Buffer-bloated router?

How to prevent it and improve performance

Home broadband routers suffering bufferbloat can lead to a poor user experience when accessing the Internet

Philippa Harrison

Bufferbloat occurs when too many data packets are queued in a router's buffer waiting to be sent. While buffering is needed to reduce data packet loss, overly large buffers lead to increased delay, causing poor performance. It is not well understood how large a buffer should be, nor how the size affects network performance. Many consumer-grade router manufacturers do not use firmware that can prevent bufferbloat. Reducing bufferbloat by another means seems to be required. An experiment is conducted to demonstrate how bufferbloat can be mitigated on a home broadband router. A number of tests are conducted with different router configurations, measuring the delay under varying loads. An approach of using data stream shaping and smart queue management (SQM) to mitigate bufferbloat is validated. The delay is low and does not significantly vary under load. Thus, bufferbloat is mitigated and user experience can be improved.

Additional Keywords and Phrases: Bufferbloat, SQM, Data stream shaping, Home broadband routers

1 INTRODUCTION

Devices in a home network connect to the router which connects to the ISP to access the Internet. An ISP could connect to a larger ISP before connecting to a fibre-optic "backbone" for an entire nation or region. To access a web server, for example, to read a news story on the BBC website, data would travel from a device on the home network via numerous devices to reach the destination – the BBC webserver.

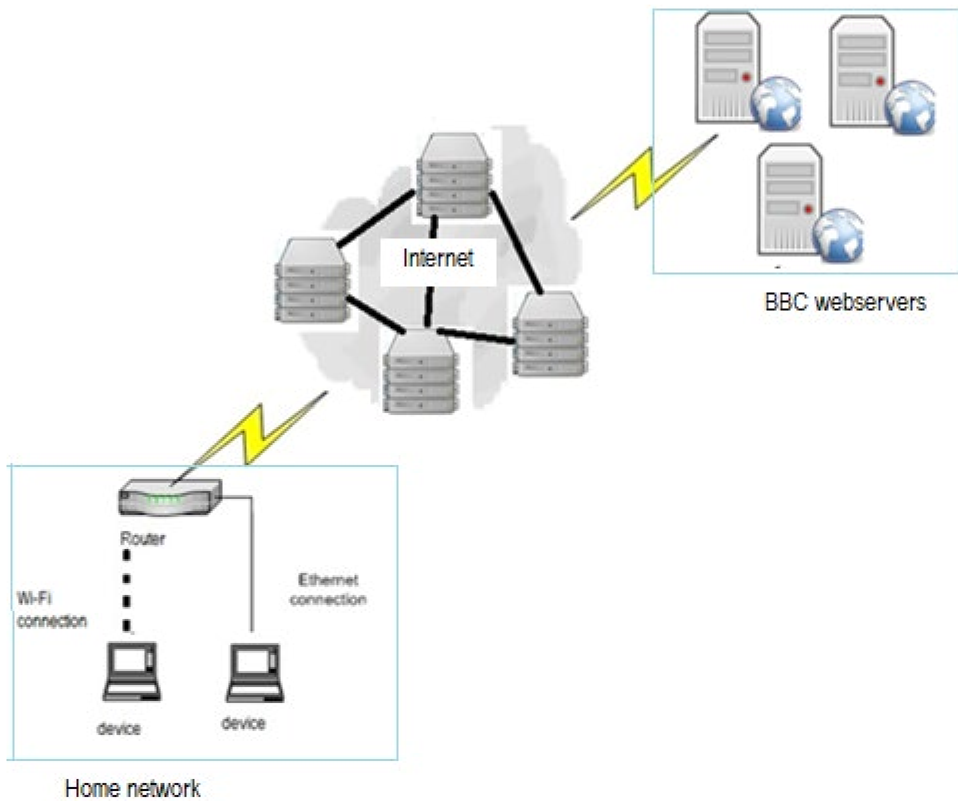


Figure 1: Theoretical devices along the path between a home computer and a web server

Multiple devices on the home network can be sending and receiving data at the same time as our device accessing the BBC webserver. These independent data traffic streams intersect at the home broadband router. The router has a number of input ports allowing multiple devices to send data outside of the home network and onto the Internet. Data packets are transmitted from output port(s) on a router. Typically, a home broadband router has one output port that connects a line to an ISP.

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are part of the Internet Protocol (IP) suite. TCP and UDP send data across the internet. TCP ensures delivery of data packets but is slower than UDP which is "best effort". A receiver acknowledges receipt of the TCP data packets. If a sender does not receive an acknowledgement for a data packet, they will retransmit it.

Our device accessing the BBC webserver does not know what the link's capabilities are. It uses TCP slow to avoid creating congestion by slowly increasing the number of packets sent until the maximum capacity is detected. TCP uses flow control to ensure that the webserver (receiver) is not overwhelmed by our device's (sender) data packets. This is accomplished by the receiver "advertising" its Receive Window size. Although the sender sends its data packets as fast as the advertised receive window, congestion can still occur at any device along the path. Figure 1 shows that there are many devices along the path between the home network device (the sender) and the BBC webserver (the receiver). Thus, the speed can be affected by the sender's performance, the receiver's performance, and the devices along the path between the two.

Data packets travelling the path can be queued or dropped due to congestion. Congestion occurs when the load (amount of data packets) is higher than the line's capacity. The TCP congestion control mechanism aims to avoid congestion by using the flow of sent data packets and acknowledgements to determine a send rate. If a TCP sender detects little or no congestion along the path between it and the receiver, it increases the transmission rate. If a sender sends too many data packets, this results in the receiver dropping data packets. Data packets that are dropped are not acknowledged. Likewise, if a data packet is not acknowledged quickly enough, it is assumed to have been dropped. Dropped data packets signal to the sender that the transmission rate is higher than capacity and the transmission rate is reduced. Thus, TCP congestion control is "feedback-control".

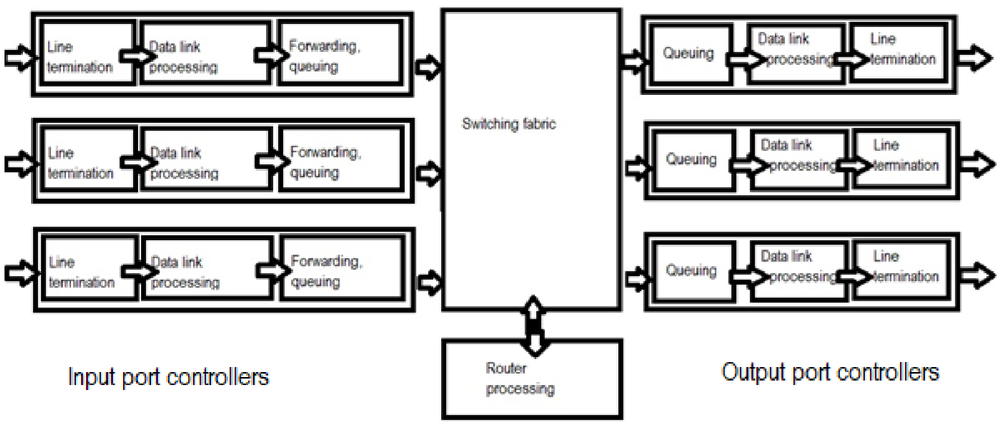


Figure 2: Buffers on a router

Latency is the combined transmission delay, processing delay and queuing delay. The queuing delay is dependent on the queue size. Thus, with excessively large buffers bufferbloat can result. Routers use buffers to temporarily store data packets that are awaiting transmission when congestion occurs. These data packets (both TCP and UDP) are queued so that they are not lost and can be sent when bandwidth becomes available on the line. These buffers are required to maintain the flow of data packets at the maximum transmission rate. A problem arises if a device along the path has a large buffer, data packets can wait a long time before they are sent. As data packets are not dropped, the sender does not receive information that the line's capacity has been exceeded. A timeout will eventually signal that there is congestion. This delay in feedback affects performance. If the delay is very large, the feedback provides an inaccurate state of the network. Thus, the mechanisms do not respond as they should when congestion occurs, i.e. reducing the data packet send rate.

Bufferbloat does affect other protocols as well as TCP, partly as a result of the same buffers being used as for TCP connections. The router's buffer can fill of data before packets begin to drop. TCP packets could be queued ahead of interactive applications. Bufferbloat occurs as too many data packets are queued which results in an increase in delay. [2] This leads to the performance of applications degrading. The effect can be seen in the figure below. If a user is participating in a video call, the result of delay could mean that they are seeing a camera image from 5 or more seconds ago.

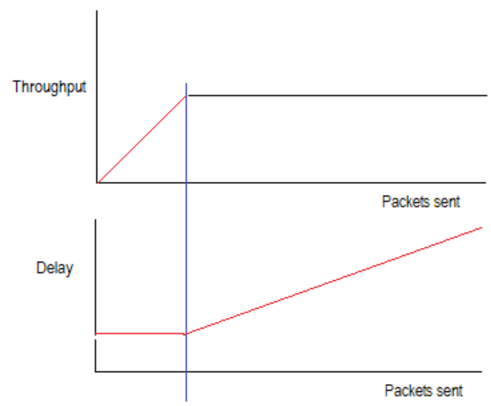


Figure 3: Throughput and delay when there are large buffers

Delay sensitive applications that bufferbloat commonly affects are as follows:

- VoIP calls
- video conferencing
- online gaming
- video streaming
- music streaming

Compared to today's routers, older routers had small buffers. These buffers filled more quickly and thus, packets would be dropped quickly after a line was saturated. Bufferbloat "has led Internet delays to occasionally exceed the light propagation delay from the Earth to the Moon" [3]. Newer routers have larger buffers which can often hold the equivalent to approximately 10 seconds of data. Thus, 10 seconds worth of data can be sent without feedback of packets being dropped [4]. This results in the TCP sawtooth shown in the figure below. TCP slowly increases the data rate as capacity on the line is available (seen as the line rising slowly). When congestion occurs, it quickly decreases the rate (seen as the line decreasing sharply).

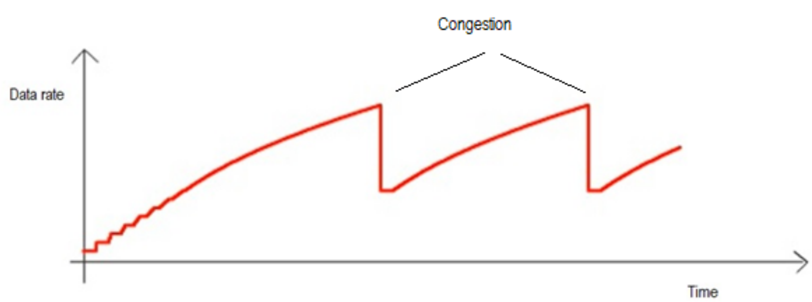


Figure 4: TCP "sawtooth"

Buffers are larger today as the cost of memory has fallen. It is not well understood how large a buffer should be, nor how the size affects network performance. Router buffers are typically sized based on a rule attributed to Villamizar and

Song's 2004 paper, "High performance TCP in ANSNET" [5]. The increased size of buffers can be seen to prevent network congestion avoidance algorithms from functioning correctly. On most home broadband routers, the user cannot change the buffer size.

Quality of Service (QoS) settings on a router do not effectively reduce bufferbloat. As some data traffic is prioritised, the rest is queued behind it. This means all data traffic in the buffer still needs to be sent.

A high-speed connection does not prevent bufferbloat. The delay can still be high. Many applications are more sensitive to delay than they are to low bandwidth.

Bufferbloat can be mitigated by Smart Queue Management (SQM) algorithms. SQM performs per-packet/per-flow network scheduling, active queue length management (AQM), traffic shaping/rate limiting, and QoS. SQM puts traffic from a single IP address or port into its own queue. This means, unlike when using QoS, queues do not become too long as data packets from flows that have a small or no queue are prioritised. If a queue becomes too large, a certain percentage of data packets are dropped to allow congestion avoidance to take effect.

Many consumer-grade router manufacturers do not use firmware that can prevent bufferbloat [6]. As the design of routers has not changed for some years and appears not to be changing soon, reducing bufferbloat by another means seems to be required. An experiment is conducted to demonstrate how the effects of bufferbloat can be mitigated.

2 MEASURING BUFFERBLOAT

The aim of the experiment is to validate the approach of using data stream shaping and SQM to prevent bufferbloat, thus improving the user experience. A number of tests are conducted with different router configurations, measuring the latency (delay) under varying loads. To signify that bufferbloat is not an issue, there should be no variation in the latency between unloaded and loaded tests run.

During the experiment, various loads are created by video streaming and file downloading. Different loads are created by using one or two devices to stream video of different resolutions. The video resolution is changed so that bandwidth requirement increases. Tests include streaming 720p video, 1080p (High Definition) video, and 2160p video (4K). Additional tests are conducted to include downloading a file on one device whilst streaming video simultaneously. To saturate the bandwidth, a speed test where the download portion only is performed is run simultaneously.

To enable SQM on the router, the SQM package for the OpenWRT firmware is installed, luci-app-sqm. The queue discipline used is cake, and the script is pieces_of_cake.qos. The SQM link layer adaptation settings used are ATM and 44byte overhead. The router configuration is changed to enable and disable data stream shaping and SQM. During the experiment when data stream shaping is used, the upstream and downstream are shaped to use 95% of the line's capacity.

For each test, the minimum, maximum, and average latency is recorded as well as data packets lost, by conducting a ping test. If the latency increases significantly with load, this indicates bufferbloat is experienced. Additionally, if the video being streamed buffers, freezes, or pixelates during a test, it is recorded. For tests where the file download is performed, the time taken to download the file is recorded.

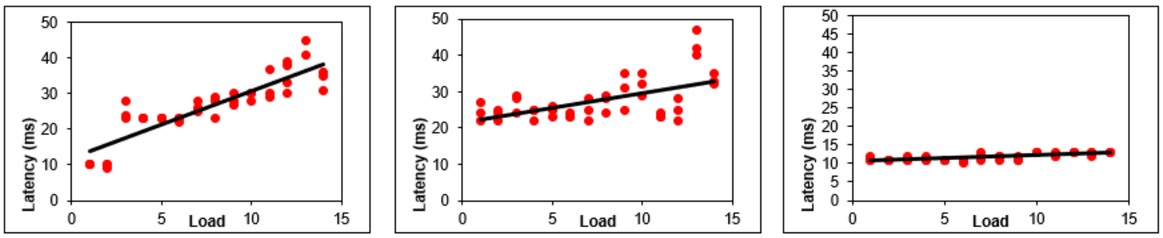
Some factors are outside of the control of the researcher: the speed of YouTube servers (for video streaming), Microsoft servers (for file downloading), Google Domain Name System (DNS) servers (for measuring the latency), and Speedof.me servers (for saturating the bandwidth). As a result, each test is repeated three times to detect any anomalies.

3 THE EFFECT OF BUFFERBLOAT

The experiment validates the approach of using SQM and data stream shaping to mitigate bufferbloat and improve user experience. The latency when SQM and data stream shaping are enabled is low and does not significantly vary under load.

Thus, bufferbloat is mitigated. Streamed video does not buffer, freeze, or pixelate when viewed. The average Round-trip Time (RTT) resulting from the ping test from the three tests is used, rounded to the nearest whole number. A high average RTT would mean bufferbloat is being experienced compared to a consistently low average RTT which means no bufferbloat.

The following graphs show plots for each load's average RTT with a trendline for each configuration. Latency is lowest when data streaming and SQM are both enabled.



SQM and shaping are disabled | SQM is enabled but shaping is not used | SQM and shaping are enabled

Figure 5: Latency for each router configuration

4 PREVENTING BUFFERBLOAT

The experiment validates the approach of mitigating bufferbloat by using data stream shaping and SQM. The variance between latency for each load is the lowest when SQM and data stream shaping are used. This signifies that bufferbloat is not experienced. User experience can be improved. When these mechanisms are used, streaming video does not buffer, freeze, or pixelate. As latency is consistently low, delay sensitive applications are not affected. People can simultaneously stream video, play online games, video conference, and conduct VoIP calls without experiencing delay or unstable applications.

The file download times increase when SQM and data stream shaping are enabled as video traffic is prioritised.

Data stream shaping does result in a small loss of bandwidth. However, it is necessary for SQM to work most effectively.

5 CONCLUSION

Bufferbloat was uncovered in 2011 by Getty. [1] Since that time, a lot of Internet traffic has moved towards small bursts dependent on RTTs (Round-trip times) or using rate-limited streaming to avoid bufferbloat. [7] Rate limiting controls the amount of incoming and outgoing traffic to or from a network. However, the issue of bufferbloat still exists.

Router capacity has doubled every 24 months following Moore's Law in 1965. Moore's Law explains how processing speed, memory capacity, sensors, et cetra, are improving at exponential rates as their price decreases exponentially. Routers now have more memory but router speed has been limited by memory access speed which has not improved since 2011. Routers typically have DRAM (Dynamic Random Access Memory). The access speed of DRAM is not as fast as SRAM (Static Random Access Memory) but is cheaper. Thus, router memory is typically designed for large storage and not for speed. [8]

The approach demonstrated in this article requires some configuration to attain the best performance from a router. As ISPs typically provide routers for free, people are unlikely to pay more for a router with SQM. The OpenWRT firmware web interface is likely to be beyond the understanding of most home broadband users. Additionally, a user would need to perform firmware upgrades themselves and support for the router would not be provided. Consumer grade and ISP provided routers are intended to be easy to set-up and operate. There are reports that some ISP providers are adding SQM to the routers they provide. [9] However, as demonstrated in this experiment, SQM in combination with data stream shaping provides the best performance. As ISPs seek to reduce customer service costs and the process of data stream shaping would require technical skills beyond most home users' abilities, it is unlikely to be seen as economically viable for ISPs to promote the use of data stream shaping.

Bufferbloat can be reduced but it requires buy-in from the ISPs to use routers with SQM and router manufacturers to change their designs.

6 FURTHER RESEARCH

This experiment demonstrates the use of SQM and data stream shaping on a single home broadband router to improve performance. The SQM FQ_CoDel (Fair Queueing Controlled Delay) seems a good algorithm. Further research to model the effect of the widespread deployment of this algorithm is recommended to ascertain that there would be no negative effects. As resources (link capacity) are finite, would a large number of routers using SQM result in resources not being shared fairly?

REFERENCES

[1] Jim Gettys. 2011. Bufferbloat: Dark Buffers in the Internet, IEEE Internet Computing 15, 3. 96-96. DOI: 10.1109/MIC.2011.56.

[2] Kathleen Nichols and Van Jacobson. 2012. Controlling Queue Delay: A modern AQM is just one piece of the solution to bufferbloat. *Queue* 10, 5 (May 2012), 20–34. DOI:https://doi.org/10.1145/2208917.2209336

[3] C. Chirichella, D. Rossi, C. Testa, C., T. Friedman, and Antonio Pescapé. 2013. Remotely Gauging Upstream Bufferbloat Delays. In: Roughan, M., Chang, R. (eds) Passive and Active Measurement. PAM 2013. Lecture Notes in Computer Science, 7799. Springer, Berlin, Heidelberg. Retrieved April 11, 2022 from https://doi.org/10.1007/978-3-642-36516-4_25

[4] Bufferbloat.net. Technical Introduction to Bufferbloat. Retrieved April 11, 2022 from: https://www.bufferbloat.net/projects/bloat/wiki/TechnicalIntro/

[5] Guido Appenzeller, Isaac Keslassy, and Nick McKeown. 2004. Sizing Router Buffers. Retrieved April 12, 2022 fromt: https://web.stanford.edu/class/cs244/papers/SizingRouterBuffersAppenzeller.pdf

[6] Bufferbloat.net. 2021. What Can I Do About Bufferbloat? Retrieved April 9, 2022 from: https://www.bufferbloat.net/projects/bloat/wiki/What_can_I_do_about_Bufferbloat/

[7] Adam McFillin. 2020. Bufferbloat may be solved, but it's not over yet, APNIC, January 2022. Retrieved 5 June, 2022 from: https://blog.apnic.net/2020/01/22/bufferbloat-may-be-solved-but-its-not-over-yet/

[8] Nick McKeown. 2022. The route to success, BCS, Retrieved 5 June, 2022 from: https://www.bcs.org/articles-opinion-and-research/the-route-to-success/

[9] Sam Crawford. 2021. Bufferbloat, SamKnows Spotlight, Episode 3, June 2021. Retrieved 5 June, 2022 from: https://samknows.com/blog/bufferbloat