Updating the Theory of Buffer Sizing

Extended Abstract

Bruce Spang
Stanford University Serhat Arslan
Stanford University Nick McKeown
Stanford University

Internet routers have packet buffers which reduce packet loss during times of congestion. Sizing the router buffer correctly is important: if a router buffer is too small, it can cause high packet loss and link under-utilization. If a buffer is too large, packets may have to wait an unnecessarily long time in the buffer during congested periods, often up to hundreds of milliseconds. While an operator can reduce the operational size of a router buffer, the maximum size of a router buffer is decided by the router manufacturer, and the operator typically configures the router to use all the available buffers. Without clear guidance about how big a buffer needs to be, manufacturers tend to oversize buffers and operators tend to configure larger buffers than necessary, leading to increased cost and delay.

This paper revisits two widely used rules of thumb for sizing router buffers in the internet. The two rules cover two different cases:

Case 1: When a network carries a single TCP Reno flow. Van Jacobson observed in 1990 [1] that a bottleneck link carrying a *single* TCP Reno flow requires a router buffer of size $B \geq \text{BDP}$, the bandwidth-delay product, in order to keep the link fully utilized.

Case 2: When a network carries multiple TCP Reno flows. Appenzeller, Keslassy, and McKeown argued in 2004 [2] that a bottleneck link carrying n long-lived TCP Reno flows requires a buffer of size $B \geq \text{BDP}/\sqrt{n}$ in order to keep the link highly utilized.

Much has changed since these rules were first introduced, and it is not clear whether these rules still apply in modern networks. The behavior of TCP Reno has changed; most notably when Rate-Halving [3, 4] and PRR [5] were introduced. New types of congestion control have become widespread, such as Cubic [6] (default in Linux, Android, and MacOS), and more recently BBR (deployed by Google for YouTube) [7] and BBRv2 [8]. Given that the analysis underlying both buffer sizing rules depends on the specific way in which TCP Reno halves the congestion window when losses are detected, there is no particular reason for either rule to still hold in today's internet.

Existing rules of thumb help us pick the buffer size to achieve full link utilization, and do not predict behavior if the buffer is made smaller. Thus, theory falls short for recent congestion control algorithms (e.g. BBR and BBRv2) which no longer aim to keep a bottleneck link running at 100% utilization. Instead, they rely on short periods of under-utilization to keep queueing delay low and to estimate propagation delay.

In light of these changes, this paper examines buffer sizing for modern TCP algorithms. We show that the two rules still allow TCP Reno to fully utilize a link, despite changes due to Rate-Halving and PRR. We show that TCP Cubic, Scalable TCP, and BBR allows us to reduce buffer sizes.

We extend our analysis for the case when link utilization is less than 100%, and we show that very small buffers can still allow high (but not 100%) link utilization. More generally, *this paper sheds new light on how to size buffers for a given congestion control algorithm and desired link utilization, under a very broad set of conditions.* In doing so, we also show how future congestion control algorithms can be designed to further reduce buffer requirements.

Throughout the paper, we will illustrate and validate our results using measurements drawn from a physical network in our lab. This is challenging: while Linux can capture per-packet measurements in the end-host TCP stack, it is not normally possible to capture the full time series of buffer occupancy at the switch. Our measurement setup uses a P4-programmable Tofino switch which we program to report the precise time evolution of its buffer, to approximately 1 nanosecond resolution. This lets us precisely compare the evolution of the congestion window and the buffer size and validate our theoretical results.

Contributions: The main contributions of this paper are:

1. Single flow case: A simple proof of TCP's required buffer size, applicable to the latest versions of TCP Reno, as well as other algorithms including Cubic, Scalable TCP, and BBR.

2. Multiple flow case: A new, more general model of how buffer size is impacted by fairness and the amount of worst-case packet drops, and square root of n-style rules for TCP Reno and other algorithms.

3. A better understanding of how congestion control algorithms interact with buffers, including how utilization depends on buffer size, how algorithms can reduce buffer requirements, and how current congestion measurement techniques rely on certain algorithmic behavior.

4. A new measurement platform allowing precise observation of TCP and the router buffer.

The full version of our paper can be accessed at [9].

Min. Buffer Size	Additional assumptions beyond Section ??	Citation
BDP	Reno, silence after loss	[1, 10]
BDP	Reno	[11], [9]
$(1/b-1)$BDP	Multiplicative decrease by b, silence after loss	[12–14]
$(1/b-1)$BDP	Multiplicative decrease by b	[9]
$\frac{3}{7}$BDP	Cubic	[13], [9]
$\frac{1}{7}$BDP	Scalable TCP	[9]
$\frac{1}{4}$BDP	BBR during the probe bandwidth phase, with loss	[9]
$\Theta(BDP/\sqrt{n})$	Reno, windows are i.i.d. uniform random variables	[2]
$O(\mathrm{BDP}/\sqrt{n})$	$\sqrt{n}+O(n^2/BDP)$ almost fair flows see loss	[9]
$O(\mathrm{BDP}/\sqrt{n})$	Almost fair BBR flows in probe bandwidth phase	[9]
$O(p \cdot \mathrm{BDP} - n + np)$	A p fraction of fair flows see losses	[11]
$O(s \cdot \mathrm{BDP}/n)$	At most $s + n^2/BDP$ almost fair flows see loss.	[9]
$O(1)$	Reno, bounded window size, Poisson pacing	[15]
$O(1)$	$\sqrt{n}+O(n^2/BDP)$ flows see loss, utilization is $\Omega(1-1/\sqrt{n})$	[9]

Table 1: Minimum buffer sizes required for full link utilization, our new results highlighted in gray.

1. REFERENCES

[1] V. Jacobson, "Modified TCP congestion avoidance algorithm," Apr. 1990. [Online]. Available: ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt

[2] G. Appenzeller, N. McKeown, and I. Keslassy, "Sizing router buffers," in *ACM SIGCOMM Computer Communication Review*. ACM, Aug. 2004, pp. 281–292. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1015467.1015499

[3] J. C. J. C.-I. Hoe, "Start-up dynamics of TCP's congestion control and avoidance schemes," Thesis, Massachusetts Institute of Technology, 1995. [Online]. Available: https://dspace.mit.edu/handle/1721.1/36971

[4] J. Semke, J. Mahdavi, and M. Mathis, "The Rate-Halving Algorithm for TCP Congestion Control." [Online]. Available: https://tools.ietf.org/html/draft-mathis-tcp-ratehalving-00

[5] N. Dukkipati, M. Mathis, Y. Cheng, and M. Ghobadi, "Proportional Rate Reduction for TCP," in *Internet Measurement Conference*, Nov. 2011, p. 15.

[6] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, Jul. 2008. [Online]. Available: https://dl.acm.org/doi/10.1145/1400097.1400105

[7] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and Van Jacobson, "BBR: Congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, pp. 58–66, Jan. 2017. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3042068.3009824

[8] N. Cardwell, Y. Cheng, S. H. Yeganeh, I. Swett, V. Vasiliev, P. Jha, Y. Seung, M. Mathis, and V. Jacobson, "BBR v2: A Model-based Congestion Control," Prague, Mar. 2019. [Online]. Available: https://www.ietf.org/proceedings/104/slides/slides-104-iccrg-an-update-on-bbr-00

[9] B. Spang, S. Arslan, and N. McKeown, "Updating the theory of buffer sizing," *Performance Evaluation*, 2021. [Online]. Available: https://arxiv.org/abs/2109.11693

[10] C. Villamizar and C. Song, "High Performance TCP in ANSNET," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 5, pp. 45–60, Oct. 1994. [Online]. Available: http://doi.acm.org/10.1145/205511.205520

[11] A. Dhamdhere, Hao Jiang, and C. Dovrolis, "Buffer sizing for congested internet links," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 2. Miami, FL, USA: IEEE, 2005, pp. 1072–1083. [Online]. Available: http://ieeexplore.ieee.org/document/1498335/

[12] S. Hassayoun and D. Ros, "Loss synchronization and router buffer sizing with high-speed versions of TCP," in *IEEE INFOCOM Workshops 2008*, Apr. 2008.

[13] W. Lautenschlaeger and A. Francini, "Global synchronization protection for bandwidth sharing TCP flows in high-speed links," in *2015 IEEE 16th International Conference on High Performance Switching and Routing (HPSR)*, Jul. 2015, pp. 1–8.

[14] N. McKeown, G. Appenzeller, and I. Keslassy, "Sizing Router Buffers (Redux)," *ACM SIGCOMM Computer Communication Review*, vol. 49, no. 5, p. 6, 2019.

[15] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Routers with Very Small Buffers." *INFOCOM*, pp. 1–11, 2006. [Online]. Available: http://ieeexplore.ieee.org/document/4146893/