Performance Analysis of Routers with TCP and UDP Connections with Priority and RED Control

Vinod Sharma
Dept. of ECE, Indian Institute of Science
Bangalore, 560012, India
vinod@ece.iisc.ernet.in

Atul Gupta
CDOT , 71/1 Millers Road
Bangalore, 560052, India
atul@cdotb.ernet.in

Abstract: **We analyse the performance of a router with multiple TCP and UDP connections. The UDP stream has preemptive resume or non pre-emptive priority over the TCP connections. The UDP stream modeled as a Markov modulated Poisson process can actually be a superposition of several UDP streams. The TCP streams can have different propagation delays, packet size distributions and maximum window sizes. We obtain the stability and closed form expressions for various performance parameters. For example the formulae for the throughputs of various TCP streams , their mean sojourn times and the mean sojourn time of the UDP stream are provided. Next we obtain all the above results when the router deploys the RED congestion control algorithm. Our approximations are verified through extensive simulation results.**
Keywords: TCP protocol, stability, performance analysis, RED algorithm, priority.

1 Introduction

The TCP/IP based Internet has become the dominant networking technology today. Even though it was originally designed to provide best effort service to data traffic, currently intensive efforts are being made to support real time services also (see Internet RFCs [7], [5]). Despite this effort, the analysis of architectures in the Internet environment is only recently being studied.

The real time services are expected to use the UDP protocol. Therefore, the UDP traffic will increase as the real time traffic increases. In order to provide delay and throughput guarantees to those UDP streams, priority is one of the methods proposed [20]. For example, the Internet draft [15] proposes a Premium service in the Internet, which is going to use the UDP protocol. This traffic will be given priority at the routers. We study the performance of such a system. We consider a router where several TCP and UDP connections are sharing the buffers at the output link of the router. Later on we also consider a system where the router employs active queue management techniques e.g. Random Early Detection (RED) [10]. RED is recommended for implementation in Routers to manage queue lengths, reduce end-to-end latency and reduce packet dropping within the Internet [8]. Therefore, we specifically concentrate on it but our techniques will be applicable to other control algorithms of similar type. For these systems we first provide the conditions under which the system is stable. Then we develop closed form expressions for the throughput and the mean sojourn times of the different TCP and UDP connections. Finally we verify the accuracy of these formulae via simulations.

We cite now some studies on the analysis of TCP connections passing through different routers. The TCP has been analyzed in Altman *et al.* [1], Bonald [6], La *et al.* [11], Padhye *et al.* [16], Lakshman and Madhow [12] and Veres and Boda [19], in addition to others. The TCP with the RED control at the routers has been studied in Firoiu and Borden [9], Misra *et al.* [14] and Sharma and Punyaslok [18]. The extra UDP stream has been considered only in Baccelli and Hong [3], Bonald [6] and Sharma and Punyaslok [18]. As explained above, in the current Internet, one will usually expect several UDP streams also. A realistic model for a superposition of UDP streams is stochastic. Under a stochastic traffic, the periodic behavior seen in the network when only TCP connections are there, will be destroyed. Also, the queue lengths and delays can be significantly affected by the UDP traffic. Furthermore among the above studies, only Baccelli and Hong [3] and Sharma and Punyaslok [18] prove the stability (ergodicity) of the system. The system with priority has not been, to the best of our knowledge, analyzed so far.

The paper is organized as follows. In section 2 we describe our model, provide the notations and briefly recall the TCP window flow control and the RED algorithm. In section 2.1 we mention the conditions under which we have proved the stability of the system. In sections 3 we provide the performance analysis of the system. Sections 3.2 and 3.3 provides the formulae for the throughput and the mean sojourn times of the TCP and the UDP packets when there is no RED control at the router. Section 3.3 extends the analysis to the system with RED control. Section 4 verifies the accuracy of these results via simulations.

2 The Model

We consider a single queue representing the buffer at the output link on a router. A TCP and a UDP connection send packets to this queue (later on we will consider multiple TCP and UDP connections). We will consider TCP Tahoe and TCP Reno versions. We assume that the TCP connection is sending a long file and hence it always has packets to send. The packet lengths of the TCP and UDP packets are assumed iid with general distributions and finite means (the two streams can have different distributions). A generic TCP packet has length s_T and a UDP packet has length s. The UDP stream is generating packets as an MMPP (Markov modulated Poisson process). If instead of one UDP connection, several connections are using the router, each generating an MMPP stream, a superposition of these streams is also an MMPP stream. Hence this generalization is covered by our model. This is a commonly made assumption and is quite realistic especially for a superposition of connections.

We assume the buffer length infinite. Later on the RED algorithm will also be employed. This will effectively make the buffer length finite. The analysis for the infinite buffer queue is useful when the buffer lengths are large. Also, as we will see, the performance indices so obtained, require minor changes to provide the performance of the RED controlled queue.

We assume that the UDP packets are provided time priority over the TCP packets. We consider both the preemptive-resume and the non-preemptive priorities . The actual system should prefer non-preemptive priority because it avoids packet fragmentation. However, if TCP packets can be long then, the delays caused by the residual service time of TCP packets at various nodes on the path of a UDP packet can add up to a significant amount. Thus, preemptive priority has also been considered in literature. The TCP packets experience a total propagation delay (in the forward pipe + the backward pipe by the ACKs) of Δ. We assume Δ to be deterministic, even though a random iid Δ can also be handled.

We also study the above system when the RED algorithm is deployed at the router. This has become a popular active queue management policy and has been implemented in various recent routers.

The following notation will be used. At time t, $w(t)$ is the TCP window size, $h(t)$ the TCP window threshold, $q(t)$ the queue length (including the packet being transmitted), $v(t)$ the total workload in the queue, $q_T(t)$ the number of TCP packets and $q_U(t)$ the number of UDP packets in the queue. At times we will be concerned about these parameters at certain embedded times e.g., at packet arrival epochs. Then we will represent them by attaching a subscript n *e.g.*, the queue length will be denoted by q_n.

Now we briefly describe the window control mechanisms of TCP Tahoe and TCP Reno. For details one may refer to Stevens [17].

For TCP Tahoe when new data is acknowledged (new ACKs are received) the sender recomputes its window size as:

$$\begin{aligned} w_{k+1} &= w_k + 1 \text{ if } w_k < h_k, \text{ (slow start phase)} \\ w_{k+1} &= w_k + 1/w_k \text{ if } w_k \geq h_k, \text{ (congestion avoidance).} \end{aligned}$$

When a packet loss is detected (either by the reception of 3 duplicate ACKs or by a timeout) the sender immediately retransmits the lost packet and sets

$$w_{k+1} = 1, \quad h_{k+1} = max(w_k/2, 2).$$

TCP Reno is an improved version of TCP Tahoe and follows the window adaptation algorithm described below. When a new ACK arrives, the window update is done as in Tahoe. When a packet loss occurs Reno updates its window size depending upon whether it receives three duplicate ACKs or a timeout occurs. When 3 duplicate ACKs are received Reno sets in motion the fast retransmit and fast recovery mechanisms:

(1) When the third duplicate ACK arrives set
$h_{k+1} = max(w_k/2, 2), \quad w_{k+1} = h_{k+1} + 3$
and retransmit the missing packet.

(2) Each time an additional duplicate ACK arrives (requesting the same packet)
$w_{k+1} = w_k + 1$
and transmit a segment if allowed by the window.

If three duplicate ACKs do not arrive at the source by the time the timer times-out,
$w_{k+1} = 1, \quad h_{k+1} = 2.$

The RED algorithm is described in [10]. We provide only the essential details which are usually considered in the analysis of this algorithm (in particular we ignore the counter by fixing it at 0 and the adjustment made during the idle period). On the arrival of the nth packet to the queue, it is dropped with probability p_n while enqueued with probability $1 - p_n$. The p_n is calculated based on the average queue length $\hat{q_n}$ at that time. Two thresholds $0 \leq T_{min} \leq T_{max} \leq \infty$ are fixed appropriately. Then

$$\begin{aligned} p_n &= 0 \text{ , if } \hat{q_n} < T_{min}, \\ &= 1 \text{ , if } \hat{q_n} > T_{max}, \\ &= \frac{p_{max}(\hat{q_n} - T_{min})}{T_{max} - T_{min}} \text{ , if } T_{min} \leq \hat{q_n} \leq T_{max}. \end{aligned}$$

where p_{max} is a parameter recommended to be 0.1 recently. Also, $\hat{q_n}$ is specified by

$$\hat{q_n} = (1 - \beta)\hat{q}_{n-1} + \beta q_n$$

where β is a small parameter, currently specified around 10^{-3}.

2.1 Stability

In this section we mention the stability results we have obtained for this system. Proofs are provided in the detailed version.

First we assume infinite buffer, no RED control and $\Delta = 0$. Consider the preemptive resume priority discipline with Poisson arrivals for the UDP stream . The window size quickly reaches W_{max}, the maximum window size allowed and stays there, if we neglect timeouts because of excessive delays in the queue. This is a good assumption if the traffic intensity $\rho = \lambda E[s]$ of the UDP stream is not very large (say less than 0.5). This is the present likely scenario in the Internet, because the TCP traffic will continue to dominate for some time to come. Thus this becomes a system with one Poisson arrival stream with iid packet lengths and one window controlled stream with fixed length window size W_{max} and iid packet lengths. Because of $\Delta = 0$, all the W_{max} TCP packets will always be present in the queue. A TCP arrival epoch (this is also the epoch when service of a TCP packet is completed) seeing no UDP packet in the queue will be a regeneration epoch for this system. Let τ be a generic regeneration length for the process $X \triangleq \{(w(t), h(t), v(t), q_T(t), q_U(t))\}$. We have shown that when $\rho < 1$, $E[s^\alpha] < \infty$ and $E[s_T^\alpha] < \infty$ for some $\alpha \geq 1$, then $E[\tau^\alpha] < \infty$ and its distribution has a spread-out component. Thus, from regenerative process theory (Asmussen [2], Chapter 5), the process X has a unique stationary distribution and starting from any initial distribution the process converges to it in total variation. A similar statement can be made if we consider the above process at TCP arrival epochs.

Now consider the case when the UDP arrival stream is an MMPP process with the modulating Markov chain a finite state irreducible process. Even for this system we have proved that $E[\tau^\alpha] < \infty$ and τ has a spread-out component under the above mentioned assumptions.

If the priority for the UDP stream is non-preemptive, then we have shown that all the above results hold (for Poisson and MMPP arrivals) under the same assumptions.

Finally we have extended the stability results to the case when $\Delta > 0$ and/or when RED control is deployed at the router with Poisson/MMPP exogenous traffic (for the RED controlled system $\rho < 1$ is not required).

3 Performance Analysis

In this section we provide approximate performance analysis of the system when the UDP stream has preemptive or non-preemptive priority over the TCP stream. In section 3.1 we consider the system with one TCP connection and without the RED control. In section 3.2 we extend the results to the multiple TCP case. In section 3.3 we provide the analysis for the system with RED control. In each case we obtain closed-form expressions for the throughput λ_T of the TCP connections and the mean sojourn times $E[S_T]$ and ES of the TCP and the UDP packets under stationarity. In section 4 we will verify the accuracy of these approximations via simulations.

3.1 Queue with one TCP connection

First we consider the case when the UDP stream is Poisson, and it has preemptive-resume priority over the TCP packets. Initially we also assume that $\Delta = 0$ and the buffer length is infinite. There is no RED control in this section.

Under the above assumptions, the UDP stream is not affected by the TCP packets at all. Thus, it is a simple M/GI/1 queue with mean sojourn time

$$E[S] = \frac{\lambda E[s^2]}{2(1-\rho)} + E[s].$$

Next consider the TCP stream. As mentioned earlier, for infinite buffer case, we can consider it as a fixed window controlled system with window size W_{max}. Also, $\Delta = 0$ implies that W_{max} TCP packets are always there in the queue and the server is never idle. Therefore the TCP packets completely utilize the bandwidth $1-\rho$ leftover by the UDP (the bandwidth normalized to 1). Thus the throughput λ_T of the TCP, in number of packets transmitted in a unit time is $(1-\rho)/E[s_T]$. Also, by Little's law, the mean sojourn time of the TCP packets is

$$E[S_T] = \frac{W_{max}}{\lambda_T} = \frac{W_{max}.E[s_T]}{1-\rho}.$$

Let us compare the above performance indices with a system without priority (see Sharma and Punyaslok [18]). One observes that λ_T and $E[S_T]$ for the TCP remain unchanged while the mean sojourn time $E[S]$ for the UDP stream in the non priority case is $W_{max}E[s_T]/(1-\rho)$. The difference of $E[S]$ can be substantial (see for example Section 4). Thus, we obtain a significant observation that by giving priority to the UDP stream we can gain significantly in the performance of the UDP without loosing anything for the TCP stream! However, the variance of S_T is expected to increase but then the variance can be important for the UDP but not for the TCP connections. These conclusions and observations will be valid in section 3.2 also. However in section 3.3, when we employ the RED control these conclusions may not hold.

Next we consider the case when $\Delta > 0$. In the present scenario, one can collect all the propagation delays in the system and keep at one place (say in the feedback pipe). Now some of the TCP packets and/or Acks may be propagating in the pipe. We assume that the receiver advertised window be large enough so that the server is never idle. Thus λ_T remains $(1-\rho)/E[s_T]$. Let $E[q_T]$ be the mean number of TCP packets in the queue under stationarity. Then, using Little's law on the propagation pipe, we obtain

$$\lambda_T \Delta = W_{max} - E[q_T].$$

and using Little's law in the queue, we obtain

$$\lambda_T E[S_T] = E[q_T].$$

From these equations we obtain $E[S_T]$ and $E[q_T]$. The $E[S]$ is not affected by Δ. Again, the TCP performance indices are not affected by the priority to the UDP.

If the UDP stream is an MMPP, then to obtain the performance of the TCP, we only need to replace λ by the mean rate of arrival for the MMPP stream. To compute $E[S]$, instead of using the result for the $M/GI/1$ queue, one should use the results for the corresponding $MMPP/GI/1$ queue.

Next we analyse the non-preemptive priority system. The performance indices for the TCP stream remain same as for the preemptive queue for all the cases. The performance of the UDP packets is affected to the extend that now, a UDP packet arriving to the queue serving a TCP packet will have to wait till that packet is fully transmitted. A little thought will convince that if the TCP fully utilizes the bandwidth, then the system seen by the UDP stream is a queue with vacation, where the vacations are caused by the service of TCP packets. Since the TCP packets are assumed to have iid distributions, from Bertsekas and Gallager [4],

$$E[S] = \frac{\lambda E[s^2]}{2(1-\rho)} + \frac{E[s_T^2]}{2E[s_T]} + E[s]$$

if the UDP arrival process is Poisson. Similarly one can handle the MMPP case.

3.2 Queue with multiple TCP connections

In this section we study the system of section 3.1 except that now, there are $m \geq 1$ TCP connection sharing the queue. The TCP parameters for connection i are denoted by $W_{max}(i), \Delta(i), s_T(i)$ etc.

We have obtained the stability results for this system for $\Delta = 0$, infinite buffer and without RED control under the additional assumption that $E[(s_T(i))^\alpha] < \infty$ for some $\alpha \geq 1$. The stability with the RED control is currently under investigation.

For the preemptive priority case, the UDP stream is not affected. Thus its sojourn times for the Poisson or the MMPP case remain as before. For the TCP connections, when $\Delta(i) = 0$ for all i, during the time one cycle of $W_{max}(i)$ services is over for connection i, $W_{max}(j)$ services of connection j are over. Therefore,

$$\lambda_T(i) = \frac{(1-\rho)W_{max}(i)}{\sum_{j=1}^{m} W_{max}(j)E[s_T(j)]}.$$

Using Little's law we also get

$$\lambda_T(i)E[S_T(i)] = W_{max}(i),\ i = 1, \ldots, m.$$

When $\Delta(i) > 0$ at least for some i, and are not excessively large for each i (so that the queue is never empty), then by Little's law , for each i,

$$\lambda_T(i)\Delta(i) = Wmax(i) - E[q_T(i)],$$

$$\lambda_T(i)E[S_T(i)] = E[q_T(i)],$$

$$\lambda_T(i) = \frac{(1-\rho)E[q_T(i)]}{\sum_{j=1}^{m} E[q_T(j)]E[s_T(j)]}.$$

Solving these equations simultaneously, we obtain $\lambda_T(i)$, $E[q_T(i)]$ and $E[S_T(i)]$ for each i.

For the non-preemptive priority case, we should replace $E[s_T]$ and $E[s_T^2]$ in the formula for $E[S]$ by $E[s'_T]$ and $E[s'^2_T]$ where

$$E[s'_T] = \frac{\sum_{i=1}^{m} E[q_T(i)]E[s_T(i)]}{\sum_{i=1}^{m} E[q_T(i)]},$$

$$E[s'^2_T] = \frac{\sum_{i=1}^{m} E[q_T(i)]E[s_T^2(i)]}{\sum_{i=1}^{m} E[q_T(i)]}.$$

3.3 Queue with TCP connections and RED control

First consider the case when the packets (of UDP and TCP) arriving at the queue are dropped independently with probability p. This is a special case of the RED queue (as we will explain below). For this problem, we propose to use the formulae given above except that now the λ will be replaced by $\lambda(1-p)$ and the window size will be changing. Therefore, instead of using W_{max} we will use $E[W]$, where EW is the average window length of the TCP connection for packet loss probability p. Various approximations for EW are available (see *e.g.*, [16], [14]) . In this paper we have used the Padhye's [16] approximation, $EW = 1 + \sqrt{(8(1-p)/3p) + 1}$, for one TCP case. In section 4 we have also simulated the system with 5 TCP's. Then this approximation is not good. Thus we have developed a better approximation (to be reported in the detailed version of the paper). Using that approximation we obtain very good match.

Now we consider the system of section 3.2 except that the queue also employs the RED algorithm. Therefore, when the nth packet arrives at the queue it is lost with probability p_n which depends upon the average queue length $\hat{q_n}$. From section 2 we observe that the parameter β used in the dynamics of $\hat{q_n}$ is quite small ($\approx 10^{-3}$). Then, the $\{\hat{q_n}\}$ moves much more slowly than other system components $q_n, w_n, q(t)$ etc as in [18]. We exploit this feature to develop a decomposition approach. In this approach, when $\hat{q_n} \approx \hat{q}$, many packets around the packet n will see the $\{\hat{q_n}\}$ process approximately constant at $\hat{q}$. Thus they will see the queue as a queue without RED but where the packets are lost independently with the probability $p(\hat{q})$ as dictated by the RED algorithm. In addition, the fast moving components of the system can reach stationarity conditioned on $\hat{q_n} \equiv \hat{q}$. Thus the $E[S(\hat{q})], E[S_T(\hat{q})], \lambda(\hat{q})$ can be computed from sections 3.1 and 3.2 as explained in the previous paragraph. In the rest of the section we explain how to compute $\hat{q_n}$.

Define $\hat{z}^\beta(t) = \hat{q}_{\lfloor \frac{t}{\beta} \rfloor}$, $\hat{z}^\beta(0) = \hat{q_0}$. Let $\{z(t)\}$ be the solution of the ODE

$$\frac{dz(t)}{dt} = E[q \mid z(t)] - z(t),\ z(0) = \hat{q_0} \qquad (1)$$

where $E[q \mid z(t)]$ is the (quasi)stationary mean of the process $\{q_n\}$ when $\hat{q_n} \equiv z(t)$. We have proved that as $\beta \downarrow 0$,

$$sup_{0 \leq t \leq T} \mid \hat{z}^\beta(t) - z(t) \mid \ \stackrel{p}{\to} 0 \qquad (2)$$

where $\stackrel{p}{\to}$ denotes convergence in probability. Since in the actual system β is quite small, we can approximate $\hat{z}^\beta(t)$ by the solution of the ODE (1). The solution of the ODE can be computed numerically if we know $E[q \mid z(t)]$.

Now we compute $E[q \mid z(t)]$. The probability of packet loss as dictated by RED when $\hat{q} = z(t)$ is $p(z(t))$. Also,

$$E[q \mid z(t)] = E[q_U \mid z(t)] + E[q_T(p(z(t)))],$$

where $E[q_U \mid s(t)]$ is the time stationary queue length of the UDP packets and $E[q_T(p(z(t)))]$ of the packets of all TCP connections. The $E[q_T(p(z(t)))]$ can be computed for each TCP connection as in sections 3.1 and 3.2 and all those are added to obtain $E[q_T(p(z(t)))]$. Also, $E[q_U \mid z(t)]$ can be computed from the corresponding results for the $M/GI/1$, $M/GI/1$ with vacation or $MMPP/GI/1$ *etc.* as the case may be, where now λ is replaced by $\lambda(1 - p(z(t)))$.

In section 4 we will see that the performance obtained via the approximation is quite good.

Finally we compare the performance of this system with the system without priority. As in section 3.1, the $E[S]$ in the present system is less than in the system without priority. However, unlike in section 3.1, we have shown in the detailed version of the paper that for the system with RED, even though the $\lambda_T(i)$ remain same, the $E[S_T(i)]$ increases in the present system. The reason this difference occurs is because now we need to use $E[W(i)]$ instead of $W_{max}(i)$. $E[W(i)]$ may be different in the present system from the system without priority. Indeed we have shown that $\hat{q}_\infty$ for the present system is strickly less than for the nonpriority case and hence $E[W(i)]$ are strictly larger.

4 Simulation results

In this section we verify the accuracy of the analysis through simulations. We have used the ns-simulator, version 2.1b6 of UCB/LBNL. For all our simulations the link speed has been kept constant at $10Mbps$ and the packet sizes for both the TCP and the exogenous streams are 750 bytes (for the one TCP case). The intensity of the exogenous stream is kept at $\rho = 0.3$. The maximum window size, W_{max} is kept at 30 unless specified otherwise. We have only simulated the non-preemptive priority case which anyway is more interesting from the practical point of view.

We first discuss the results for 1 TCP case without RED. Table 2 provides the theoretical and simulated values for $\lambda_T, E[S_T]$ and $E[S]$ when $\Delta = 0$ and $10\ ms$. One observes a excellent match between the theory and the simulations. Table 1 provides these results when the UDP stream does not have priority. As we concluded from our theory in Section 3.1, the λ_T and $E[S_T]$ are not affected by priority to the UDP but the mean sojourn time of the UDP, $E[S]$ is drastically reduced in the priority case. We have also measured the variance of S_T (not shown in the table) and observed that as predicted earlier, the variance of S_T does increase in the priority case, but not significantly. Table 3 provides the results for the priority case where the UDP is an MMPP stream. The MMPP stream considered is an ON-OFF stream where the ON and the OFF periods are independent and each exponentially distributed with mean $0.004s$ each. During the OFF period the UDP does not generate any packet, while in the ON period the packets are generated as a Poisson stream with rate 1000 pac/s. The mean λ of the MMPP stream is kept as in the Poisson case while the packet lengths are also same.

Next we consider the system with 5 TCP connections. The results are provided in Table 4 for the case when the UDP stream is Poisson . The TCP parameters $W_{max}(i), E[s_T(i)]$ and $\Delta(i)$ are provided in the table. The $E[S]$ for the UDP stream remains as in Table 2. One observes that our theoretical results provide very good match to the simulations.

Finally we describe the results for the system with RED. The RED parameters are $T_{min} = 5$, $T_{max} = 15$ and $p_{max} = 0.1$ for the one TCP case. The TCP and the UDP stream parameters are as above. We plot the theoretical and simulated ($\hat{z}^\beta(t)$) ODE in Figs. $1-3$. Figs $1, 2$ provide the results for the Poisson and the MMPP case respectively when $\beta = 10^{-4}$. We also provide the simulated ($\hat{z}^\beta(t)$) for the non-priority case.We have compared the simulated and theoretical results for the $\lambda_T, E[S_T]$ and $E[S]$. The approximation is good (better than the 5 TCP case provided below). These results are not provided for lack of space. Fig 3 provides the same curves when $\beta = 10^{-3}$. The approximation in Fig. 3 is obviously not as good. The performance indices are provided in Table 5. From this we observe that even for $\beta = 10^{-3}$ the approximations are quite good for except for $E[S_T]$. In this table we also provide the results for the system without priority. This confirms our results that unlike for the infinite buffer case, the $E[S_T]$ for the priority case is higher than for the nonpriority case.

Fig. 4 gives the curves for the 5 TCP case. The TCP parameters $W_{max}(i), E[s_T(i)]$ and $\Delta(i)$ are the same as mentioned above for the without RED case. In this we have plotted the theoretical ODE using the Padhye's approximation as well as our own. One sees that there is a substantial improvement by using our approximation. Table 6 provides the comparison of the theoretical and the simulated values for $\lambda_T, E[S_T]$ and $E[S]$ under stationarity and when $\hat{q} < T_{min}$. In calculating the theoretical values we have used our own approximation of $E[W]$. The approximations under the transient conditions (when $\hat{q} < T_{min}$) are not good. However, since the system acquires stationarity quite fast (even more so for $\beta = 10^{-3}$) , the performance under stationarity is more important.

References

[1] E. Altman, K. Avrachenkov and C. Barakat, *A stochastic model of TCP/IP with stationary losses*, TCP Workshop, Sept 2000, INRIA, France.

[2] S. Asmussen, *Applied Probability and Queues*, John Wiley and Sons, 1987.

[3] F. Baccelli and D. Hong, *TCP is Max-plus linear and what it tells on its throughput*, INRIA Tech. Report, Aug 2000.

[4] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987, N.J.

[5] S. Blake, D. Black, M. Carlson, E. Davies, W. Wang and Weiss, *An Architecture for differentiated service*, RFC 2475, IETF, Dec 1998.

[6] T. Bonald, *Comparison of TCP Reno and TCP Vegas: efficiency and fairness*, Performance Evaluation, Vol. $36-37, 1999, 307-322$.

[7] R. Braden, D. Clack and S. Shankar, *Integrated services in the Internet Architecture: an Overview*, Internet RFC 1633, June 1994.

[8] B. Braden *et. al.*, *Recommendations on Queue Management and congestion Avoidance in the Internet*, In-ernet RFC 2309, April 1998.

[9] V. Firoiu and M. Borden, *A Study of active queue management for congestion control*, IEEE conf. INFOCOM, 2000.

[10] S. Floyd and V. Jacobson. *Random Early Detection Gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking, Vol. $1, 1993, 397-413$

[11] R. J. La, V. Ananthram and J. Walrand, *Analysis and comparison of TCP Reno and Vegas*, IEEE Proc. INFOCOM, 1999.

[12] T. Lakshman and U. Madhow, *The performance of TCP/IP for networks with high bandwidth-delay products and random loss*, IEEE/ACM Trans. Networking, June 1997.

[13] V. Misra, W. B. Gong and D. Towsley, *Fluid based analysis of a network of AQM routers supporting TCP flows with applications* , Proc. SIGCOMM $2000, 1714-1726$.

[14] A. Misra, T. Ott and J. Baras, *The window distribution of multiple TCPs with random loss queues*, IEEE Proc. Globecom 99, $1714-1726$.

[15] K. Nichols, V. Jacobson and L. Zhang, *A Two-bit Differentiated Services Architecture for the Internet*, Internet draft, <draft-nichols-diff-svc-arch-00.txt>, May 1998.
[16] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, *Modeling TCP throughput: A simple model and its empirical validation*, Proc. SIGCOMM 98.
[17] W. R. Stevens, *TCP/IP Illustrated, Volume 1, The Protocols*, Addison-Wesley, 1994.
[18] V. Sharma and P. Punyaslok, *Performance Analysis of TCP Congestion with RED control and Exogenous Traffic*, in Proc. IEEE conf., Globecom 2001.
[19] A. Veres and B. Boda, *The chaotic nature of TCP congestion control*, in Proc. IEEE conf. INFOCOM, 2000.
[20] W. Zhao, D. Olshefski, H. Schulzrinne, *Internet Quality of Service: an Overview*, Columbia Technical report, CUCS-003000, Feb. 2000.

TABLE 1: UDP POISSON TRAFFIC WITHOUT PRIORITY, THROUGHPUT , DELAY

Δ (ms)	λ_T (pac/s)		$E[S_T]$ (ms)		$E[S]$ (ms)	
	Siml.	Theo.	Siml.	Theo.	Siml.	Theo.
0	1166	1167	25.67	25.70	26.12	25.68
10	1166	1167	15.68	15.70	16.10	15.69

TABLE 2: UDP POISSON TRAFFIC WITH NON PREEMPTIVE PRIORITY, THROUGHPUT, DELAY

Δ (ms)	λ_T (pac/s)		$E[S_T]$ (ms)		$E[S]$ (ms)	
	Siml.	Theo.	Siml.	Theo.	Siml.	Theo.
0	1165	1167	25.70	25.70	1.02	1.02
10	1165	1167	15.70	15.70	1.02	1.02

TABLE 3: UDP MMPP TRAFFIC WITH AND WITHOUT NON PREEMPTIVE PRIORITY, $\Delta = 10ms$, THROUGHPUT, DELAY

Priority	λ_T (pac/s)		$E[S_T]$ (ms)		$E[S]$ (ms)	
	Siml.	Theo.	Siml.	Theo.	Siml.	Theo.
None	1162	1167	15.76	15.70	17.05	15.69
UDP	1167	1167	15.59	15.70	1.23	1.23

TABLE 4: 5 TCP CASE WITH NON PREEMPTIVE PRIORITY TO UDP, THROUGHPUT AND DELAYS

		TCP1	TCP2	TCP3	TCP4	TCP5
	Δ	10	10	15	20	10
	W_{max}	20	25	30	35	40
	$E[s_T]$	1000	900	800	700	600
λ_T (pac/s)	Theo.	155.8	194.8	225.0	253.0	311.0
	Siml.	155.9	194.8	224.9	252.7	311.5
$E[S_T]$ (ms)	Theo.	118.3	118.3	118.3	118.3	118.3
	Siml.	118.2	118.1	118.2	118.2	118.2

TABLE 5: UDP POISSON TRAFFIC WITH AND WITHOUT NON PREEMPTIVE PRIORITY AND RED CONTROL WITH $\beta = 10^{-3}$

		Δ	λ_T (pac/s)		$E[S_T]$ (ms)		$E[S]$ (ms)	
		(ms)	Siml.	Theo.	Siml.	Theo.	Siml.	Theo.
Prio.	Stat.	0	1173	1187	7.37	7.66	1.02	1.02
		10	1115	1172	6.04	4.72	1.01	1.02
	Tran.	0	1204	1167	20.63	25.70	0.98	1.02
		10	1032	1167	12.95	15.70	0.99	1.02
No Prio.	Stat.	0	1170	1203	5.94	6.10	6.32	6.10
		10	1116	1173	4.67	3.58	4.87	3.58

TABLE 6: 5 TCP CASE WITH NON PREEMPTIVE PRIORITY TO UDP AND RED CONTROL WITH $\beta = 10^{-4}, T_{min} = 20, T_{max} = 60$

			TCP1	TCP2	TCP3	TCP4	TCP5
		Δ	10	10	15	20	10
		W_{max}	20	25	30	35	40
		$E[s_T]$	1000	900	800	700	600
Statio-narity	λ_T (pac/s)	Theo.	233.4	234.4	204.8	182.5	233.4
		Siml.	233.0	230.2	212.6	192.3	230.6
	$E[S_T]$ (ms)	Theo.	25.9	25.9	25.9	25.9	25.9
		Siml.	27.8	27.8	27.1	26.6	26.8
Tran-sient	λ_T (pac/s)	Theo.	155.8	194.8	225.0	253.0	311.0
		Siml.	170.6	220.6	217.8	220.0	293.9
	$E[S_T]$ (ms)	Theo.	118.3	118.3	118.3	118.3	118.3
		Siml.	93.4	96.0	100.2	102.0	105.2

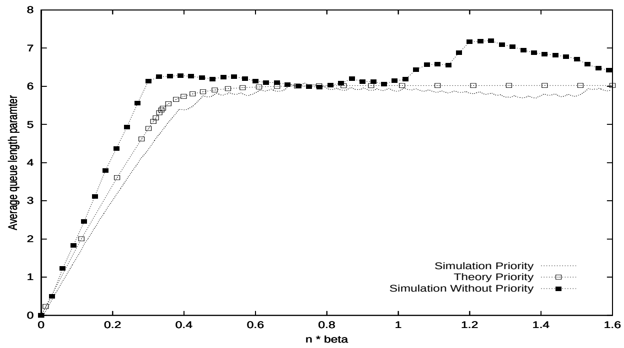

Figure 1: UDP Poisson traffic with $\beta = 10^{-4},\ \Delta = 10\,ms$.

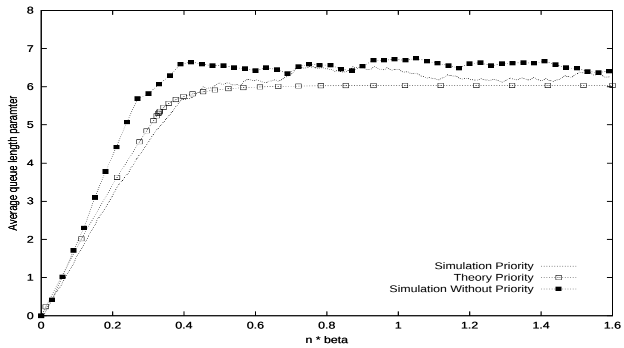

Figure 2: UDP MMPP traffic with $\beta = 10^{-4},\ \Delta = 10\,ms$.

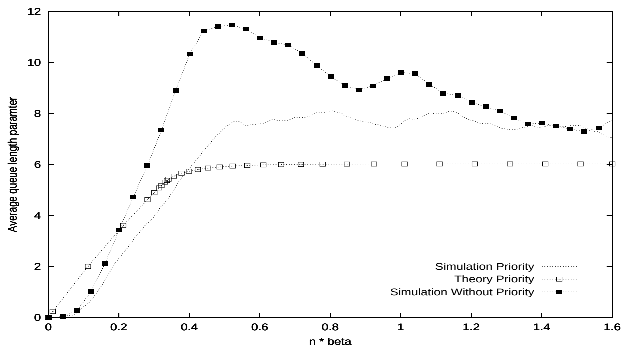

Figure 3: UDP Poisson traffic with $\beta = 10^{-3},\ \Delta = 10\,ms$.

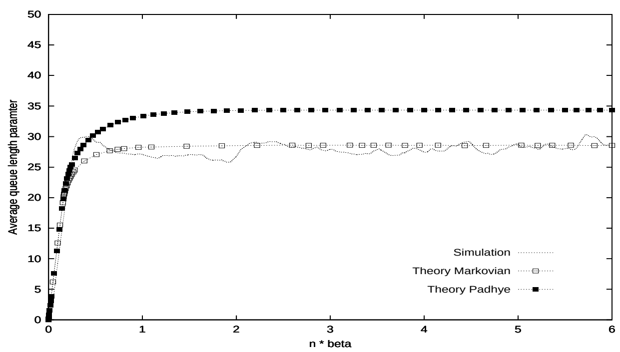

Figure 4: RED Queue with 5 TCP and UDP connection Shows curves for both Padhye as well as our Markovian approximation for EW with $\beta = 10^{-4}, T_{min} = 20, T_{max} = 60$