Considerations for Sizing Buffers in Optical Packet Switched Networks

Arun Vishwanath[†], Vijay Sivaraman[†] and George N. Rouskas[‡]
[†]School of EE&T, University of New South Wales, Sydney, NSW 2052, Australia
Emails: {arunv@ee.unsw.edu.au, vijay@unsw.edu.au}
[‡]Department of Computer Science, North Carolina State University
Raleigh, NC 27695-8206, USA, Email: {rouskas@ncsu.edu}

***Abstract*— Optical packet switches of the foreseeable future are expected to have severely limited buffering capability, since storage of optical signals remains a difficult and expensive operation. Our observations in simulation of TCP and real-time traffic in networks with such small buffers have revealed regions of anomalous performance in which losses for real-time traffic become higher as buffers get larger. The detrimental impact of larger optical buffers is studied in this paper and three new contributions are made. First, we develop a Markov chain model that allows analytical computation of loss. Our model validates observations from simulation, and opens the doors to an analytical understanding of how various factors affect the anomaly. Second, we study the anomaly under realistic traffic mixes containing persistent and non-persistent TCP flows, and show that the traffic mix does not significantly alter the anomaly. Third, we show that larger diversity in packet size between TCP and real-time traffic increases the severity of the anomaly, and is an important consideration when sizing optical switch buffers, particularly since real-time and TCP ACK packets are significantly smaller than the TCP data packets. Our study informs switch manufacturers and network operators of factors to consider when selecting optical buffer sizes in order to achieve desired performance balance between TCP and real-time traffic.**

I. Introduction

In recent years there has been vigorous debate on how large buffers at an Internet router should be. Conventional wisdom, attributed to [1], holds that a router should be able to store a round-trip-time worth of data so as to keep the output link fully utilised while TCP ramps up its window size after a loss event; equivalently, this rule-of-thumb mandates buffer size $B = RTT \times W$ where RTT is the average round-trip time of a TCP flow through the router, and W the capacity of the bottleneck link. This buffer sizing rule was first challenged in 2004 [2] by researchers from Stanford who showed that when a large number M of long-lived TCP flows multiplex at a bottleneck link router, the lack of synchronisation amongst the flows permits a near-100% utilisation of the bottleneck link with only $B = RTT \times W/\sqrt{M}$ buffers. This means that a router carrying $10,000$ TCP flows need only buffer $10,000$ packets as compared to the million required by the original rule-of-thumb.

Since 2004 several new arguments on buffer sizing have been put forth. The Stanford group proposed in [3] that as few as 20 packet buffers suffice at core nodes to realise high (over 80%) link utilisation if TCP flows were to space out the packets they send into the core. This claim was supported by their experimental results at Sprint ATL and Verizon Communications. Researchers from Georgia Tech [4] and the University of Illinois at Urbana-Champaign [5] have focused on per-flow TCP throughput and shown that the ratio of output-to-input link capacity at the router plays a fundamental role in sizing buffers – if the output link has higher capacity than each input, losses fall exponentially with buffer size and small buffers suffice, whereas if the output link has lower capacity than each input, losses follow a power-law reduction and significant buffering is needed. Other studies have considered factors such as application layer performance [6], [7] and fairness [8] influencing buffer sizing.

A. Buffer Sizing in Optical Networks

Most prior studies on buffer sizing have focused on electronic Internet routers, which operate in a fundamentally different buffer size regime to emerging all-optical switches. Electronic chips for packet switching can easily store a few thousand packets in integrated on-chip RAM, unlike optical switches which are expected to be able to store at most a few tens of KiloBytes of data. This is because optical buffering is implemented either via fibre delay lines (FDLs) [9] that circulate the optical data in spools of fibre, which does not scale due to bulk and the need for large optical cross-connects, or via emerging on-chip random-access solid-state optical memory devices [10] that have very limited capacity. The orders of magnitude lower buffering capacity in optical switches necessitates a deeper study of the impact on end-to-end traffic performance in this regime.

While some earlier works such as [3], [11] have applied buffer sizing principles to optical switches, they focus entirely on TCP traffic performance, and completely ignore the performance implications for real-time traffic. Though it is argued that 90-95% of today's Internet traffic is carried over (closed-loop) TCP, there is evidence to indicate that there is a growing demand for (open-loop) real-time traffic, driven by the rising popularity of audio/video, online gaming, VoIP, and IPTV applications. We therefore believe that performance for real-time traffic should also be considered in determining an appropriate buffer size for optical packet switches.

Our previous work in [12] undertook a simulation study of the impact of optical buffer size on performance for mixed TCP and real-time (UDP) traffic. We considered a simple dumbbell-shaped topology that multiplexes 1000 TCP sources (with random round-trip and start times) on a single

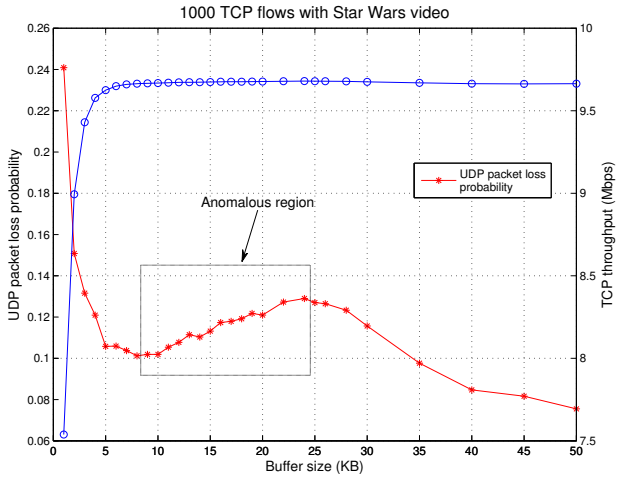


Fig. 1. Star Wars packet loss with 1000 TCP flows

bottleneck link, together with a real UDP traffic trace from the movie *Star Wars* (obtained from [13]). The UDP traffic constituted $\approx 4\%$ of the bottleneck link rate (consistent with the UDP traffic volume in the Internet core). TCP and UDP packet sizes were set at 1000 Bytes and 200 Bytes respectively, in accordance with observations in the Internet that UDP packet sizes are usually smaller as they require low latencies (more details on the simulation setups will be provided in Section III). We profiled the aggregate TCP throughput and real-time packet loss as the buffer size at the optical switch varies from 1 to 50 KiloBytes. Fig. 1 shows that the aggregate TCP throughput increases with buffer size, but the loss for real-time traffic shows surprising behaviour: there is a region of buffer-sizes (9-24 KiloBytes in this case) in which UDP loss increases as buffers get larger. This "anomaly" has serious implications, since it suggests that in this region of buffer sizes, larger buffers marginally benefit TCP throughput but have a **detrimental** impact of UDP loss. Given that each extra KiloByte of optical buffering can add significantly to the cost of the optical switch, manufacturers and operators should be wary of the potential for **negative** returns on this investment.

B. Contributions of This Work

While our prior work in [12] profiled the above mentioned anomalous loss performance under a wide range of traffic settings for UDP (short-range and long-range models) and TCP (round-trip times, number of flows, etc.), it was predominantly simulation based and employed only a crude analytical model based more on intuition rather than rigour. Further, it did not evaluate real-world models for TCP flow durations and packet sizes. In this paper we address these deficiencies and make three new contributions:

1) First, we develop a Markov chain based analytical model that captures the interaction of TCP and UDP traffic at the bottleneck buffers, and allows numerical evaluation of packet loss (albeit under Markovian assumptions). Our model validates the anomalous loss seen in simulations, and provides a handle for analytically evaluating the impact of several system parameters on the severity of the anomaly.
2) Second, in addition to long-lived TCP flows ("elephants"), we consider non-persistent TCP flows ("mice") that arrive and depart, with distributions drawn from long-range dependent models. We show that in the presence of realistic mixes of "elephants" and "mice", anomalous loss still occurs, and the magnitude of the loss increases with the fraction of persistent flows.
3) Third, we study via simulation and analysis the impact of packet sizes on the anomalous loss performance, and show that a larger disparity in packet size leads to a more pronounced anomaly. This is significant since real-time packets (as well as TCP ACK packets) are typically smaller than TCP data packets, and this disparity should be considered when sizing the optical buffer.

We believe our study in this paper adds valuable insight to the ongoing debate on buffer sizing in the context of optical packet switches, and aids switch manufacturers and operators in selection of optical buffer sizes that achieve desired performance balance between TCP and real-time traffic.

The rest of this paper is organised as follows: In Section II, we present an analysis using the M/M/1/B and the M/D/1/B queueing models, which validate the anomaly observed in simulations. In Section III, we study the implications of realistic TCP traffic on the anomaly, i.e., with TCP flows arriving and departing the network. In Section IV, we investigate the impact of varying UDP packet sizes on the anomaly. We conclude the paper in Section V and point to directions for future work.

II. Markov Chain Based Analysis

Knowing how complex it is to mathematically analyse finite buffer systems, and in particular, the interaction between several thousand TCP flows and real-time traffic, we resort to developing a simplified and yet realistic model, which as we shall see, yields valuable analytical insight into why real-time traffic shows anomalous loss behaviour when multiplexed with TCP traffic in the regime of optical buffer sizes.

A. M/M/1/B Analysis

We develop a Markovian model of the optical buffering system, based on an M/M/1/B queue, using the following simplifications:

Assumption: TCP packet arrivals are Poisson. If a large number (thousands) of long-lived TCP flows multiplex at a bottleneck link, it is believed [2] they do not synchronise their window dynamics behaviour, and can thus be treated as independent flows. Combined with the fact that each TCP flow's window will be quite small (since bottleneck buffers are small), implying that each flow will only generate a small amount of traffic per RTT, the aggregation of a large number of such independent flows can reasonably be assumed to yield Poisson traffic. Prior studies on buffer sizing have also employed this assumption [5].

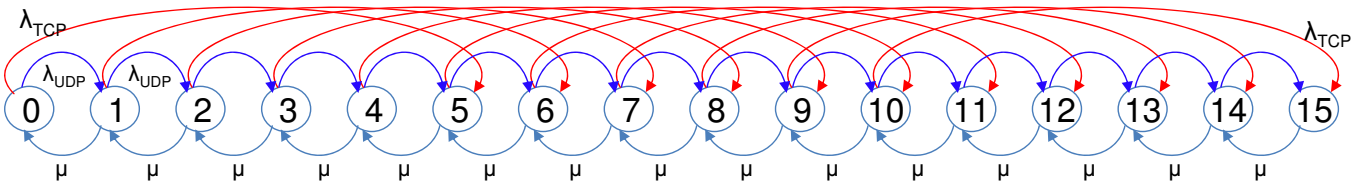


Fig. 2. Markov chain state transition diagram for buffer occupancy with buffer size = 3000 Bytes

Assumption: UDP packet arrivals are also Poisson. Stochastic studies such as [14] have shown that the aggregation of traffic from a large number of flows (as can be expected at an optical core link) converges to Poisson. This important result makes the analysis tractable and helps validation against simulation (our prior work in [12] has shown anomalous losses for variable packet size and long-range dependent UDP traffic).

Claim: UDP packets are on average smaller in size than TCP packets. This has been reported in several measurements of traffic in the Internet core [15], and is attributed to the stricter latency and loss requirements of real-time applications such as video, and on-line gaming applications that use UDP [16]. The study showed that almost all UDP packets were under 200 Bytes. Consistent with our example presented in Fig. 1 above, we choose average TCP and UDP packet sizes to be 1000 and 200 Bytes respectively.

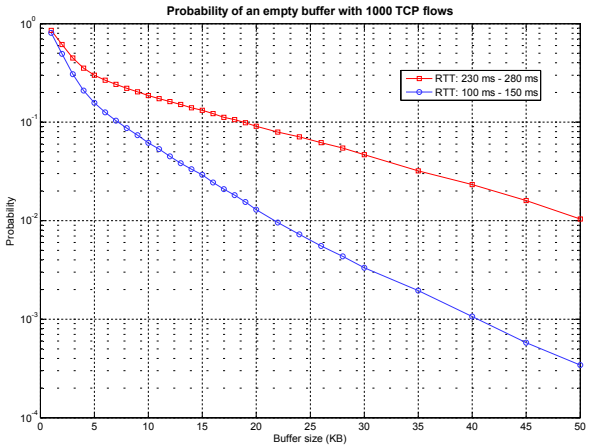


Fig. 3. Probability of empty buffer versus buffer size for TCP traffic

Claim: The aggregate TCP rate increases exponentially with bottleneck link buffer size. If B denotes the bottleneck buffer size (in KiloBytes), then the TCP throughput λ_{TCP} is given by:

$$\lambda_{TCP} \approx \{1 - (e^{-B/B^*})\} * \lambda_{TCP}^{sat} \quad (1)$$

where λ_{TCP}^{sat} denotes the saturation throughput of TCP (when buffer size is large), and B^* is a constant (with same unit as B) that depends on system parameters such as link capacity, round-trip times, etc.

The exponential rise in TCP throughput with buffer size has been reported by previous researchers [17, Sec. III], [3, Fig. 1]. Our observations in simulation corroborate this behaviour: Fig. 3 shows on log scale the idle buffer probability as a function of buffer size when 1000 TCP flows multiplex at the bottleneck link. The linear behaviour in the range 5-50 KB demonstrates that TCP leaves the buffers idle exponentially less often as buffer size grows, implying that its throughput rises exponentially with buffer size. This plot also allows us to estimate B^* (the slope of the log-linear curve being $-1/B^*$) to be 7 KB, which we will use in our analysis below.

With the further assumption that packet sizes are exponentially distributed (we will relax this assumption in the next subsection), we can model the FIFO queue at the bottleneck link router as an M/M/1 system with finite buffer B and with two classes of customers:
1) UDP arrivals are Poisson at fixed rate (denoted by λ_{UDP}), and require exponential service time with unit mean (the service rate is normalised to average UDP packet size), and
2) TCP arrivals are Poisson at rate λ_{TCP} derived from (1), where each TCP packet arrival brings a bulk of 5 customers (corresponding to the packet size ratio 1000/200), each requiring exponential service time with unit average.

For illustrative purposes, let us consider the buffer size B to be 3 KiloBytes. Then, we can model the state of the system as the number of customers in the FIFO queue. Fig. 2 shows the resulting Markov chain. A transition from state j to state $j+5$ corresponds to the arrival of a TCP packet, whereas a transition from state j to state $j+1$ corresponds to the arrival of a UDP packet.

Denoting $B_{bytes} = B * 1000 = 3000$ to be the corresponding buffer size in Bytes, and N the number of states in the Markov chain, then

$$N = \frac{B_{bytes}}{UDP\,packet\,size} + 1 = \frac{3000}{200} + 1 = 16. \quad (2)$$

If p_j represents the steady-state probability of the queue being in state j (i.e., the probability that the queue contains j customers), then we can write the global balance equations as follows:

$$p_0\,(\lambda_{UDP} + \lambda_{TCP}) = p_1\,\mu \quad (3)$$

$$p_i\,(\lambda_{UDP} + \lambda_{TCP} + \mu) = p_{i-1}\,\lambda_{UDP} + p_{i+1}\,\mu \quad (1 \le i \le 4) \quad (4)$$

$$p_i\,(\lambda_{UDP} + \lambda_{TCP} + \mu) = p_{i-1}\,\lambda_{UDP} + p_{i+1}\,\mu + p_{i-5}\,\lambda_{TCP} \quad (5 \le i \le 10) \quad (5)$$

$$p_i\,(\lambda_{UDP} + \mu) = p_{i-1}\,\lambda_{UDP} + p_{i+1}\,\mu + p_{i-5}\,\lambda_{TCP} \quad (11 \le i \le 14) \quad (6)$$

$$p_{15}\,\mu = p_{14}\,\lambda_{UDP} + p_{10}\,\lambda_{TCP} \quad (7)$$

The above equations and the normalising constraint $\sum_{i=0}^{15} p_i = 1$ form a set of linear equations that can be solved

to compute the probability that an incoming UDP packet will be dropped, which in this example is p_{15}. Obtaining balance equations as the buffer size B increases is straightforward, and the resulting set of linear equations is easily solvable numerically (in MATLAB) to obtain the UDP packet loss probability.

The analytical result shown in this paper chooses model parameters to match the simulation setting as closely as possible: the normalised UDP rate is set to $\lambda_{UDP} = 0.05$ (i.e. 5% of link capacity), and the TCP saturation throughput $\lambda_{TCP}^{sat} = 0.94/5$ (so that TCP and UDP customers have a combined maximum rate less than the service rate of $\mu = 1$ in order to guarantee stability). The constant $B^* = 7$ KB, is consistent with what is obtained from Fig. 3.

Fig. 4 plots the UDP loss (on log scale) obtained from solving the M/M/1 chain with bulk arrivals and finite buffers, as well as the TCP rate in Equation 1, as a function of buffer size B. It can be observed that in the region of 1-8 KB of buffering, UDP loss falls monotonically with buffer size. However, in the buffer size region between 9-30 KB, UDP packet loss increases with increasing buffer size, showing that the model is able to predict the anomaly found in simulations. A qualitative explanation of the anomaly is as follows: when the buffers are extremely small (say 1-8 KB), the congestion window of each TCP source remains extremely small as well. This results in each TCP flow transmitting only a few packets during an RTT, thus remaining idle for the most part. As a result, UDP gets exclusive access of the buffers resulting in its packet loss falling monotonically. However, in the range of about 9-30 KB, a larger fraction of TCP flows are able to increase their congestion windows, thereby leaving only a smaller fraction of the buffers for UDP traffic to use. This results in losses for UDP traffic going up in this region.

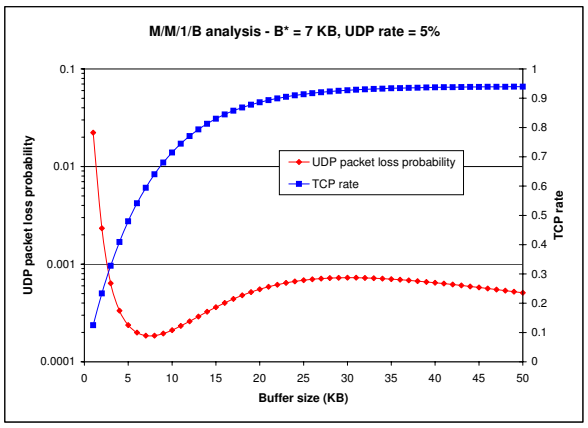


Fig. 4. Anomalous UDP loss results from the M/M/1/B model

B. M/D/1/B Analysis

In this subsection, we refine the M/M/1/B model by relaxing the assumption that packet sizes are exponentially distributed. It has been observed that Internet packet sizes have a tri-modal distribution [18]: TCP data packets are large (1500 Bytes), TCP ACK packets are small (40 Bytes), and real-time and other streams generate intermediate packet sizes (200-500 Bytes). To develop a model that is tractable yet reflective of these dominant modes, we employ an M/D/1/B model in which packet sizes are bi-modal: large for TCP packets (1000 Bytes) and small for UDP packets (say 200 Bytes). Incorporating tri-modal packet sizes makes the computation of transition probabilities significantly more complex without providing additional insight.

The M/D/1/B system is modeled as a semi-Markov process, with the state denoting the number of customers left behind in the system by a departing customer. Transitions occur in this imbedded Markov chain at customer departure instants. For illustration, Fig. 5 shows the resulting imbedded Markov chain for buffer size of 2000 Bytes. Under steady-state conditions, let π_i^* denote the probability that a **departing** customer leaves behind i customers in the system. We are now interested in deriving the stationary probability vector π_i^*, $i = \{0, 1, 2, \ldots, 9\}$, which is necessary to compute the probability that an **arriving** UDP packet is dropped.

Before proceeding with the derivation, we first explain how the chain is constructed. Recall that the state is the number of customers left behind in the system by a departing customer. Again, let λ_{UDP} and λ_{TCP} denote the arrival rate of UDP and TCP packets. As in the M/M/1/B model, a TCP packet arrival brings with it 5 customers (corresponding to TCP/UDP packet size ratio of 5). The service rate is normalised to the average UDP packet size, which is 200 Bytes.

Without loss of generality, let us consider the system to be in state 1, which denotes that there is exactly one 200 Byte customer in the system, and this customer is currently undergoing service. Transitions from state 1 can occur in any of the following ways.

1) The customer finishes service, but while the customer is being served, there are no UDP and TCP arrivals. Thus, when the customer leaves the server, the system transitions into state 0 with probability $p_{1,0}$, which can be computed as follows

$$p_{1,0} = e^{-\lambda_{UDP}} \times e^{-\lambda_{TCP}} \tag{8}$$

2) The customer finishes service, and while the customer is being served, there is exactly one UDP arrival and no TCP arrivals. This is $p_{1,1}$, which is

$$p_{1,1} = e^{-\lambda_{UDP}} \lambda_{UDP} \times e^{-\lambda_{TCP}} \tag{9}$$

3) Continuing, note that we can transition from state 1 to any state $j = \{2, 3, \ldots, 9\}$ with probability $p_{1,j}$. This denotes the arrival of j customers during the service time of a single customer. If j is greater than or equal to 5, then it is important to take into account different TCP and UDP packet arrival combinations while computing the transition probabilities. For example, consider the transition from state 1 to state 9. This denotes the arrival of 9 customers, or equivalently 1800 Bytes, since each customer is equivalent to 200 Bytes. Different packet arrival combinations can add up to 1800 Bytes. One possibility is to have exactly nine UDP packet arrivals and no TCP packet arrivals, while the other could be exactly one TCP packet arrival and four UDP

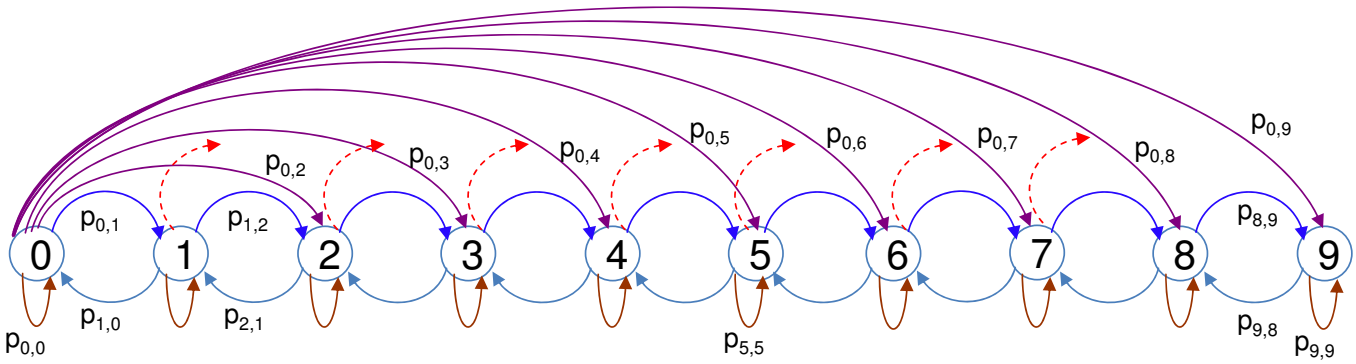


Fig. 5. M/D/1/B state transition diagram for buffer occupancy with buffer size = 2000 Bytes

packet arrivals. Clearly, the number of possible combinations we have to consider increases with buffer size. Finally, it is also important to note that in theory, we can have potentially infinite customers arriving while a customer is being served, although the probability of a large number of customer arrivals is infinitesimally small as the arrival process is Poisson. Since we are dealing with a system with finite buffer capacity, $\{10, 11, 12, \dots\}$ or more customer arrivals will all result in a transition from state 1 to state 9. Thus $p_{1,9}$ is simply $1 - \sum_{j=0}^{8} p_{1,j}$.

In general, if the maximum number of customers that can be left behind in the system by a departing customer is C, and we are transitioning from state i to state j ($i \leq j < C$) due to K customers arriving, then $K = j - i + 1$. The transition probability $p_{i,j}$ can be written as

$$p_{i,j} = \sum_{t=0}^{\lfloor K/5 \rfloor} \sum_{u=0}^{K} \frac{e^{-\lambda_{TCP}}(\lambda_{TCP})^t}{t!} \times \frac{e^{-\lambda_{UDP}}(\lambda_{UDP})^u}{u!} \quad (10)$$

where t and u denote the number of TCP and UDP packets respectively, such that $5t + u = K$ always holds. Further, we can only transition from state i down to state $i - 1$. This occurs only when there are no TCP or UDP arrivals during the service time of a customer, and happens with probability

$$p_{i,i-1} = e^{-\lambda_{UDP}} \times e^{-\lambda_{TCP}} \ , \ i \geq 1 \quad (11)$$

Lastly, transition from state $i \geq 1$ to state C occurs with probability

$$1 - \sum_{j=i-1}^{C-1} p_{i,j} \ , \ i \geq 1 \quad (12)$$

Let us now consider transitions from state 0. Since these transitions are unlike the ones we have discussed so far, as we shall see, they have to be treated differently. A transition from state 0 can proceed in any of the following ways.

Consider the transition from state 0 to itself. This can occur **iff** a UDP packet arrives before a TCP packet, and while the UDP packet is being served, there are no TCP or UDP packet arrivals. Therefore,

$$p_{0,0} = \left(\frac{\lambda_{UDP}}{\lambda_{UDP} + \lambda_{TCP}} \right) \times e^{-\lambda_{TCP}} \times e^{-\lambda_{UDP}} \quad (13)$$

Transitions from 0 to states $\{1, 2, 3\}$ happen in a similar way. A UDP packet must arrive before a TCP packet, and while it is being served, exactly $\{1, 2, 3\}$ UDP packets and no TCP packets should arrive. It is important to note that transitions from state 0 to states $\{1, 2, 3\}$ cannot occur if a TCP packet arrives before a UDP packet. Further, a transition from state 0 to state 4 can occur in one of two ways. If a UDP packet arrives first, then while it is being served, there are exactly four new UDP arrivals and no TCP arrivals. On the other hand, if a TCP packet arrives first, and since each TCP packet brings five customers with it, there are no UDP and TCP arrivals while the first TCP customer is being served.

In general, the transition probability from state 0 to state $K < C$ can be computed as follows. If a UDP packet arrives first, then during its service time we must have exactly K new customers arriving. On the other hand, if a TCP packet arrives first, and since each TCP arrival brings five customers with it, we must have exactly $K - 4$ new customer arrivals during the service time of the first TCP customer. Therefore, $p_{0,K}$ is

$$\frac{\lambda_{UDP}}{\lambda_{UDP} + \lambda_{TCP}} \left[\sum_{t_1=0}^{\lfloor \frac{K}{5} \rfloor} \sum_{u_1=0}^{K} \frac{e^{-\lambda_{TCP}}(\lambda_{TCP})^{t_1}}{(t_1)!} \frac{e^{-\lambda_{UDP}}(\lambda_{UDP})^{u_1}}{(u_1)!} \right] + \frac{\lambda_{TCP}}{\lambda_{UDP} + \lambda_{TCP}} \left[\sum_{t_2=0}^{\lfloor \frac{K-4}{5} \rfloor} \sum_{u_2=0}^{K-4} \frac{e^{-\lambda_{TCP}}(\lambda_{TCP})^{t_2}}{(t_2)!} \frac{e^{-\lambda_{UDP}}(\lambda_{UDP})^{u_2}}{(u_2)!} \right] \quad (14)$$

where t_1, t_2 and u_1, u_2 denote the number of TCP and UDP packet arrivals respectively such that $5t_1 + u_1 = K$ and $5t_2 + u_2 = K - 4$ always holds. Finally, a transition from state 0 to the last state C occurs with probability $1 - \sum_{i=0}^{C-1} p_{0,i}$.

Once the transition probability matrix P is computed, the steady-state probability vector π_i^* can be obtained by solving in MATLAB the vector equation $\boldsymbol{\pi_i^* = \pi_i^* P}$ taking into account the normalising constraint that the sum of the steady-state probabilities of all states in π_i^* is 1. Deducing the exact loss probability (as seen by an arriving customer) from the steady-state probabilities of an imbedded Markov chain is non-trivial even for a simple M/D/1/B birth-death system [19]; it is further complicated in our chain since we are dealing with bulk arrivals and two classes of customers. In what follows, we provide a useful lower bound on the UDP packet loss probability by considering the system at only the customer departure points.

If the buffer size in Bytes is B_{bytes}, then the total number of customers that can be accommodated in the system is $TotalCus = \frac{B_{bytes}}{UDP\,packet\,size} = \frac{B_{bytes}}{200}$. Now, if the system is in state i and given that a UDP packet arrival happens, then a UDP packet loss event occurs if there are at least $(TotalCus - i + 1)$ customer arrivals (in any possible combination of TCP and UDP customers) before the arrival of the UDP packet. Obtaining the above loss event and summing over all i gives us an estimate of the overall UDP packet loss probability. For practical computation we found that, as buffer size increases, larger number of TCP and UDP packet arrival combinations were needed to induce a UDP packet loss event. However, the probability of a large number of arrivals falls exponentially since the arrival process is assumed to be Poisson. Thus, as we vary the buffer size, we consider the UDP packet loss event by taking into account only the rightmost five states in the resulting Markov chain, which gives us an accurate estimate of the overall UDP packet loss probability.

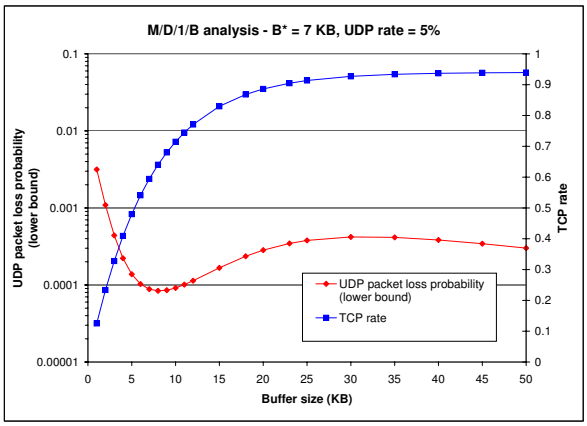


Fig. 6. Anomalous UDP loss (lower bound) results from the M/D/1/B model

Fig. 6 shows the lower bound results from the M/D/1/B model with $B^* = 7$ KB, consistent from Fig. 4. The figures show that the results obtained from the M/D/1/B and the M/M/1/B analysis are qualitatively similar and both models predict the inflection point to occur at 8 KB. Moreover, the M/D/1/B lower bound follows the M/M/1/B loss results very closely, and is also able to validate the anomaly from the realistic scenario of having fixed length packets and not relying on exponential service times.

III. Impact of Short and Long Lived TCP Flows

Our analysis relied on the assumption that TCP's usage of buffers increases exponentially with buffer size. Though this has been observed when all TCP flows are long-lived ([17, Sec. III], [3, Fig. 1], Fig. 3 above), the reader may wonder if similar behaviour is seen when many of the TCP flows are short-lived (or equivalently, the number of TCP flows is time-varying). This is an important consideration since measurement based studies at the core of the Internet suggest that a large number of TCP flows (e.g. HTTP requests) are short-lived ("mice") and carry only a small volume of traffic,

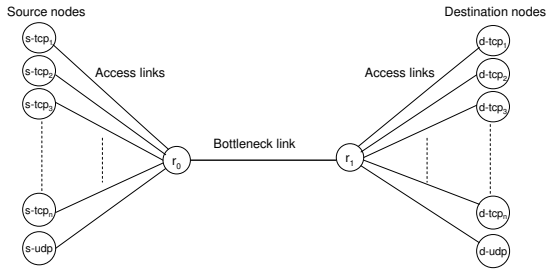


Fig. 7. ns2 dumbbell simulation topology

while a small number of TCP flows (e.g. FTP) are long-lived ("elephants") and carry a large volume of traffic.

We studied UDP loss for such TCP traffic mixes by simulating in $ns2$ a dumbbell topology shown in Fig. 7. In Fig. 8, we plot the TCP empty buffer probability on a 200 Mbps core link for four different ratios of long-lived to short-lived flows. Ideally, we would like to simulate tens of thousands of TCP flows at Gbps (and higher) speeds to emulate core optical links. However, this seems beyond the scope of $ns2$. The total number of TCP flows is kept constant at 2000. In order to incorporate realistic TCP traffic, we consider the closed-loop flow arrival model described in [20] and [4], operating as follows. A given number of users (up to a maximum of 2000 in our example) perform successive file transfers to their respective destination nodes. The size of the file to be transferred follows a Pareto distribution with mean 100 KB and shape parameter 1.5. These chosen values are representative of Internet traffic, and comparable with measurement data. After each file transfer, the user transitions into an idle or off state, or as the authors of [4] suggest, a "thinking period". The duration of the thinking period is exponentially distributed with mean 1 second. It is widely believed that Internet traffic exhibits self-similar and long-range dependent characteristics. It can be noted that the above traffic generation mechanism, which is a combination of several ON-OFF sources with Pareto-distributed ON periods, is in fact long-range dependent [21].

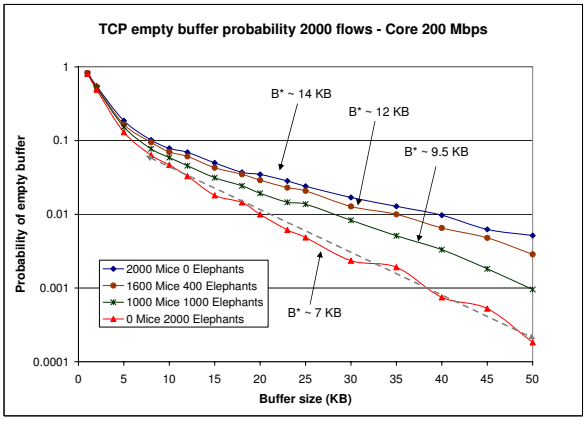
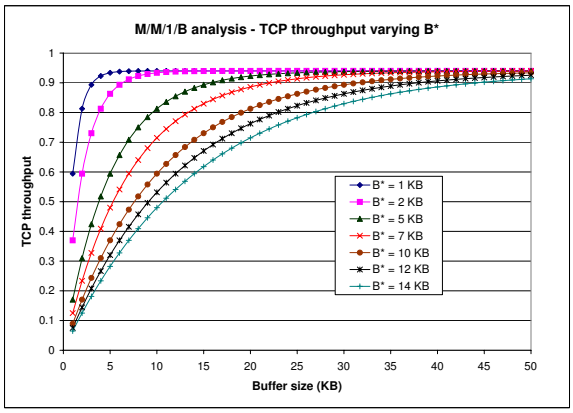
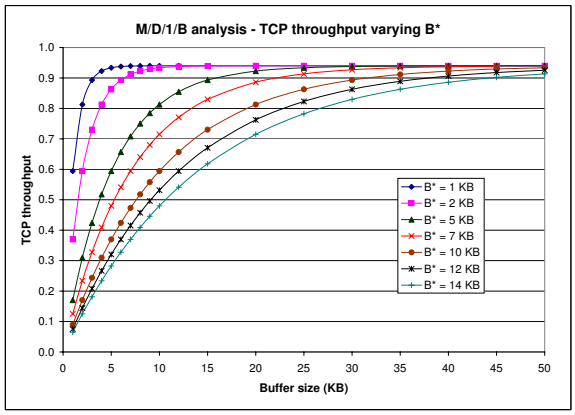


Fig. 8. Empty buffer probability with varying number of mice and elephants

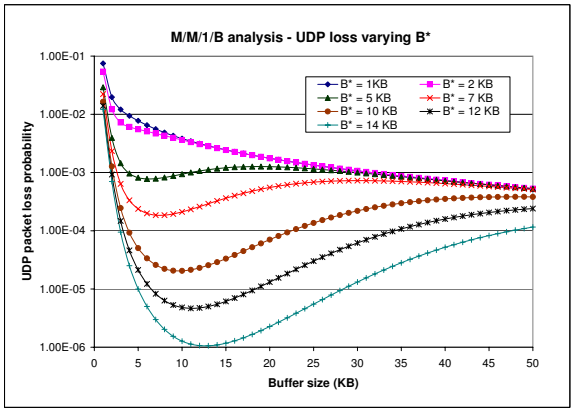
Fig. 8 plots the empty buffer probability for the four different ratios of mice and elephants. Our first observation is that the empty buffer probability falls fairly linearly (on



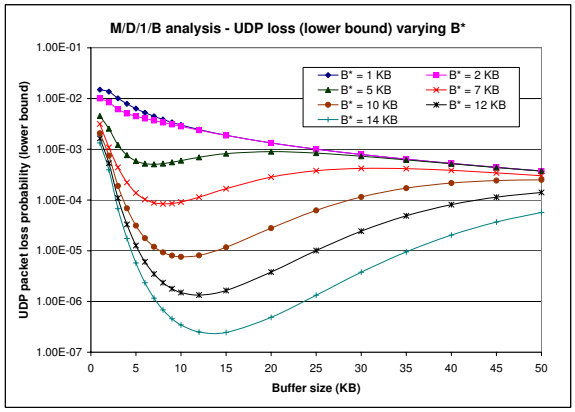
(a) M/M/1/B analysis TCP throughput varying B*



(b) M/D/1/B analysis TCP throughput varying B*



(c) M/M/1/B analysis UDP loss varying B*



(d) M/D/1/B analysis UDP loss (lower bound) varying B*

Fig. 9. UDP loss and TCP throughput from M/M/1/B and M/D/1/B analysis

log-scale) with buffer size (in the region of 8-50 KiloBytes) irrespective of the traffic mix. This satisfies one of the important assumptions required for the analytical model, thereby rendering it valid for any mix of short and long TCP flows. However, the slope of the linear region, which in turn determines the B^* required by the model, does seem to depend on the relative fractions of short and long lived flows. The figure shows that as the fraction of long-lived flows increases from 0 to 1, the value of B^* decreases correspondingly from 14 KB down to 7 KB. Intuitively, this is because short-lived flows do not generate sufficient traffic to continuously saturate the link and most of them remain in the slow-start phase without entering into the congestion avoidance mode during the entire file transfer process. However, with the increase in the number of long-lived flows, there is a corresponding increase in the buffer occupancy, since long-lived flows always have data to send and are for the most part in the congestion avoidance mode. This results in the core link being saturated, and reduces the probability of the buffer being empty, explaining why B^* reduces as the number of long-lived flows increases.

Having observed how B^* changes with the mice-elephant TCP mix, we study the corresponding impact on the performance predicted by our analytical model. Figures 9(a), 9(b) are identical TCP throughput curves obtained from the M/M/1/B and M/D/1/B analysis (λ_{TCP} is the same for the two models) as a function of core link buffer size for different values of B^*, ranging from 1 KB to 14 KB. The key point to note from the figure is that as B^* increases, TCP requires bigger buffers to attain saturation throughput, which is 0.94 or 94% of the core link rate since this analysis plot considers the presence of 0.05 or 5% UDP traffic. In other words, the smaller the B^*, the faster TCP rises, thus needing fewer buffers to attain saturation throughput.

Figures 9(c), 9(d) show the impact of B^* on the UDP loss prediction from our model (both M/M/1/B and M/D/1/B are found to be similar qualitatively). We observe that the UDP packet loss (and hence the severity of the anomaly) is more pronounced when B^* is larger. Moreover, the inflection point, i.e., the point at which UDP packet loss begins to increase, shifts slightly to the right as B^* increases.

To verify if these observations are corroborated in simulation, we multiplex 10 Mbps Poisson traffic with the above four mixes of TCP mice and elephant flows, and record the UDP packet loss as a function of buffer size.

Fig. 10 shows that as the number of long-lived flows increases, there is a corresponding increase in losses for UDP traffic. Referring back to our analytical model, this can be argued as follows: when the fraction of long-lived TCP flows increases from 0 to 1, the core link buffer occupancy due to TCP traffic alone increases, reducing B^* from 14 KiloBytes to 7 KiloBytes. This in turn permits

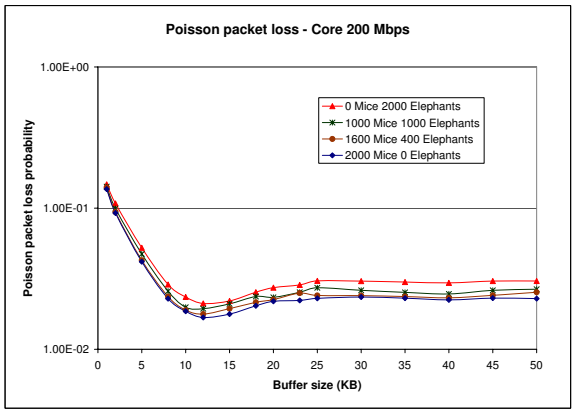


Fig. 10. UDP loss for varying number of mice and elephant TCP flows

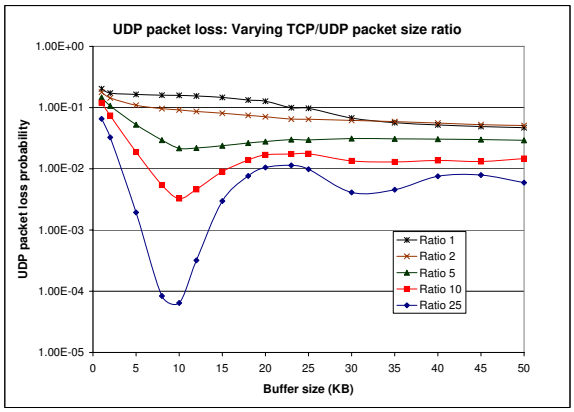


Fig. 11. UDP packet loss from simulation with varying packet size ratios

lesser opportunity for UDP traffic to access the buffers, leading to higher UDP losses. Though the prediction from analysis qualitatively matches simulation, and even predicts the inflection point with reasonable accuracy, we notice a quantitative discrepancy between them, particularly when short TCP flows dominate. We believe this is because our assumption that TCP arrivals are Poisson breaks down when the volume of short-lived TCP traffic is significant, since the mice TCP flows used in simulation have long range dependent characteristics. Nevertheless, our model validates the anomaly even in the presence of mice-elephants TCP mixes, and gives a good indication on how the severity of the anomaly changes with the TCP traffic mix. Given that TCP traffic is by itself notoriously difficult to analyse, let alone thousands of TCP flows interacting with real-time traffic, we believe our model offers sufficient accuracy to be a valuable aid in sizing buffers at optical packet switches.

IV. Impact of varying packet sizes

In this section, we investigate the impact of varying TCP/UDP packet size ratios on the anomalous loss performance, and also point out the implications for TCP ACK (acknowledgement) packet losses.

We consider UDP packet sizes of 40, 100, 200, 500 and 1000 Bytes, while fixing TCP packets at 1000 Bytes. TCP flows, 1000 in number, along with 5% (i.e. 5 Mbps) Poisson UDP traffic, are multiplexed on the dumbbell topology with a 100 Mbps core link. Fig. 11 shows the UDP packet loss observed in simulation as a function of core link buffer size for different packet size ratios.

The figure indicates that UDP losses get progressively smaller as the packet size ratio gets larger (i.e. UDP packets get smaller). This by itself is not surprising, since for a given Poisson rate larger packets act as bulk arrivals that are more bursty, and are moreover dropped in their entirety even if a large part (but not whole) of the packet can be accommodated in the buffer. What is however interesting to note from the plot is that reducing the UDP packet size makes the anomaly more pronounced: when TCP and UDP packets have identical sizes, the anomaly is not seen in simulation, but when UDP packets are only 40 Bytes long, the anomaly is quite severe.

We compare the simulation results against prediction from our M/M/1/B model (the M/D/1/B model is qualitatively similar and not shown here in the interest of space). We use the same $B^* = 7$ KB that was observed in simulation of this scenario (already shown in Fig. 3), and employ in our M/M/1/B chain a bulk arrival size equal to the TCP/UDP packet size ratio. Fig. 12 shows the UDP loss curves obtained via analysis for different packet size ratios. Although the analysis and simulation curves are not a perfect match, it nevertheless corroborates with the results obtained from simulation, and indicates that:

1) First, the analysis predicts correctly that as the TCP/UDP packet size ratio gets larger, losses for UDP traffic become smaller. The bottom curve in Fig. 12 depicts losses for 40 Byte UDP packets (i.e., ratio 25) and clearly shows that loss corresponding to this packet size is significantly lower than loss for larger UDP packets.

2) Second, the analysis seems to be fairly accurate in predicting the inflection point. While the simulations suggest that the inflection point occurs at 10 KB, the analysis predicts it to happen at around 8 KB. Further, that losses (for all the packet size ratios except 25) fall again beyond 30 KB of buffers is predicted successfully by analysis.

3) Third, when TCP and UDP packets are of equal or near-equal size, both simulation and analysis show that the anomaly is insignificant, and it gets more noticeable as the disparity in packet sizes increases.

The importance of packet size to the anomalous loss performance also has an implication for TCP ACK packets that are typically 40 Bytes long. We therefore undertook a study of whether TCP ACK packets will also exhibit similar anomalous behaviour. We simulated 1000 bidirectional TCP flows (without UDP) on the dumbbell topology and recorded the ACK packet drops at routers r_0 and r_1 (see Fig. 7). The simulation parameters are identical to the setup described earlier. In Fig. 13 we plot the ACK packet loss probability as a function of core link buffer size. ACK drops in the forward direction correspond to losses at r_0, while the losses in the reverse direction correspond to losses at r_1. Clearly, ACK packets also suffer from the anomaly, and indeed match well with the analytical estimate plotted in Fig. 12.

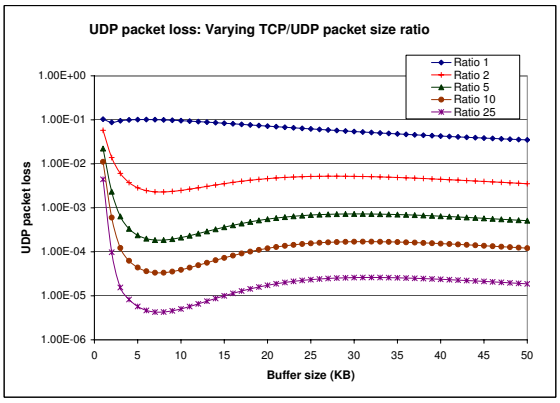


Fig. 12. M/M/1/B UDP packet loss results with varying packet size ratios

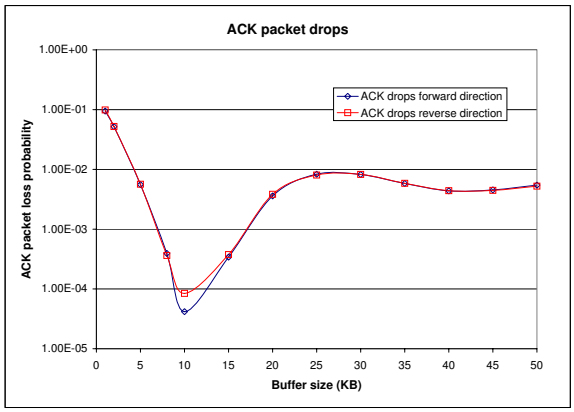


Fig. 13. Losses for TCP ACK packets from simulation

V. Conclusions and Future Work

In this paper, we addressed the joint performance of TCP and UDP traffic at a bottleneck link optical switch equipped with very small buffers and made three new contributions. First, we presented a simplified and yet insightful Markov chain based analytical model to gain valuable insights as to why real-time traffic can show counterintuitive loss behaviour when mixed with TCP traffic. Second, we showed that in the presence of realistic TCP traffic, comprising of short and long-lived flows, UDP traffic can still exhibit anomalous loss performance, and the losses get more pronounced as the fraction of long-lived flows increases. Third, we observed that with the increase in the ratio of TCP to UDP packet size, the anomaly gets more significant, and TCP ACK packets are also equally susceptible to the anomaly.

It is apparent that emerging optical packet switched networks are capable of buffering only a few dozen KiloBytes of data. Given this stringent constraint and the fact that each KB of extra buffering adds significantly to the cost of the optical switch, the anomaly studied here can be of serious concern to switch manufacturers and network providers who make considerable investment in these optical packet switches, only to realise worse performance if they inadvertently operate their buffer sizes in this anomalous region.

Several aspects of the problem require further investigation. In our analysis, we have assumed TCP arrivals to be Markovian. This may not be a very accurate assumption in the presence of short-lived flows, since the traffic is better modeled as long-range dependent. The TCP empty buffer probability on log-scale is fairly linear in the range 8-50 KB. However, in the range 1-7 KB it appears to have a different slope. We can model the empty buffer probability between 1-50 KB range as two-piece linear, and obtain a more accurate expression for the TCP arrival rate. The impact of the ACK drop anomaly on TCP throughput and average flow completion times warrants a deeper understanding. Finally, conducting experiments using NetFPGA boards, similar to the ones reported in [22], will provide more insight into the existence of the anomaly in real test-beds.

References

[1] C. Villamizar and C. Song, "High performance TCP in ANSNet," *ACM SIGCOMM Computer Communications Review*, vol. 24, no. 5, pp. 45-60, 1994.

[2] G. Appenzeller, I. Keslassy and N. McKeown, "Sizing router buffers," *Proc. ACM SIGCOMM*, USA, Aug/Sep. 2004.

[3] M. Enachescu et al., "Routers with very small buffers," *Proc. IEEE INFOCOM*, Spain, Apr. 2006.

[4] R. S. Prasad, C. Dovrolis and M. Thottan, "Router buffer sizing revisited: The role of the output/input capacity ratio," *Proc. ACM CoNEXT*, USA, Dec. 2007.

[5] A. Lakshmikantha, R. Srikant and C. Beck, "Impact of file arrivals and departures on buffer sizing in core routers," *Proc. IEEE INFOCOM*, USA, Apr. 2008.

[6] A. Dhamdhere and C. Dovrolis, "Open issues in router buffer sizing," *ACM SIGCOMM Computer Communications Review*, vol. 36, no. 1, pp. 87-92, Jan. 2006.

[7] G. Vu-Brugier et al., "A critique of recently proposed buffer-sizing strategies," *ACM SIGCOMM Computer Communications Review*, vol. 37, no. 1, pp. 43-47, Jan. 2007.

[8] M. Wang and Y. Ganjali, "The effects of fairness in buffer sizing," *Proc. IFIP NETWORKING*, USA, May 2007.

[9] D. Hunter, M. Chia, and I. Andonovic, "Buffering in optical packet switches," *IEEE/OSA Journal of Lightwave Technology*, vol. 16, no. 12, pp. 2081–2094, Dec. 1998.

[10] H. Park et al., "40-Gb/s optical buffer design and simulations," *Proc. Numerical simulation of optoelectronic devices (NUSOD)*, California, USA, Aug. 2004.

[11] N. Beheshti et al., "Buffer sizing in all-optical packet switches," *OFC/NFOEC*, USA, Mar. 2006.

[12] A. Vishwanath and V. Sivaraman, "Routers with very small buffers: Anomalous loss performance for mixed real-time and TCP traffic," *Proc. IEEE International Workshop on Quality of Service (IWQoS)*, The Netherlands, Jun. 2008.

[13] V. Markovski, F. Xue and L. Trajkovic, "Simulation and analysis of packet loss in video transfers using user datagram protocol," *The Journal of Supercomputing*, vol. 20, no. 2, pp. 175-196, 2001.

[14] J. Cao and K. Ramanan, "A Poisson limit for buffer overflow probabilities," *Proc. IEEE INFOCOM*, USA, Jun. 2002.

[15] Packet traces from measurement and analysis on the WIDE Internet backbone. *http://tracer.csl.sony.co.jp/mawi*

[16] W. Feng et al., "Provisioning on-line games: A traffic analysis of a busy counter-strike server," *Proc. ACM SIGCOMM Internet Measurement Workshop*, France, Nov. 2002.

[17] L. Andrew et al., "Buffer sizing for nonhomogeneous TCP sources," *IEEE Communications Letters*, vol. 9, no. 6, pp. 567-569, Jun. 2005.

[18] K. Thompson, G. J. Miller and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, pp. 10-23, Nov/Dec. 1997.

[19] R. B. Cooper, *Introduction to Queueing Theory*, Elsevier North Holland Publication, 1981.

[20] B. Schroeder, A. Wierman, and M. Harchol-Balter, "Closed versus open: A cautionary tale," *Proc. USENIX NSDI*, USA, May. 2006.

[21] W. Willinger et al., "Self-similarity through high-variability: Statistical analysis of ethernet LAN traffic at the source level," *Proc. ACM SIGCOMM*, USA, Aug/Sep. 1995.

[22] N. Beheshti et al., "Experimental study of router buffer sizing," *Proc. ACM/USENIX Internet Measurement Conference*, Greece, Oct. 2008.