Perspectives on Router Buffer Sizing: Recent Results and Open Problems

Arun Vishwanath
School of EE&T
University of New South Wales
Sydney, NSW 2052, Australia
arunv@ee.unsw.edu.au

Vijay Sivaraman
School of EE&T
University of New South Wales
Sydney, NSW 2052, Australia
vijay@unsw.edu.au

Marina Thottan
Networking Research Lab
Bell Labs Alcatel-Lucent, USA
marinat@alcatel-lucent.com

This article is an editorial note submitted to CCR. It has NOT been peer reviewed. Authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

The past few years have witnessed a lot of debate on how large Internet router buffers should be. The widely believed rule-of-thumb used by router manufacturers today mandates a buffer size equal to the delay-bandwidth product. This rule was first challenged by researchers in 2004 who argued that if there are a large number of long-lived TCP connections flowing through a router, then the buffer size needed is equal to the delay-bandwidth product divided by the square root of the number of long-lived TCP flows. The publication of this result has since reinvigorated interest in the buffer sizing problem with numerous other papers exploring this topic in further detail - ranging from papers questioning the applicability of this result to proposing alternate schemes to developing new congestion control algorithms, etc.

This paper provides a synopsis of the recently proposed buffer sizing strategies and broadly classifies them according to their desired objective: link utilisation, and per-flow performance. We discuss the pros and cons of these different approaches. These prior works study buffer sizing purely in the context of TCP. Subsequently, we present arguments that take into account both real-time and TCP traffic. We also report on the performance studies of various high-speed TCP variants and experimental results for networks with limited buffers. We conclude this paper by outlining some interesting avenues for further research.

Categories and Subject Descriptors

C.2.6 [**Internetworking**]: Routers

General Terms

Design, Performance, Theory, Experimentation

Keywords

Survey, Buffer size, Optical, Mixed real-time and TCP traffic

1. INTRODUCTION

In today's network routers, buffers are used to reduce packet loss by absorbing transient bursts of traffic. They are also instrumental in keeping output links fully utilised during times of congestion. However, the increasing speed of network interfaces raises an important question concerning the size of these buffers. Under buffered routers lead to packet loss, thus adversely affecting application performance, while an over buffered router entails increased latency, complexity and cost.

We begin this survey with an overview of the recommendation from the traditional rule-of-thumb, and then explore in detail various arguments put forth during the past few years.

1.1 Rule-of-thumb (GigaByte Buffers)

The widely used *rule-of-thumb*, commonly attributed to [1], suggests that the amount of buffering needed at a router's output interface is given by $B = C \times RTT$, where B is the buffer size, RTT the average round-trip time of a TCP connection flowing through the router, and C the capacity of the router's network interface. This rule-of-thumb is also called the Bandwidth Delay Product (BDP) rule. The motivation behind this rule was to guarantee 100% link utilisation. In other words, the BDP rule ensures that even when a buffer overflows, and TCP reacts by reducing its transmission rate, there are enough packets stored in the buffer to keep the output link busy, thereby ensuring that the link capacity is not wasted when TCP is increasing its transmission rate. This BDP rule was obtained experimentally using at most 8 TCP flows on a 40 Mbps core link in 1994. No recommendation was made for sizing buffers when there is a significant number of TCP flows that have different RTTs.

In current electronic routers, for a typical RTT of 250 ms and capacity C of 40 Gbps, the rule-of-thumb mandates a buffer size of 1.25 GigaBytes, which poses a considerable challenge to router design. Further, the use of such large buffers (implemented using a combination of SRAM and DRAM chips) complicates router design, increases its power consumption, and makes them very expensive. The scaling and power consumption requirements for next generation routers can be successfully addressed by building and deploying optical routers in the Internet core. However, one of the primary technological limitations of optical routers is the difficulty in building large optical buffers. All-optical buffers can only be used to delay packets for about 100 ns [2]. In the case of a 40 Gbps link, this optical delay line translates to a buffer size of only a few hundred bits. It is thus worthwhile to revisit the buffer sizing problem and examine if we indeed require the amount of buffering as dictated by the BDP rule.

2. SIZING BASED ON LINK UTILISATION

We now outline different buffer sizing models that primarily takes into account link utilisation as the performance metric.

2.1 Near-100% utilisation (MegaByte Buffers)

Researchers from Stanford University showed in 2004 that when a large number N of long-lived TCP flows share a bottleneck link in the core of the Internet, the absence of synchrony among the flows permits a central limit approximation

of the buffer occupancy. The combined effect of multiplexing a large number of asynchronous flows leads to a buffer size $B = RTT \times C/\sqrt{N}$ to achieve near-100% link utilisation [3], [4]. This result assumes that there are a sufficiently large number of TCP flows so that they are asynchronous and independent of each other. In addition, it assumes that the buffer size is largely governed by long-lived TCP flows only. Thus if this result holds, a core router carrying $10,000$ TCP flows needs only 12.5 MB of buffering instead of 1.25 GB as governed by the earlier rule-of-thumb. This model has been referred to as the *small-buffer model* in the literature.

Other recent papers have also reported that near-100% link utilisation can be achieved with significantly fewer buffers than that suggested by the rule-of-thumb. Using a fluid model, [5] formulates the buffer sizing problem as a multi-criteria optimisation problem and suggests that the buffer size needed to get full link utilisation indeed decreases as the number of long-lived TCP flows increases. However, the recommended minimum required buffer size of $(RTT \times C)^2/32N^3$ is lower than the recommendation made by the small-buffer model. [6] shows that in the absence of TCP timeouts and in the presence of a large number of long-lived TCP flows, buffer size much smaller than the BDP is sufficient to get high throughput. As an example, even with 5 users, 99% throughput can be realised with only slightly more than half the buffer size recommended by the BDP. [7] analyses the performance of TCP/AQM systems with packet marking and buffer size in the range $O(N^\alpha)$, where $\alpha \in (0, 1/2)$. A doubly-stochastic Markovian model was developed and was shown to yield good performance as the number of flows increases, in terms of ensuring full link utilisation, with almost zero packet loss and negligible queueing delay.

2.2 80-90% utilisation (KiloByte Buffers)

More recently, using control theory, differential equations, and extensive simulation, [8, 9, 10, 11, 12, 13, 14] have argued in favour of further reducing the buffer size and recommend that as few as 20-50 packets of buffering suffice at core routers for TCP traffic to realise acceptable link capacities. This model has been referred to as the *tiny-buffer model* in the literature. The use of this model however comes with a tradeoff. Reducing buffers to only a few dozen KB can lead to a 10-20% drop in link utilisation. The model relies on the fact that TCP flows are not synchronised and network traffic is not bursty. Such a traffic scenario can happen in two ways. First, since core links operate at much higher speeds than access links, packets from the source node are automatically spread out and bursts are broken. Second, if the TCP stack running at end-hosts is altered such that it can space out packet transmissions (also called TCP pacing) [15]. The slight drop in link utilisation resulting from the tiny-buffer model seems worthwhile since core links are typically overprovisioned, and it pays to sacrifice a bit of link capacity if this permits a move to either an all-optical packet switch or more efficient electronic router design.

While previous results suggest that acceptable link utilisation can be obtained if smooth TCP connections flow through a *single* router with tiny buffers, in [16], the authors explore if an arbitrary network topology can sustain this lowered utilisation. Their results indicate that if a network has a tree structure then no modification is needed in the routers. However, for a general topology, the use of a simple active queue management mechanism called bounded jitter policy is suggested, which will render the arrival traffic at each router to behave as if it is directly being fed by the ingress ports of the network, and thus tiny buffers still suffice to maintain acceptable link utilisation.

Akin to the tiny buffer model, [17] and [18] also propose using very small buffers by devising a strategy that takes into account the needs of other diverse Internet applications. They suggest a size of $2L$ packets, where L is the number of input links. The objective being that the end systems have means to deal with link underutilisation but they have no way of mitigating queueing delays induced by other applications at a drop-tail buffer. With such small buffers, they show that TCP-NewReno can achieve at least 75% link utilisation. To obtain higher utilisation, the recommendation made is to use smoother versions of TCP such as TCP-Vegas. The constant 2 only helps to keep the buffers small enough so that queueing delays do not become significant.

2.3 Critique and alternate approaches

Concerns have been raised regarding the use of *link utilisation* as the only performance metric in determining buffer size. Researchers from Georgia Tech argue that in addition to link utilisation, *packet loss rate* is also an important metric to consider, and that the study of buffer sizing should aim to keep the loss rate bounded to a small value. Consequently, the work in [19] derives the minimum buffer size required to keep the link fully utilised by N long-lived TCP flows with varying round-trip times (i.e., heterogeneous flows), while at the same time attempting to bound the loss rate and queueing delay. Specifically, they show that the buffer size required by N heterogeneous TCP flows at a router employing the drop-tail queue management depends on the harmonic mean of their round-trip times. The resulting buffer size formula is called Buffer Size for Congested Links (BSCL). The authors conclude that Stanford's small-buffer model is more appropriate when provisioning buffers at core routers since it rarely becomes a bottleneck for the majority of the traffic flowing through it. However, the small-buffer model can result in high losses in edge and access routers where links can become congested with large TCP flows that are locally bottlenecked. In this case, BSCL should be preferred.

The importance of packet loss probability in the buffer sizing context was considered in [20] and [21], which showed that the loss rate increases with the square of the number of competing TCP flows. Thus, sizing buffers based just on the rule-of-thumb alone can result in frequent TCP timeouts and significant variations to the per-flow throughput of the various competing flows. To alleviate this problem, [20] also presents an adaptive buffer sizing mechanism called Flow-Proportional Queueing (FPQ), which typically adjusts the amount of buffers according to the number of TCP flows.

Staying with the context of adaptive buffer sizing, [22] models the buffer sizing problem as the Lur'e problem and presents an active drop tail algorithm to determine the buffer size that minimises queueing delay while maintaining a certain average link utilisation. [23] also presents an adaptive buffer sizing algorithm (ABS) where the router adapts its buffer size to suit the dynamics of incoming traffic. Using the monotonic relationship between buffer size, link utilisation, loss rate and queueing delay, ABS aims to maintain system performance above a certain given target objective. Performance results illustrate the applicability of ABS to generic Internet traffic consisting of dynamic HTTP sessions, various existing TCP versions and non-TCP traffic, while being scalable to increasing link capacities.

Additional work highlighting the need to consider loss rate as a performance metric are also reported in [24] and [25]. Us-

ing simulations, the authors of [24] show that employing the small-buffer model can result in 5-15% losses for TCP, which may be unacceptable. Further, the high loss rate can prove perilous to certain audio/video applications that require high reliability and interactivity. They show that the traditional rule-of-thumb can also lead to high packet loss rates in access networks, and that bigger buffers may be needed. [25] reports using experimental data that loss rates can be as high as 25% during times of peak load even when buffer size is high enough to maintain 95% link utilisation. With tiny buffers, their results indicate that loss rates can remain fairly high (about 20%) while link utilisation is consistently low (below 60%), thus deterring the use of tiny buffers on a heavily loaded link.

Some of the concerns raised above have since been partially addressed in [26]. The main conclusions are as follows. First, at the core of the Internet where there are a large number of TCP flows at any given time, buffers can be safely reduced by a factor of ten without affecting the network performance. Second, care should be exercised when directly employing the small-buffer model since it may not hold in all parts of the network, particularly on the access side. Third, the use of tiny buffers is justifiable in a future all-optical network, since bandwidth will be abundant, but technological challenges limit the buffer size to a few dozen packets. Thus, the 10-20% reduction in link utilisation may be acceptable.

[27] argues that the Stanford model (to maintain near-100% link utilisation) is not applicable in small buffer networks since it does not account for the traffic variability on input links. A new method is introduced using a single fixed-point equation to statistically characterise a large network with small buffers. This is derived by combining a series of models that capture the traffic arrival distribution on bottleneck links, instantaneous arrival rates, queue occupancy and packet loss rates. These interrelated models can be used to size buffers given certain target objectives such as maximum packet loss probability or maximum expected packet delay.

3. SIZING BASED ON PER-FLOW METRICS

In this section, we will first review alternate sizing strategies using per-flow TCP metrics and then examine some of the fairness issues associated with packet dropping.

3.1 Average per-flow TCP throughput

Very recently, researchers from Georgia Tech and Bell-Labs have tackled the buffer sizing problem from a completely different perspective [28]. Rather than assuming that most of the TCP traffic is persistent, i.e., long-lived flows that are mostly in the congestion avoidance mode, they consider the more realistic case of non-persistent flows with flow sizes drawn from a heavy-tailed distribution. This differs from some of the early work in that non-persistent flows may not saturate the links along their paths, unlike the persistent flows, and can remain in the slow-start phase without entering into the congestion avoidance mode. Also, the number of active flows at any instant is highly time variant. It follows that flows that spend most of their time in the slow-start phase require significantly fewer buffers than flows that spend most of their time in the congestion avoidance mode.

Further, instead of focusing purely on link utilisation, their work focuses on the average per-flow TCP throughput, which is an important metric as far as an end-user is concerned. It can be the case that a link may have sufficient buffers so that it always maintains high utilisation, but the per-flow throughput can be very low. The objective is to find the buffer size that maximises the average per flow TCP throughput. Analytical, simulation and experimental evidence are presented to suggest that the output/input capacity ratio at a router's interface largely governs the amount of buffering needed at that interface. If this ratio is greater than one, then the loss rate falls exponentially, and only a very small amount of buffering is needed. However, if the output/input capacity ratio is less than one, then the loss rate follows a power-law reduction and significant buffering is needed.

The study concludes by pointing out that it may not be possible to derive a single universal formula to dimension buffers at any router's interface in a network. Instead, a network administrator should decide taking into account several factors such as flow size distribution, nature of TCP traffic, output/input capacity ratios, etc.

3.2 Average flow completion time

A similar treatment is given in [29] by researchers from the University of Illinois at Urbana-Champaign (UIUC). They study buffer sizing requirements in core routers when TCP flows arrive and depart. Using average flow completion time as the metric, the authors show that the core-to-access speed ratio is the key parameter that governs buffer size. The number of flows and buffer size should not be treated independently since depending on the core-to-access speed ratio, buffer size may itself affect the number of flows in the network. Further, if the core-to-access speed ratio is large, then having only a few KiloBytes of buffering is sufficient to realise good performance and this does not reduce link utilisation, as reported in some previous studies.

3.3 Fairness issues

An important factor that warrants consideration when sizing router buffers is the notion of fairness, which reflects the inherent quality of service of the network. If there are a large number of flows sharing a bottleneck link, then a desirable property to have is to ensure that all the flows receive roughly the same amount of bandwidth. [30] presents a simple model to capture the throughput of an individual TCP flow when N long-lived flows are multiplexed at a single link. This turns out to be a function of the link capacity and the queueing and propagation delays of the respective flows. The effects of fairness in buffer sizing is studied in [31] and [32]. Using ns2 simulations and mean-field analysis, the authors investigate the interplay between fairness and desynchronisation of long-lived TCP flows. It is shown that drop-tail queueing can result in unfair packet drops. This combined with TCP-Reno in the presence of a large number of long-lived TCP flows causes the flows to be desynchronised, and thus small buffers suffice. On the other hand, if fairness in packet drops is imposed, then drops can occur over a small time duration using the drop-tail scheme. If these drops are to occur fairly, then a majority of the flows will have to drop packets at roughly the same time. This results in the global synchronisation of TCP flows, which is an undesirable effect, and necessitates larger buffers. If maintaining fairness in packet drops is inevitable and the ill-effects of synchronisation have to be avoided at the same time, then random early detection is a good active queue management that can be exploited.

4. WHAT ABOUT REAL-TIME TRAFFIC?

It must be mentioned that the arguments considered so far deal only with closed-loop TCP traffic, since nearly 90-95% of Internet traffic today is carried by TCP. All previous studies on buffer sizing have largely ignored the impact of open-loop (real-time) traffic, notably UDP. As real-time multimedia applications such as online gaming, audio-video services, IPTV,

VoIP etc. continue to become more prevalent in the Internet, increasing the fraction of Internet traffic that is UDP, it seems appropriate for the study of router buffer sizing to consider the presence of real-time traffic, and not ignore it completely.

The work in [33] and [34], was the first to investigate the impact of optical packet switched (OPS) networks with tiny buffers on the performance of real-time traffic. It is shown that although OPS networks have high capacities, tiny buffers can significantly impact performance when the traffic exhibits short-time-scale burstiness. To alleviate this problem, the authors propose pacing of traffic at the optical edge nodes so that the resulting traffic entering the core nodes is less bursty at short-time-scales. Algorithms of poly-logarithmic complexity in the number of queued packets is proposed to achieve pacing of packets at high data rates. Finally, using analysis and simulations, loss-delay tradeoffs of packet pacing is quantified both for a single bottleneck link and for an OPS network topology. The authors conclude that pacing at the optical edge can be instrumental in realising acceptable performance in emerging OPS networks with tiny buffers.

While [33] and [34] considers the impact of tiny buffers on real-time traffic alone, the recent work in [35] and [36] explores the impact of tiny buffers (up to about 50 KB of buffering) on mixed real-time and TCP traffic at a bottleneck link employing FIFO queues and drop-tail queue management. To understand the dynamics of buffer occupancy at a bottleneck link router, a small fraction of UDP traffic was mixed along with TCP, and measurements were taken to obtain UDP packet loss and end-to-end TCP throughput. Conventional wisdom suggests that bigger switch buffers translate to lower packet loss. Surprisingly, the observation was contrary to conventional wisdom. Using analysis and simulations, the authors show that there exists a certain continuous region of buffer size (typically in the range of about 8-25 packets) wherein the performance of real-time traffic degrades with increasing buffer size. This region is called an "anomalous region" with respect to real-time traffic.

The anomaly has a lot of practical implications. First, it underscores the belief that the study of router buffer sizing should not ignore the presence of real-time traffic. Second, in this regime of tiny buffers, it is prudent to size router buffers at a value that balances the performance of both TCP and UDP traffic appropriately. Operating the router buffers at a very small value can adversely impact the performance of both TCP and UDP traffic. Furthermore, operating it in the "anomalous region" can result in increased UDP packet loss, with only a marginal improvement in end-to-end TCP throughput. Third, building an all-optical packet router and buffering of packets in the optical domain is a rather complex and expensive operation, as envisaged by IRIS; a working prototype of an all-optical packet router from Bell-Labs Alcatel-Lucent [2], [37]. It is mentioned that the optical buffers are the most expensive resource in the IRIS router. In addition, it has been shown in [38] that emerging solid-state optical storage devices can at best buffer a few dozen packets. Thus, the anomaly revealed by the study can be of serious concern to all-optical packet switch designers and network service providers, who make huge investment in setting up the network infrastructure, only to realise potentially degraded performance if appropriate care is not taken when dimensioning their router buffer sizes.

5. CONGESTION CONTROL AND PACING

TCP-Reno/TCP-NewReno has been used in most previous studies on buffer sizing since this is the predominant TCP version employed in various end-hosts. There is ongoing work in understanding the performance of other TCP (and paced) variants, and new congestion control algorithms particularly suited for very small buffers are also being developed.

Exploring the relationship between packet loss synchronisation and buffer size using various high-speed versions of TCP such as BIC, H-TCP, HSTCP and SACK is the objective of [39]. Preliminary conclusions using ns2 simulations is that synchronised packet losses between these different aggressive TCP variants decreases with bottleneck buffer size. Even in the case of high levels of synchronised losses, buffers can be made much smaller than the rule-of-thumb, provided a slight decrease in goodput is acceptable.

In [40], the authors investigate via ns2 simulations if tiny buffers suffice in the presence of high performance scientific applications that require high-speed access links. The results indicate that TCP-Reno flows with a significant proportion of high-speed access links feeding into the core necessitates larger buffers at the core. However, end-host TCP pacing by such applications can alleviate the problem effectively. [41], [42] further highlight the importance of pacing TCP and XCP traffic for improved performance in small buffered networks. [43] studies buffer requirements with HSTCP and per-flow fair queueing, and concludes that a buffer size of 100-200 packets would suffice at any arbitrary link. [44] analyses buffering requirements for RCP and suggests that around 10% of the bandwidth-delay product is sufficient for it to perform well.

[45] undertakes ns2 simulations and reports that Stanford's small-buffer model yields good performance only when the file transfer size is around 50-100 KB or when the propagation delay between the sender and receiver is very small. Also, when paced and non-paced TCP flows coexist, it appears that paced TCP flows suffer from significantly low throughput due to synchronised packet losses. [46] looks at synchronisation and coherence of TCP flows in drop-tail queues using a weakly coupled oscillator model. [47] reports on the implications of reducing buffer size in a large network consisting of both edge and core nodes along with $100,000$ TCP connections. An important result is that as edge networks get faster, reducing buffer size at core routers results in unfairness between those TCP flows that traverse the core and those that do not.

The work in [48] extends the study in [8] to cope with bursty traffic, which in the context of TCP is defined as a point process where each point represents a bulk of TCP data packets. If w represents the average window size, then the size of the bulk is assumed to be uniformly distributed between $2w/3$ and $4w/3$. Using this definition, a simple model is derived to compute the impact of bursty TCP traffic and buffer size on TCP throughput, link utilisation and packet loss. A rather paradoxical conclusion is that by pacing TCP traffic at end-hosts, the TCP traffic can in fact become bursty as it enters the network. Hence, the relationship between burstiness and buffer size warrants a detailed evaluation.

A new congestion control algorithm for networks with tiny buffers is developed in [49]. In such networks, it is shown that when pacing is introduced with TCP-NewReno, link utilisation can tend towards zero with increasing work load and connection bandwidth. To overcome this problem, the authors propose E-TCP (evolutionary TCP), which is an end-to-end transport protocol that effectively utilises the bottleneck link bandwidth in networks equipped with only about 20 packets of buffering. This is achieved by controlling the sending rate so that the packet loss probability at the link is above a certain value p_0. E-TCP operates at equilibrium by using a generalised additive increase multiplicative decrease algorithm.

It is stable and preserves fairness in the presence of multiple flows. Performance studies illustrate the superior nature of E-TCP when compared to TCP-NewReno, and other high performance variants such as HSTCP, STCP and FAST.

6. EXPERIMENTAL STUDIES

The vast majority of buffer sizing literature reported thus far have relied on analysis and simulations to substantiate and validate their claims. In this section, we will draw upon experimental work reported in [50] and [51].

Since the internal architecture of commercial routers and the precise set of queues a packet goes through are not easily available, it becomes difficult to analyse the effect of reducing buffer size. Further, it has also been pointed out that some routers tend to have hidden buffers that are not accessible to the user, and hence buffer size cannot be controlled directly. The study in [28] is an exception. The authors were able to control buffer registers and confirm that there were no hidden buffers. To overcome some of these deficiencies associated with commercial routers, researchers from Stanford University have developed NetFPGA based router linecards [52], which is a PCI board containing programmable FPGA elements and four Gigabit Ethernet interfaces that can be used to perform buffer sizing experiments while controlling the buffers with high precision.

A wide range of experiments were performed in [50] using both the small-buffer and tiny-buffer models with system loads ranging from 25-100%, varying number of users and traffic patterns, round-trip times, access link capacities and congestion window sizes. The authors note that the small-buffer model appears to hold in both the laboratory and operational network environments. It is therefore safe to apply the small-buffer model and reduce buffers at core routers. The tiny-buffer model also appears to hold, and is consistent with theoretical results [12], [13]. However, since this relies extensively on the pacing assumption, it has to be applied with caution since it is possible for certain components such as network interface cards to interfere with the pacing of traffic along the end-to-end path.

The importance of sizing buffers within the context of service level agreements (SLA) is reported in [51]. Contributions of this work are twofold. First, in contrast to most other studies that use idealised router models or network simulators, this work conducts comprehensive laboratory experiments using Cisco GSR and Juniper M320 routers to examine the underlying assumptions used in deriving the $RTT \times C/\sqrt{N}$ (small-buffer) result. The various performance metrics used are throughput, goodput, delay, loss and jitter, both from a traffic aggregate and per-flow point of view. Traffic models range from long-lived TCP flows to self-similar traffic combining both TCP and UDP. Further, drop-tail and RED AQM schemes are also employed.

The main result using the drop-tail scheme is that while aggregate throughput (link utilisation) is largely independent of router architecture, buffer size and offered load, other metrics such as loss and delay are much more sensitive. Their results also indicate that metrics such as throughput, delay, and loss on a *per-flow basis* can show a high degree of dependence on buffer size and offered load. This sheds further light on some of the concerns raised earlier regarding why link utilisation is not a very useful metric when sizing router buffers. RED is better than drop-tail when computing throughput and delay, both from an aggregate and per-flow point of view. However, loss rates are about the same under the two schemes.

Their second contribution is a recommendation to size buffers not purely from a technical standpoint, but also from ISP economics with emphasis on SLAs that drive their networks. Using a set of representative SLAs it is shown that coupling SLA-specific performance with traffic mix can lead to a more informed decision regarding buffer size. In particular, through careful engineering, it may be possible to use smaller buffers even when technical results may suggest otherwise.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have undertaken a comprehensive survey of recent results in the area of router buffer sizing, an important problem in its own right that has gained considerable interest over the past few years. The widely believed rule-of-thumb was challenged by the Stanford group, who also argued that under certain circumstances, as few as 20-50 packets of buffering is sufficient. Objections to the Stanford models have been raised and alternate schemes have been suggested by researchers from Georgia Tech, Bell Labs and UIUC, among others. This survey categorised and outlined the specific contributions made by these different research groups. Our own research effort investigated the performance of mixed UDP and TCP traffic under the tiny-buffer model regime.

So far we have only scratched the surface on the nuances of buffer sizing. We believe there are several research directions in this topic that can have significant impact on router design, and lead to the evolution of novel network architectures.

To the best of our knowledge, there is no experimental work that considers both TCP and UDP traffic. Although our analytical and simulation results suggest the presence of an anomalous region, it will be interesting to observe this in practice. If this is indeed the case, then developing ways to mitigate the anomaly using techniques that may be feasible in an all-optical packet router [53] is a promising direction.

Architectures for emerging all-optical routers such as IRIS are fundamentally different from current electronic routers. As a natural evolution of the core router architecture, the feasibility of tiny buffers makes it possible to consider an increased role for optical technologies in the switching fabric.

Although there is some work regarding fairness and synchronisation amongst TCP flows, understanding their performance implications with tiny buffers will be more insightful.

Finally, the impact of tiny buffers on next generation applications such as online gaming and telepresence can lead to the development of new application architectures and routing schemes.

8. REFERENCES

[1] C. Villamizar and C. Song. High Performance TCP in ANSNet. *ACM CCR*, 24(5):45–60, 1994.

[2] P. Bernasconi et al. Architecture of an Integrated Router Interconnected Spectrally (IRIS). In *IEEE HPSR*, Poland, 2006.

[3] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing Router Buffers. In *ACM SIGCOMM*, USA, 2004.

[4] G. Appenzeller. Sizing Router Buffers. PhD Thesis, Dept. of EE, Stanford University, 2005.

[5] K. Avrachenkov, U. Ayesta, and A. Piunovskiy. Optimal Choice of the Buffer Size in the Internet Routers. In *IEEE Conference on Decision and Control*, Spain, 2005.

[6] L. Andrew et al. Buffer Sizing for Nonhomogeneous TCP Sources. *IEEE Communications Letters*, 9(6):567–569, 2005.

[7] D. Y. Eun and X. Wang. Achieving 100% Throughput in TCP/AQM Under Aggressive Packet Marking With Small Buffer. *IEEE/ACM Transactions on Networking*, 16(4):945–956, 2008.

[8] G. Raina and D. Wischik. Buffer Sizes for Large Multiplexers: TCP Queueing Theory and Instability Analysis. In *EuroNGI*, Italy, 2005.

[9] D. Wischik. Buffer Requirements for High-Speed Routers. In *ECOC*, Scotland, 2005.

[10] D. Wischik and N. McKeown. Part I: Buffer Sizes for Core Routers. *ACM CCR*, 35(2):75–78, 2005.

[11] G. Raina, D. Towsley, and D. Wischik. Part II: Control Theory for Buffer Sizing. *ACM CCR*, 35(2):79–82, 2005.

[12] M. Enachescu et al. Part III: Routers with Very Small Buffers. *ACM CCR*, 35(2):83–89, 2005.

[13] M. Enachescu et al. Routers with Very Small Buffers. In *IEEE INFOCOM*, Spain, 2006.

[14] N. Beheshti et al. Buffer Sizing in All-Optical Packet Switches. In *OFC/NFOEC*, USA, 2006.

[15] A. Aggarwal, S. Savage, and T. Anderson. Understanding the Performance of TCP Pacing. In *IEEE INFOCOM*, Israel, 2000.

[16] N. Beheshti et al. Obtaining High Throughput in Networks with Tiny Buffers. In *IEEE IWQoS*, Netherlands, 2008.

[17] S. Gorinsky, A. Kantawala, and J. Turner. Link Buffer Sizing: A New Look at the Old Problem. In *ISCC*, Spain, 2005.

[18] S. Gorinsky, A. Kantawala, and J. Turner. Simulation Perspectives on Link Buffer Sizing. *Simulation*, 83(3):245–257, 2007.

[19] A. Dhamdhere, H. Jiang, and C. Dovrolis. Buffer Sizing for Congested Internet Links. In *IEEE INFOCOM*, USA, 2005.

[20] R. Morris. TCP Behavior with Many Flows. In *IEEE ICNP*, USA, 1997.

[21] R. Morris. Scalable TCP Congestion Control. In *IEEE INFOCOM*, Israel, 2000.

[22] C. Kellett, R. Shorten, and D. J. Leith. Sizing Internet Router Buffers, Active Queue Management, and the Lur'e Problem. In *IEEE Conference on Decision and Control*, USA, 2006.

[23] Y. Zhang and D. Loguinov. ABS: Adaptive Buffer Sizing for Heterogeneous Networks. In *IEEE IWQoS*, Netherlands, 2008.

[24] A. Dhamdhere and C. Dovrolis. Open Issues in Router Buffer Sizing. *ACM CCR*, 36(1):87–92, 2006.

[25] G. Vu-Brugier et al. A Critique of Recently Proposed Buffer-Sizing Strategies. *ACM CCR*, 37(1):43–47, 2007.

[26] Y. Ganjali and N. McKeown. Update on Buffer Sizing in Internet Routers. *ACM CCR*, 36(5):67–70, 2006.

[27] M. Shifrin and I. Keslassy. Modeling TCP in Small Buffer Networks. In *NETWORKING*, Singapore, 2008.

[28] R. S. Prasad, C. Dovrolis, and M. Thottan. Router Buffer Sizing Revisited: The Role of the Output/Input Capacity Ratio. In *ACM CoNEXT*, USA, 2007.

[29] A. Lakshmikantha, R. Srikant, and C. Beck. Impact of File Arrivals and Departures on Buffer Sizing in Core Routers. In *IEEE INFOCOM*, USA, 2008.

[30] D. Wischik. Fairness, QoS, and Buffer Sizing. *ACM CCR*, 36(1):93, 2006.

[31] M. Wang and Y. Ganjali. The Effects of Fairness in Buffer Sizing. In *NETWORKING*, USA, 2007.

[32] M. Wang. Mean-Field Analysis of Buffer Sizing. In *IEEE GLOBECOM*, USA, 2007.

[33] V. Sivaraman et al. Packet Pacing in Short Buffer Optical Packet Switched Networks. In *IEEE INFOCOM*, Spain, 2006.

[34] V. Sivaraman et al. Packet Pacing in Small Buffer Optical Packet Switched Networks. *IEEE/ACM Transactions on Networking*, 2009 (To appear).

[35] A. Vishwanath and V. Sivaraman. Routers with Very Small Buffers: Anomalous Loss Performance for Mixed Real-Time and TCP Traffic. In *IEEE IWQoS*, Netherlands, 2008.

[36] A. Vishwanath, V. Sivaraman, and G. N. Rouskas. Considerations for Sizing Buffers in Optical Packet Switched Networks. In *IEEE INFOCOM*, Brazil, 2009.

[37] A. S. Wander, A. Varma, and M. Thottan. Traffic Management Framework for Optical Routers with Small Buffers. *Journal of Optical Networking*, 7(11):958–976, 2008.

[38] H. Park et al. 40-Gb/s Optical Buffer Design and Simulations. In *Numerical Simulation of Optoelectronic Devices*, USA, 2004.

[39] S. Hassayoun and D. Ros. Loss Synchronization and Router Buffer Sizing with High-Speed Versions of TCP. In *IEEE INFOCOM HSN Workshop*, USA, 2008.

[40] B. Zhao, A. Vishwanath, and V. Sivaraman. Performance of High-Speed TCP Applications in Networks with Very Small Buffers. In *IEEE ANTS*, India, 2007.

[41] A. Razdan et al. Enhancing TCP Performance in Networks with Small Buffers. In *IEEE ICCCN*, USA, 2002.

[42] O. Alparslan, S. Arakawa, and M. Murata. Performance of Paced and Non-Paced Transmission Control Algorithms in Small Buffered Networks. In *ISCC*, Italy, 2006.

[43] J. Auge and J. Roberts. Buffer Sizing for Elastic Traffic. In *EuroNGI*, Spain, 2006.

[44] A. Lakshmikantha et al. Buffer Sizing Results for RCP Congestion Control Under File Arrivals and Departures. *ACM CCR*, 2009 (To appear).

[45] G. Hasegawa et al. Simulation Studies on Router Buffer Sizing for Short-lived and Pacing TCP Flows. *Elsevier Computer Communications*, 31:3789–3798, 2008.

[46] H. Han et al. Synchronization of TCP Flows in Networks with Small DropTail Buffers. In *IEEE Conference on Decision and Control*, Spain, 2005.

[47] H. Hisamatsu, G. Hasegawa, and M. Murata. Sizing Router Buffers for Large-Scale TCP/IP Networks. In *AINAW*, Canada, 2007.

[48] D. Wischik. Buffer Sizing Theory for Bursty TCP Flows. In *International Zurich on Seminar Communications*, Switzerland, 2006.

[49] Y. Gu et al. Congestion Control for Small Buffer High Speed Networks. In *IEEE INFOCOM*, USA, 2007.

[50] N. Beheshti et al. Experimental Study of Router Buffer Sizing. In *ACM/USENIX IMC*, Greece, 2008.

[51] J. Sommers et al. An SLA Perspective on the Router Buffer Sizing Problem. *ACM SIGMETRICS Performance Evaluation Review*, 35(4):40–51, 2008.

[52] NetFPGA: Programmable Hardware www.netfpga.org.

[53] A. Vishwanath and V. Sivaraman. Shared versus Dedicated Buffers for Real-Time Traffic in Optical Routers with Very Small Buffers. In *IEEE ANTS*, India, 2008.