Interaction or Interference: can AQM and Low Priority Congestion Control Successfully Collaborate?

Y. Gong, D. Rossi, C. Testa, S. Valenti*
Télécom ParisTech
46 Rue Barrault
75013 Paris France
first.last@enst.fr

M. D. Täht
bufferbloat.net
dave.taht@gmail.com

ABSTRACT

Heterogeneity in the Internet ecosystem sometimes turns interaction into interference. Over the years, active queue management (AQM) and end-to-end low-priority congestion control (LPCC) have been proposed as alternative solutions to counter the persistently full buffer problem – that recently became popular under the "bufferbloat" term.

In this work, we point out the existence of a negative interplay among AQM and LPCC techniques. Intuitively, as AQM is designed to penalize the most aggressive flows it mainly hit best effort TCP: it follows that LPCC is not able to maintain its low priority, thus becoming as aggressive as TCP. By an extended set of simulation on various AQM policies and LPCC protocols, including the very recent CoDel AQM and LEDBAT LPCC proposals, we point out that this interference is quite universal and deserves further attention.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network communications; C.2.5 [**Local and Wide-Area Networks**]: Internet (e.g., TCP/IP)

Keywords

Bufferbloat, AQM, Scavenger protocol, Simulation

1. INTRODUCTION

Internet is a very heterogeneous ecosystem, where multiple protocol species coexist, evolve, and sometimes extinguish. TCP is a good example of this evolution, as over the years numerous species proliferated under this protocol family. Since the most widespread flavors of TCP follow a loss-based design, an old problem resurged in recent years: namely, the "persistently full" buffer problem, which was nicknamed "bufferbloat" [2]. Bufferbloat refers to an excess of buffering that is exacerbated by two factors: (i) TCP loss-based design fills up the bottleneck buffer before the sender reduces its rate and (ii) larger buffer in front of low-capacity links possibly lead to multiple-seconds of queuing delay [4].

This problem is well known since the 90s, to which two classes of solutions have been proposed. First, researchers focused on *active queue management (AQM)* techniques deployed inside the network that, e.g., aims at reducing TCP sending rate by intentional packet drop. Yet, despite numerous AQM proposals such as RED [3], SFQ [6] and, very recently CoDel [7], they have so far encountered limited adoption.

Since the early 2000, researchers turned their attention to *low priority congestion control (LPCC)* such as NICE [12], TCP-LP [5] and LEDBAT [10] as alternative solutions. This approach intends to transfer at a lower priority by reacting faster to network congestion using indicators other than packet loss . Different from AQM, LPCC is an end-to-end solution, and is presently widely deployed – about half of BitTorrent traffic is now carried over LEDBAT [1].

The temporal separation of the AQM vs. LPCC research topics results in the lack of in-depth investigation of interaction. In this work, we focus on the coexistence of best effort TCP CC and Low Priority CC (LPCC) transiting a bottleneck link governed by AQM. Our simulation-based investigation shows that *that AQM can induce a reprioritization of heterogeneous CC flavors*: in other words, while AQM succeeds in limiting bottleneck buffer thus reducing the queuing delay, it cancels the different priorities introduced by LPCC.

2. AN ILLUSTRATIVE EXAMPLE

Fig. 1, based on `ns2` simulation results, shows a typical change of link utilization breakdown due to AQM. In this case, 5 TCP NewReno and 5 LEDBAT backlogged flows (homogeneous RTT delay) share the same bottleneck with a capacity equals to $C = 10$ Mbps and the buffer up to $Q_{max} = 500$ packets. Plots report the temporal evolution of each flow throughput, and are additionally annotated with average queue size in packets ($E[Q]$), the capacity share exploited by the best effort TCP aggregate ($TCP\%$), and the average link utilization (η).

The DropTail case (left plot) shows a typical bufferbloat with nearly 400 TCP packets being queued on average. The introduction of LEDBAT exploits the spare capacity left unused by TCP but doesn't add delay thanks to its operation in a lower-than-best-effort mode [9].

---

[*] Current affiliation: Google Inc.

CoNEXT Student'12, December 10, 2012, Nice, France.

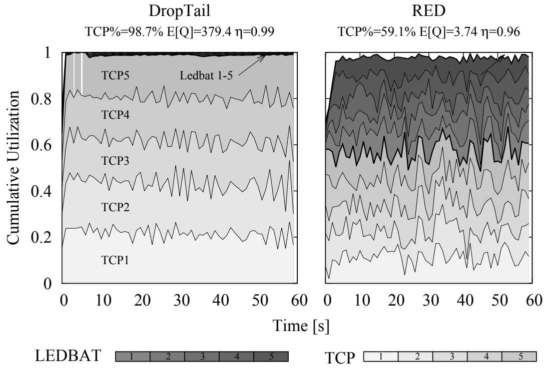

Figure 1: LPCC vs. AQM interference: AQM induces reprioritization among CC.

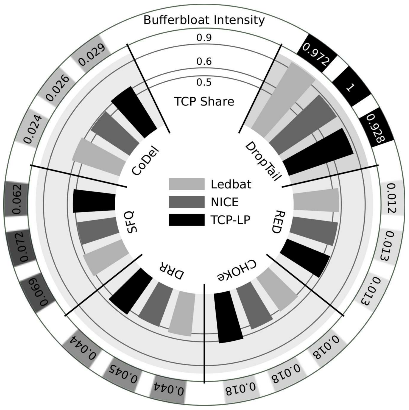

Figure 2: Joint impact of AQM and LPCC on the queuing delay and TCP breakdown.

The RED case (right plot) solves the bufferbloat, as the queue size is limited to less than 4 packets on average, at the price of a slight 3% reduction of the link utilization. Most important, however, RED invalidates LEDBAT low priority: the plot clearly illustrates the similar throughput of TCP and LEDBAT, both at flow and aggregate levels.

3. EXTENDED SIMULATION CAMPAIGN

To verify the extent of the reprioritization phenomenon, we performed and extended set of simulation considering multiple AQM techniques (namely, Choke [8], RED [3], DRR [11], SFQ [6], CoDel [7]) vs. LPCC algorithms (namely, LEDBAT [10], TCP-LP [5], and NICE [12]), keeping the same network parameters as in Sec. 1. All results are based on average of 10 runs. Shortly, our investigation confirms the negative interference: *while AQM fixes the bufferbloat, it destroys the relative priority among CC protocols.*

Fig. 2 reports the bufferbloat intensity $E[Q]/Q_{max}$ (outer ring) and aggregate TCP% breakdown (inner ring) for varying combinations of AQM policies and LPCC flavors, including DropTail as a comparison. As expected, AQM successfully solves the bufferbloat. Different LPCC flavors have little impact on the intensity of the bufferbloat, which is decided largely by AQM policies.

However, AQM induces the reprioritization of CC. Under DropTail, LPCC flavors all achieve the lower priority with respect to best effort TCP (TCP%>90%). With AQM instead, regardless of the specific LPCC mechanism, the aggregate TCP% drops 40% on average. This severe phenomenon shows that AQM totally jeopardizes LPCC's low priority mechanism: furthermore, this holds for all LPCC and AQM investigated.

4. CONSEQUENCES

We now discuss the implication of these findings. Considering that LPCC is presently used, a solution of this negative interference is necessary. Also, since only one best effort TCP could still cause bufferbloat, AQM is a needed piece of the solution. We argue that classification capabilities will be needed in AQM to account for flows *explicitly advertised lower* level of priority.

In our future work, we plan to investigate the generality of our findings with a wider simulations campaign Experiments on testbed and real Internet environment are also on schedule.

Acknowledgement

This work was carried out at LINCS `http://www.lincs.fr`, and funded by the FP7 mPlane project (grant agreement no. 318627).

5. REFERENCES

[1] B.Cohen. How has bittorrent as a protocol evolved over time. `http://www.quora.com/BitTorrent-protocol-company`.

[2] V. Cerf, V. Jacobson, N. Weaver, and J. Gettys. Bufferbloat: what's wrong with the internet? *Commun. ACM*, 55(2):40–47, 2012.

[3] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[4] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: Illuminating the edge network. In *ACM IMC*, pages 246–259, 2010.

[5] A. Kuzmanovic and E. Knightly. Tcp-lp: A distributed algorithm for low priority data transfer. In *IEEE INFOCOM*, volume 3, pages 1691–1701, 2003.

[6] P. McKenney. Stochastic fairness queueing. In *IEEE INFOCOM*, 1990.

[7] K. Nichols and V. Jacobson. Controlling queue delay. *Commun. ACM*, 55(7):42–50, July 2012.

[8] R. Pan, B. Prabhakar, and K. Psounis. Choke - a stateless active queue management scheme for approximating fair bandwidth allocation. In *IEEE INFOCOM*, 2000.

[9] D. Rossi, C. Testa, S. Valenti, and L. Muscariello. LEDBAT: the new BitTorrent congestion control protocol. In *IEEE ICCCN*, 2010.

[10] S. Shalunov. Low Extra Delay Background Transport (LEDBAT). IETF Draft, Mar. 2010.

[11] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. In *ACM SIGCOMM*, 1995.

[12] A. Venkataramani, R. Kokku, and M. Dahlin. TCP Nice: A mechanism for background transfers. In *USENIX OSDI*, Boston, MA, Dec. 2002.