrival rate $E[\lambda']$ and packet lengths iid, with a generic length s'. Then its mean sojourn time can be approximated as

$$\hat{\alpha}\left(\hat{\beta} + \frac{E[\lambda']E[s']}{1 - E[\lambda']E[s']}\right). \tag{9}$$

Furthermore, as a result of this stream the mean sojourn time of existing connections also becomes (9).

Finally we consider the case when a Poisson stream with rate $\lambda(p)$ and iid packet length $s(p)$ is used as a probing stream i.e., UDP protocol is used by the probing stream. The mean sojourn time

$$E[V(p)] = \alpha\left(\beta + \frac{\lambda(p)E[s(p)]}{1 - \lambda(p)E[s(p)]}\right)$$

is estimated from the sample average. Thus we obtain an equation in α and β. Again running the experiment with a UDP probing stream with different parameters gives another equation. The rest of the details remain as for the TCP probing stream.

3. System With RED Control And Drop Tail

In this section we extend our approach to the router with RED control (see Floyd et al. [11] for details of the algorithm). In RED a packet arriving at the queue at the time t is dropped with probability $p(\hat{q}(t))$ independently of everything else, where $\hat{q}(t)$ is the average queue length at time t.

The process $\{\hat{q}(t)\}$ moves much more slowly than the other components of the system. In particular, many consecutive arrivals to the queue will find approximately the same probability of loss. In this section we consider the scenario where on arrival, a new connection request finds the queue stationary. Then the $\{\hat{q}(t)\}$ process is approximately constant (has small variance - see Sharma and Punyaslok [10] for analysis). If its mean is $\hat{q}$, then the packets at that time are being dropped with probability $p(\hat{q})$. We assume (quite realistically) that the new probing stream will not disturb the equilibrium state of $\{\hat{q}(t)\}$ significantly.

Under the above setup we send a TCP probing stream to the queue with RED control. Let its parameters be $W_{max}(p)$ and $E[s(p)]$. Now however, because of possible packet drop at the queue, the window size will not stay $W_{max}(p)$. We have found in Sharma and Punyaslok [10] that if we replace in the formulae (2) -(5) $W_{max}(p)$ with $E[W(p)]$, the mean window size under stationarity, the approximations so obtained are quite good. Therefore, in the present setup we use the same approximations. Now, however, we do not know $\hat{q}$ and hence the probability of packet loss at the bottleneck router. But, since we have an estimate of $\lambda_T(p)$, $E[D_T(p)]$ and $\Delta(p)$, from (2)-(4), we obtain estimates of $E[V_T(p)]$, $E[W_T(p)]$ and $E[q_T(p)]$. Substituting in (5) we obtain an equation in α and β. Sending another probing stream with different $W_{max}(p)$, $E[s(p)]$ we obtain another equation in α and β. Solve for $\hat{\alpha}$, $\hat{\beta}$, the estimates for α and β. The arguments for consistency of the estimators remain as in section 2.2.

Once we have estimated $\hat{\alpha}$ and $\hat{\beta}$, the estimates of $E[V'_T]$, λ'_T for a new TCP stream or of $E[V']$ for a new UDP stream can be obtained as in section 2.2. Based on our experience via simulations on the methods in section 2.2, we did not consider a UDP connection for the probing stream. The scheme of this section can be used in other routers where there is packet drop but RED is not employed.

If the router does not employ the RED algorithm, but packets are dropped due to finite buffer size, we can still use the above scheme. We assume that the queue at the router is under stationarity when the probing stream is sent. The packets are dropped (under stationarity) with probability p. Now the above procedure can be used to estimate the different parameters. The errors in approximation are more in this case than in the system with RED. We report our simulation results in next section.

4. Simulation Results

In this section we verify the accuracy of our measurement based schemes. We have used ns simulator version 2.1b5 of UCB/LBNL. For all our simulations the link speed is 10mbps, the UDP stream (existing connection) is Poisson or MMPP with 750 bytes as packet size. The number of existing TCP connections on the bottleneck router is one or three. We have run simulations for TCP Reno and Tahoe. The approximations for the Tahoe version are slightly worse than for Reno. We report the results for Reno only. Each simulation was run for 100sec. The TCP and UDP probing stream parameters are provided in the tables (which were sent twice with different parameters).

Tables 1-7 are for the system with infinite buffers and no RED control (studied in section 2.2). Tables 1-3 compare the results when the TCP or the UDP protocol is used for the probing stream. In the cross traffic we consider 1 TCP ($W_{max} = 100$, packet size = 750 bytes) and 3 TCP ($W_{max} = 100, 90, 80$, packet size = 950,900,700 bytes) cases. The existing UDP connection is a Poisson stream with arrival rate 333 packets/sec. In Table 1 the estimates of parameters α and β are provided. Table 2 compares the results predicted for a new TCP connection based on the estimates $\hat{\alpha}$ and $\hat{\beta}$ obtained in Table 1 with the simulation results. Table 3 provides the results when the new stream uses UDP. We observe that estimates of β are quite accurate while that of α may not always be. However the estimates of QoS parameters $E[V'_T]$, λ'_T, $E[V']$ are quite accurate. From all the three tables we observe that TCP as probing stream provides more accurate results. However from Table 3 we see that if the new connection is also a UDP stream then the UDP probing stream is not overall worse than the TCP probing stream. Thus, rest of the simulations are for the TCP probing stream.

Tables 4-7 consider the case where the *existing* UDP stream is an MMPP. The MMPP process is an ON - OFF type with ON and OFF periods independent of each other, each exponentially distributed with means 2msec and 4msec. During the ON period the traffic arrives as a Poisson process with rate 233 packets/sec. The UDP packet sizes are 750 bytes. Tables 4 and 5 contain results with 1 TCP and 3 TCP with parameters as in the above paragraph. Table 4 provides the estimates for α and β. In Table 5 the theoretical and simulated results are presented when TCP and UDP are new streams respectively.

Tables 1-5 have no delays. In Tables 6 and 7 we provide results when there are 3 TCP connections, the UDP stream is an MMPP (with parameters given in the last paragraph) and the propagation delay Δ is 10msec and 20msec (for each existing TCP connection). We observe that the theory predicts the $E[V'_T]$, λ'_T and $E[V']$ quite well. The errors are more for the 30msec delay.

Tables 8 and 9 provide results when the router employs RED control. The parameters for the RED algorithm are $p_{max} = 0.033$, T_{min} = 20, T_{max} = 50, β = 0.002. There are 3 TCP connections with W_{max} 50, 45 and 50 packets. The TCP packet sizes are 700,750 and 550 byte. The UDP stream is Poisson with packet size 750. The results are provided when the propagation delays for each of the three TCP connections are 0,5 and 10 msec respectively. The theoretical approximations are very accurate for Δ = 0,5 but for Δ = 10 the errors can be quite large. We have also simulated the system with p_{max} = 0.1. The errors in this case are some what more than for p_{max} = 0.033. This is also the case in other studies on TCP where the probability of loss of the order of 0.1 causes quite large errors in the performance prediction.

Tables 10 and 11 report results for the droptail systems. The cross traffic in this system remains as in the last paragraph. The buffer can store 85 packets.

TABLE 1: TCP Vs UDP probing streams: Estimating α , β Packet sizes of TCP probing stream 650,750. W_{max} =40,50. UDP probing stream: packet sizes are 650 & 450. with arrival rates 200 & 215 packets/sec.

Cross Traffic	α			β		
	True	sim.		True	sim.	
		UDP as probing stream	TCP as probing stream		UDP as probing stream	TCP as probing stream
1 TCP 1 UDP	0.03	0.0525	0.0297	1.25	0.712	1.245
3 TCP 1 UDP	0.1536	0.2347	0.1515	1.25	0.6462	1.2623

TABLE 2: TCP Vs UDP probing streams : Estimation for a new TCP stream Packet sizes of new TCP 350 W_{max} = 30packets

Cross Traffic	Sojourn Time			Throughput		
	Sim	From TCP	From UDP	Sim.	From TCP	From UDP
1 TCP 1 UDP	0.04763	0.0475	0.0435	631.7	625.6	690.4
3 TCP 1 UDP	0.2096	0.2095	0.1529	214.77	214.8	296.2

TABLE 3: TCP Vs UDP probing stream: Estimation for a new UDP stream. New packet sizes 550 and rate 200 packets/sec.

Cross Traffic	Sojourn Time		
	Simulated	From TCP	From UDP
1 TCP 1 UDP	0.0425	0.0408	0.0441
3 TCP 1 UDP	0.2735	0.2392	0.2258

TABLE 4: TCP prob. MMPP UDP cross traffic. Probing TCP: packet sizes 950,650bytes. W_{max} =90,65 packets.

Cross Traffic	α		β	
	True	Sim	True	Sim
1 TCP 1 MMPP	0.06	0.0603	1.1111	1.1042
3 TCP 1 MMPP	0.1754	0.17535	1.1111	1.10805

TABLE 5: TCP prob, MMPP UDP cross traffic. New TCP: packet size 400. W_{max} = 40. New MMPP: packet size 450 bytes and 500 packets/sec with ON-OFF periods 1msec and 2msec.

Cross	TCP new stream				UDP new stream	
	Sojourn Time		Throughput		Sojourn Time	
Traffic	True	Sim	true	Sim	True	Sim
1TCP 1 MMPP	0.0808	0.0807	495	495.4	0.06985	0.0692
3 TCP 1 MMPP	0.2120	0.2122	189.5	189	0.2182	0.2247

TABLE 6: TCP prob, MMPP UDP cross traffic, Δ = 10, 20msec. probing TCP: packet sizes 850,750bytes. W_{max} =85,75 packets. Δ = 15,30msec.

Cross traffic	Delay msec	α		β	
		True	Sim	True	Sim
3 TCP 1 MMPP	10	0.1777	0.1693	1.075	1.0906
3 TCP 1 MMPP	20	0.17	0.1554	1.0465	1.1286

TABLE 7: TCP prob, MMPP UDP cross traffic, Δ = 10, 20msec. New TCP: packet size 400. W_{max} = 35 packets Δ = 15, 30msec. new MMPP packet size 450 bytes and rate 500 packets/sec with ON-OFF periods 1msec and 2msec.

Cross traffic	TCP extra stream				UDP extra stream	
	Sojourn Time		Throughput		Sojourn Time	
	True	Sim	True	Sim	True	Sim
3 TCP 1 MMPP	0.2052	0.1959	158.9	166	0.2015	0.1955
3 TCP 1 MMPP	0.1878	0.1871	183	183.9	0.1888	0.1724

TABLE 8: With RED, TCP prob, Poisson UDP cross traffic, probing TCP: packet sizes 450,650bytes. W_{max} =45,65 packets, Δ = 0,5,10msec.

Cross traffic	Delay msec	α		β	
		True	Sim	True	Sim
3 TCP 1 MMPP	0	0.0139	0.014	1.25	1.2325
3 TCP 1 MMPP	5	0.0116	0.0122	1.25	1.2929
3 TCP 1 MMPP	10	0.0127	0.0071	1.25	1.774

TABLE 9: With RED, TCP prob, Poisson UDP cross traffic Δ = 0,5,10msec. New TCP: packet size 300. W_{max} = 25 packets, new UDP: packet size 450 bytes and rate 200 packets/sec.

Cross traffic	TCP extra stream				UDP extra stream	
	Sojourn Time		Throughput		Sojourn Time	
	True	Sim	True	Sim	True	Sim
3 TCP 1 MMPP	0.0202	0.0193	419.5	419.6	0.0217	0.0204
3 TCP 1 MMPP	0.0179	0.0176	371.4	431	0.0197	0.0192
3 TCP 1 MMPP	0.0144	0.0172	288.1	368.5	0.015	0.029

TABLE 10: With finite buffer, TCP prob, Poisson UDP, Δ = 0,5msec. probing TCP: packet sizes 450,650bytes. W_{max} =45,65 packets, Δ = 0,5msec.

Cross traffic	Delay msec	α		β	
		True	Sim	True	Sim
3 TCP 1 UDP	0	0.025	0.0257	1.25	1.195
3 TCP 1 UDP	5	0.0196	0.0245	1.25	1.049

TABLE 11: With finite buffer, TCP prob, Poisson UDP, Δ = 0,5msec. New TCP: packet size 300, W_{max} = 25 packets, new UDP: packet size 450 bytes, rate 200 packets/sec.

Cross traffic	TCP extra stream				UDP extra stream	
	Sojourn Time		Throughput		Sojourn Time	
	True	Sim	True	Sim	True	Sim
3 TCP 1 UDP	0.0354	0.0346	452.6	514.4	0.0328	0.0374
3 TCP 1 UDP	0.0296	0.0299	520	495	0.0276	0.0335

References

[1] S.Black, et al., An architecture for differentiated service, RFC 2475, IETF, Dec. 1998.

[2] R. Braden, D. Black, S. Shenker, Integrated services in the Internet architecture: an overview, RFC 1633, IETF, June 1994.

[3] A Adams et al., The use of end-to-end multicast measurements for characterizing internal network behavior, IEEE Comm. Magazine, May 2000, Vol 38, no. 5, 152-158.

[4] R. Caceres et al., Measurement and analysis of IP network usage and behavior, IEEE Comm. Magazine, May 2000, Vol 38, no. 5, 144-151.

[5] CAIDA measurement tool taxanomy, http://www.caida.org/tools/meastools.html #traceroute.

[6] V. Paxson, J. Mahdavi, A. Adams and M. Mathis, An architecture for large Scale Internet measurement, IEEE Comm. Magazine, Aug 98, Vol. 36. no. 8, 48-54.

[7] V. Sharma and R. Mazumdar, Estimating traffic parameters in queueing systems with local information, Performance Evaluation, Vol. 32, no.3, 1998, 217-230.

[8] V. Sharma, Estimating traffic intensities at different nodes in networks via a probing stream, in Proc. IEEE Globecom'99, 1999.

[9] S. Alouf, P. Nain and D. Towsley, Inferring network characteristics via moment-based estimators, in Proc. IEEE INFOCOM'01.

[10] V. Sharma and Punyaslok, Performance analysis of TCP connections with RED control and exogenous traffic, to be presented in IEEE conf. GLOBECOM, 01.

[11] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Trans. Networking, Vol. 1, 1993, 397-413.