Update on Buffer Sizing in Internet Routers[*]

Yashar Ganjali, Nick McKeown
Dept. of Electrical Engineering, Stanford University
Stanford, CA 94305, USA
{yganjali, nickm}@stanford.edu

ABSTRACT

In the past two years, several papers have proposed rules that suggest two to five orders of magnitude reduction in Internet core router buffers. Others present scenarios where buffer sizes need to be significantly increased. So why the different rules? In this paper we briefly compare the different results and proposals, and summarize some recent preliminary experiments to validate the proposals. We'll see that different results apply to different parts of the network, and depend on several assumptions. For example, we believe that buffers can be safely reduced by an order of magnitude in the routers in service provider backbone networks; but it would be premature to reduce them in routers closer to the edge.

Categories and Subject Descriptors

C.2 [**Internetworking**]: Routers

General Terms

Design, Performance, Theory

Keywords

TCP, buffer size, congestion control, all-optical routers

1. INTRODUCTION

Most large commercial routers (*e.g.* those at service providers) that are installed today follow the rule of thumb that a router should have a buffer size B approximately equal to $C \times T$, where C is the capacity of the bottleneck link, and T is the effective two-way propagation delay of the flows going through the router. In 2004, we suggested that for large routers with lots of flows, we can use a *small buffers rule* $B = C \times T/\sqrt{N}$, where N is the number of long-lived TCP flows going through the router [3]. In contrast, Dhamdhere and Dovrolis presented a case where even $B = C \times T$ leads to high packet loss rates [8] in access networks, and they suggest that we need to increase buffer sizes. We call this the *drop-based buffers rule.* Recently, Raina and Wischik [14], and independently Enachescu *et al.* [10] suggested the *tiny buffers rule* $B = O(\log W)$, under certain constraints. Here W is the maximum congestion window size for the flows going through the router.

The different rules lead to very different buffer sizes. For example, consider a 40Gb/s linecard on an Internet core router with a round-trip propagation delay of 250ms, carrying about ten thousand TCP flows. If $B = C \times T$ we need about five million packet buffers. With the *drop-based* rule we would need more. With the *small buffers* rule we need about fifty thousand packets, and only 20-50 packets with the *tiny buffers* rule.

So what's going on? Why so many different rules, with buffer sizes ranging over five orders of magnitude? It turns out that the different rules apply in different parts of the network, or under different assumptions. In this paper we try to summarize which rules appear to hold where and when. We'll draw on some recent preliminary experiments on buffer sizing in operational networks and test labs.

Given the potential benefits (and the risk of getting it wrong!) of reducing buffer sizes, it is worth asking if small buffers results hold in real, operational networks. If the results are wrong, then the consequences of reducing the buffers in a router, or in an operational commercial network, could be quite severe. The problem is, how to decide if the result is correct, without trying it in an operational network? But who would reduce buffers in an operational network, and risk losing customers' traffic, before knowing if the result is correct?

2. OVERVIEW OF BUFFER SIZING RULES

We'll start by looking at the assumptions and intuition behind each rule; and try to summarize the evidence from simulations and experiments so far.

2.1 Rule of thumb

2.1.1 Assumptions and Intuition

The rule that $B = T \times C$ assumes there is a single long-lived TCP flow going through the bottleneck link. Essentially, B is determined by the shape of the TCP window-size. Because the window-size follows the well-known sawtooth, with a distance from peak to trough of $T \times C$, then we need this much buffering to ride out reductions in window-size to make sure the bottleneck buffer doesn't go empty and lose throughput.

2.1.2 Validation

It's very easy to show from inspection, simulation or in the lab that with a single long-lived TCP flow we need $B = T \times C$ to maintain 100% utilization [16].

[*] This work was supported under DARPA/MTO DOD-N award no. W911NF-04-0001/KK4118 (LASOR PROJECT) and the Buffer Sizing Grant no. W911NF-05-1-0224.

2.1.3 Discussion

Villamizar and Song's first experiments in 1994 consisted of one to eight flows [16]. With such a small number of flows, the sawtooths tend to synchronize because losses hit each flow at roughly the same time. As a result, the aggregate window-size process is also a sawtooth with the same amplitude, hence the buffer size doesn't change.

2.2 Small Buffers Rule

2.2.1 Assumptions and Intuition

Appenzeller *et al.* proposed reducing buffers by a factor of $\sqrt{N}$ when there are N long-lived TCP flows sharing the link [3]. The claim is that if there are sufficiently large number of flows, they tend to desychronize – it seems to start happening with a hundred flows or so. As the number of flows increases, the amplitude of the aggregate window-size process decreases (and hence the traffic smooths) according to central limit theorem. In the absence of another need for buffers, we can steadily reduce the buffer size as we increase N. (Eventually, though, other effects start to dominate, placing a lower bound on the buffer size, as we will see later.)

Given that a typical, congested 2.5Gb/s or 10Gb/s link will carry tens of thousands of flows, it suggests buffers can be reduced by two orders of magnitude.

2.2.2 Validation

We've now been part of several experiments to test the *small buffers* rule on real networks – either laboratory networks with commercial routers, or operational backbone networks. We summarized the results we know about in [11]. Different experiments were performed with different traffic patterns, network topologies, router architectures, and with different types of measurement infra-structure. To a limited extent, the results represent a variety of scenarios. In each experiment, the buffer size is reduced to several different values, to determine when the link utilization starts to fall below 100%. So far, results are promising and every experiment found that so long as the buffers are larger than $T \times C/\sqrt{N}$ then no utilization is lost; and somewhere close to this value, utilization starts to fall.

As an example, consider an experiment on an operational commercial backbone network in the US. The 2.5Gb/s link under study was highly congested: It ran at 90% utilization for several hours each day. (The operator was just about to upgrade the link to a higher capacity). The default buffer size on the Juniper router was 190ms. We reduced it to 10ms, 5ms, 2.5ms and 1ms. Interestingly, for buffer sizes down to 5ms, we did not see a single packet drop for the duration of our experiments (5-7 days) even for load levels as high as 95%. For buffer sizes of 2.5ms and 1ms there were limited packet drops, below 0.2%.

2.2.3 Discussion

The *small buffers* rule makes two main assumptions: (1) That utilization is the right metric for buffer sizing in a router, and (2) When there are many flows, they aren't synchronized.

Utilization is an operator-centric metric – if a congested link can keep operating at 100% throughput then it makes efficient use of the operator's congested resource. It's not necessarily ideal for an individual end-user as the metric doesn't guarantee a short flow-completion time (*i.e.* quick downloads), or that there won't be too many packet drops. However, there is reason to think that this metric reflects short flow-completion times and appropriate numbers of packet drops. If the buffers are smaller (but not so small as to reduce throughput), then the round-trip time is reduced which for TCP leads to higher throughput for each flow, and they will complete quickly. Moreover, we can expect the feedback loop to be better behaved when the delay and delay variation is reduced. Packet drops are a mixed blessing: Lose too many and goodput suffers, lose too few and TCP's feedback loop doesn't work and the flow misbehaves, taking a long time to complete. In our experiments we've assumed that TCP will behave well with a drop rate of approximately 1%.

We believe that the metric of fairness needs closer attention. Because utilization is an average measure across flows, the utilization metric doesn't guarantee that some flow won't receive less goodput than others.

While simulations and experiments seem to indicate that with sufficient flows they become desynchronized, there is not uniform agreement. To understand the relationship between the number of flows and their synchronization, Raina and Wischik modeled a network with various buffer sizes [14]. They concluded that with the *small buffers* rule, the network is not stable, and should have low throughput – due to periodic changes in the aggregate window size, a direct consequence of synchronization. In our simulations we found small residual ripples, but much smaller than those predicted. We have not found evidence of the synchronization in experiments on real networks. It is possible that the differences arise from the fairness in packet drops in Raina and Wischik's mathematical model, compared to the unfair packet drops when TCP-Reno is combined with drop-tail queue management – as in Appenzeller *et al*'s case [17]. To the best of our knowledge, there is not a comprehensive (theoretical and experimental) study of synchronization in the presence of small buffers and with different queue management schemes that can explain these differences.

It isn't clear yet how to decide the value of N. In theory, it is the number of long-lived active flows (*i.e.* those that have started but not yet finished). In practice, we need to determine what fraction of flows are long-lived and short-lived – not an easy thing to do when designing the router for one buffer size. And so for now, we would cautiously conclude that at the core of the Internet, where the number of flows is very large, the buffers can be reduced by a factor of ten, without expecting any adverse change to the network behavior; in fact, we would expect delays and delay variation to be reduced. We believe more work is needed before reducing buffer sizes further.

2.3 Drop-based Buffers Rule

2.3.1 Assumptions and Intuition

Dhamdhere and Dovrolis [8] studied a particular network example to argue that when packet drop rates are considered, you can conclude that much larger buffers are needed – perhaps larger than the buffers in place today. They studied an example where a large number of flows share a heavily congested low capacity bottleneck link towards the edge of the network, and showed that one might get substantial packet drop rates (up to 17%).

In their example, a 50Mb/s link carries 200 long-lived

TCP flows, as well as some additional short-lived flows. The effective RTT of the flows is 60ms (*i.e.* the average congestion window size is about two packets). If $B = C \times T$ then the buffer will contain about 1500 packets. The small buffers rule suggests a buffer size of only about 100 packets.

Because of the high drop rate they measured, the authors propose *increasing* buffer sizes.

2.3.2 Discussion

These results show that we need to be careful when applying the small buffer rule – it probably isn't going to hold everywhere in the network, particularly towards the edge of the network, such as the experiment described above. In this scenario, the problem comes from congestion window dropping to such a low value that TCP starts to drop a lot of packets. Increasing the buffer size doesn't directly reduce the drop-rate in the way we might expect (*e.g.* like it would if the source were open loop). Increasing the size of the buffer will increase the propagation delay of each flow which, in turn, increases the average congestion window size to greater than two packets; the drop-rate goes down. It's not clear if we always want to keep the drop-rate low on a heavily congested link. After all, if the link is congested, we'd like to get the bad news quickly to the sources so they can reduce their window size. Increasing the buffer size only delays the feedback to the sender. On the other hand, large drop-rates eventually cause TCP performance to fall apart. This suggests a lower-bound on the buffer size, that may or may not come into play, depending on the speed of the link. In this example, where the link has quite a small capacity (50Mb/s), the buffer size will be small from any of the rules (it is only 1500 packets if $B = C \times T$).

2.4 Tiny Buffers Rule

Tiny buffers are most interesting for all-optical routers. In order to build an all-optical router, several problems need to be solved; and one of them is that we need to buffer the packets – something not easily done if the data stays in the optical domain. Recently, researchers have demonstrated small, integrated photonic circuits that can buffer a few packets in an on-chip switched optical delay line [13]. Larger all-optical buffers remain infeasible, except with unwieldy spools of optical fiber (that can only implement delay lines, not true FCFS packet buffers).

It is interesting to ask how performance of the network would be affected if we reduced the size of the buffers to just 10-20 packets. How much capacity would be lost, and what would become of the drop-rate?

2.4.1 Assumptions and Intuition

Raina and Wischik [14], and Enachescu *et al.* [10] suggested that we could build a network with tiny buffers if we are willing to sacrifice a small amount of throughput. For instance, when access links are much slower than the core links we have a natural smoothing of packet arrivals into core routers and with only a few dozen packets we can gain small drop rates and a throughput of 85-90%. Also, when access links have rates comparable to the core links, one can get the same results by using Paced TCP [2]. The 10-15% reduction in throughput might be a reasonable trade-off in the context of all-optical network where capacity is abundant, and buffers are the bottleneck.

When we have slow access links, or use Paced TCP, the traffic coming from individual sources is not bursty. Now, if each source injects packets independent of other sources, we can show that the aggregate traffic will not be bursty, and looks like a Poisson process. Now, if arrivals to a queue are Poisson, and if the link load is below 100% (for example 80%), we can show that the drop rate will be very small, and thus we will gain a high throughput.

2.4.2 Validation

Recently, there has been a number of experiments on tiny buffer sizing model performed by us in Sprint ATL, and also by other groups at Verizon Communications, and Lucent Technologies. In these experiments, commercial traffic generators, or clusters of Linux boxes are used to generate up to 1Gb/s of live TCP traffic (mainly FTP, and HTTP). This traffic is fed to a commercial router with reduced buffer sizes. The performance (throughput, packet drops, delay) is measured using special purpose measurement equipment, as well as the statistics collected by the router, and traffic generators.

These preliminary experiments, show very small degradation in performance in the presence of tiny buffers, and suggest the possibility of having a network with just 20-50 packets under the constraints mentioned before. However, there are many more scenarios and boundary cases to consider before deciding it is time to reduce buffers to just 20-50 packets. Meanwhile, more experiments are needed to fully understand these results.

2.4.3 Discussion

Tiny buffers rule assumes we are willing to sacrifice some throughput, and for example operate the network at 85-90% utilization. This might sound wasteful at first glance. However, we should note that in an all-optical network network capacity is abundant, and the buffer size is the bottleneck. Additionally, even in today's opto-electronic networks, most Internet core networks are operated at extremely low link utilizations (*e.g.* 20-30%). These two points might justify the 10-15% reduction in throughput as a result of tiny buffers rule.

3. NETWORK EVOLUTION

We have clearly seen the impact of different assumptions on buffer sizing results. One reason different people have different assumptions for their buffer sizing studies is the constant evolution of networks, specially the Internet. We tend to design systems to operate in typical scenarios. The constant changes in network, necessitates constant examination and revision of the underlying assumptions [9]. For example, at the time TCP was invented, it was quite typical for the system to have very few flows, and since long-haul links were very expensive, the systems were designed to keep long-haul links at 100% throughput, thus we have the rule of thumb. In an all-optical network however, a 10-15% reduction in throughput is not a pressing issue anymore. This is a major shift in assumptions, and has significant impact on the results as we have seen.

4. OTHER ISSUES IN BUFFER SIZING

Internet routers are complex systems consisting of sophisticated hardware and software components. Any incoming packet to the router goes through several stages of processing, buffering, and possible segmentation and reassembly.

Mathematically modeling all these stages is very complicated, and that's why almost all the buffer sizing results described so far are based on a simple output-queued switch model. In practice, building an output-queued router is very difficult due to its high speedup requirements. Most routers today are based on Combined Input-Output Queued (CIOQ) architectures.

Beheshti *et al.* studied the buffer sizing problem in a CIOQ router rather than the output-queued model used in previous results [4]. Based on the tiny buffer sizing result, they showed that it is feasible to create a CIOQ router with a speedup of just two, and buffers are small as a few dozen packets. This is extremely important for filling up the gap between theoretical results and practical systems with tiny buffers.

In the context of all-optical routers, another interesting problem related to buffer sizing is the architecture of the optical buffering system [7, 12]. Sarwate and Anantharam showed how one can build an optical buffer which can hold K packets by combining optical delay lines and a switch of size $\sqrt{K}$ [15]. C.S. Chang *al.* showed how the switch size can be reduced to just $\log K$ [6, 5].

5. CONCLUSION

We can study the impact of buffer sizing on network performance along five different axes: traffic patterns, network topology and settings, router architecture and settings, dynamics of the network, and the appropriate performance metric. All the previous theoretical, simulation-based, and experimental studies of buffer sizing, are based on simplified scenarios in most/all of these directions. For a comprehensive study on buffer sizing, we need to explore all these axes and their combinations.

Studying network performance along all these directions and their combinations seems extremely difficult if not infeasible. We need to develop theoretical tools that can describe complex traffic patterns, take into account feedback loops in the system, and are able to describe both the transient and equilibrium state of the system. Such theoretical tools must easily lend themselves to experimental validation.

As part of a larger attempt in gaining a better understanding of the buffer sizing, we are working on creating FPGA-based switches called *Controllable and Observable Buffer (COB) routers*, as well as FPGA-based tools for traffic generation, and measurement [1]. Our goal is to be able to perform buffer sizing experiments under realistic settings (*e.g.* more complex topologies, different traffic patterns, presence of failures, . . .). Hopefully, this work along with the work of others will lead to a better understanding of the buffer sizing problem, and thus better networks for the future.

6. REFERENCES

[1] NetFPGA project. http://yuba.stanford.edu/NetFPGA/.

[2] A. Aggarwal, S. Savage, and T. Anderson. Understanding the performance of TCP pacing. In *Proceedings of the IEEE INFOCOM*, pages 1157–1165, Tel-Aviv, Israel, March 2000.

[3] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *SIGCOMM '04*, pages 281–292, New York, NY, USA, 2004. ACM Press.

[4] N. Beheshti, Y. Ganjali, R. Rajaduray, D. Blumenthal, and N. McKeown. Buffer sizing in all-optical packet switches. In *Proceedings of OFC/NFOEC*, Anaheim, CA, March 2006.

[5] C. S. Chang, Y. T. Chen, and D. S. Lee. Constructions of optical FIFO queues. *IEEE Transactions on Information Theory*, 52(6):2838–2843, June 2006.

[6] C. S. Chang, D. S. Lee, and C. K. Tu. Recursive construction of fifo optical multiplexers with switched delay lines. *IEEE Transactions on Information Theory*, 50(12):3221–3233, December 2004.

[7] R. L. Cruz and J. T. Tsai. COD: alternative architectures for high speed packet switching. *IEEE/ACM Transactions on Networking*, 4(1):11–20, February 1996.

[8] A. Dhamdhere and C. Dovrolis. Open issues in router buffer sizing. *ACM Sigcomm Computer Communication Review*, 36(1):87–92, January 2006.

[9] N. Dukkipati, Y. Ganjali, and R. Zhang-Shen. Typical versus worst case design in networking. In *Proceedings of the Fourth ACM Workshop on Hot Topics in Networks (HotNets-IV)*, College Park, Maryland, November 2005.

[10] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden. Routers with very small buffers. In *Proceedings of the IEEE INFOCOM'06*, Barcelona, Spain, April 2006. ¡br¿Also available as technical report TR05-HPNG-060606, High Performance Networking Group, Stanford University.

[11] Y. Ganjali and N. McKeown. Experimental study of router buffer sizing. Manuscript. Also available as technical report, HR06-HPNG-07-30-00, Stanford University, July 2006.

[12] D. K. Hunter, M. C. Chia, and I. Andonovic. Buffering in optical packet switches. *Journal of Lightwave Technology*, 16:2081–2094, December 1998.

[13] H. Park, E. F. Burmeister, S. Bjorlin, and J. E. Bowers. 40-gb/s optical buffer design and simulations. In *Numerical Simulation of Optoelectronic Devices (NUSOD)*, 2004.

[14] G. Raina and D. Wischik. Buffer sizes for large multiplexers: Tcp queueing theory and instability analysis. In *EuroNGI*, Rome, Italy, April 2005.

[15] A. D. Sarwate and V. Anantharam. Exact emulation of a priority queue with a switch and delay lines. *Queueing Systems: Theory and Applications*, 53(3):115–125, July 2006.

[16] C. Villamizar and C. Song. High performance TCP in ANSNET. *ACM Computer Communications Review*, 24(5):45–60, 1994.

[17] M. Wang and Y. Ganjali. Unifying buffer sizing results through fairness. Manuscript submitted for publicatoin. Also available as technical report, HR06-HPNG-060606, Stanford University, June 2006.