A Control-Theoretic Analysis of Low-Priority Congestion Control Reprioritization under AQM

A control theoretic tale of two species: reprioritization of low-priority congestion control under AQM

LUCA DE CICCO, Telecom SudParis, SAMOVAR UMR 5157, France
YIXI GONG, Telecom ParisTech, France
EMILIO LEONARDI, Politecnico di Torino, Italy
DARIO ROSSI, Telecom ParisTech, France

Recently, a negative interplay has been shown to arise when scheduling/AQM techniques and low-priority congestion control protocols are used together: namely, AQM resets the relative level of priority among congestion control protocols. This work explores this issue by carrying out a control theoretic analysis of the dynamical system to prove some fundamental properties that fully characterize the reprioritization phenomenon. In particular, (i) we provide the closed-form solution of the equilibrium in the open-loop (i.e., fixing a target loss probability p); (ii) we provide a stability analysis and a characterization of the reprioritization phenomenon when closing the loop with AQM (i.e., that dynamically adjusts the system loss probability). Our results are important as the characterization of the reprioritization phenomenon is not only quantitatively accurate for the specific protocols and AQM considered, but also qualitatively accurate for a broader range of congestion control protocol and AQM combinations. Finally, while we find a sufficient condition to avoid the reprioritization phenomenon, we also show at the same time, such conditions to be likely impractical: therefore, we propose a simple and practical system-level solution that is able to reinstate priorities among protocols.

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

Additional Key Words and Phrases: Bufferbloat, AQM, Scavenger protocol, Simulation

ACM Reference Format:

L. De Cicco, Y. Gong, E. Leonardi, D. Rossi, A control theoretic tale of two species: reprioritization of low-priority congestion control under AQM ACM ToMPECS 1, 1, Article 1 (April 2015), 25 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

The Internet is a rather complex ecosystem in which different species (i.e., hardware equipment), speaking a babel of languages (i.e., protocols) with numerous dialects (i.e., software implementations) coexists. Equipment vendors, normalization fora, and software developers let these species, languages and dialects exist and evolve. Of course, within each category a finer grained taxonomy is possible. For instance, taking the Internet "protocols" category, the layering principle allows to breakdown the protocol ecosystem into, according the Internet (or OSI) terminology, an application-level (or L7), a transport-level (L4), an Inter-networking (L3) and a host-to-network (L2) levels. Let us, for the sake of simplicity refer to engineers of the application and transport level protocols as the "upper-layer" engineer species, and to engineers of the network and host-to-network layers as "lower-layer" specimens.

Recent evolutions in the "upper-layer" have confirmed TCP to be still the largely most commonly adopted congestion control protocol over the Internet for the support of either elastic data transfer or transfer of data with weak real-time constraints such as in the case of video streaming. Nevertheless, in the last years an ensemble of application-specific Layer-7 congestion control algorithms, running over UDP, have been defined. With few exceptions[1] these protocols typically support application executed in background, such as P2P file transfers, file synchronization etc.,

[1]The most notable of which is represented by Google's QUIC, an application-layer protocol over UDP, targeted for TCP replacement in the context of SPDY/HTTP.

and are targeted to low-priority congestion control (LPCC) services, so that background applications do not subtract bandwidth from interactive applications.

Notable examples of LPCC protocols are represented by Microsoft Background Intelligent Transfer Service[2], TCP-LP [Kuzmanovic and Knightly 2003], TCP-NICE [Venkataramani et al. 2002] and BitTorrent low extra delay background transport (LEBDAT) [Shalunov et al. 2012], which is especially relevant as BitTorrent is still credited as primary contributor for uplink bandwidth [Sandvine 2014]. Differently from standard TCP, which reduces the source sending rate only in occurrence of packet loss events, LPCC congestion control schemes decrease the source sending rate as soon as the estimated packet delay grows beyond a given target. As a result of the previous design choices, the LPCC schemes exhibit a less aggressive profile than TCP when they compete for the bottleneck bandwidth, which is typically placed at the access-link (especially in upstream, in today scenarios), thus leave to the TCP flows most of the bottleneck bandwidth.

However, this is true only when TCP flows are able to fill the bottleneck buffer inducing a significant increase of the packet delay, i.e. when buffers are dimensioned sufficiently large and DropTail packet discarding policies are adopted. With this regard we recall that Internet buffers are traditionally dimensioned sufficiently large to store up 250-500 msec of packets at the nominal line-speed of the link. This simple rule-of-thumb approach however may be cause of the "bufferbloat" phenomenon, i.e., the unexpected increase of the packet round-trip time that may reach *several second*, in which cases the real access bandwidth is significantly smaller than the nominal bandwidth under which the buffer has been designed (this may happen over heavily congested 3G/4G[Jiang et al. 2012] and WiFi networks[Nichols and Jacobson 2012], or low-quality ADSL/Cable lines[Kreibich et al. 2010]).

To cope with bufferbloat, recent evolutions of the "lower-layer" have been specifically designed, and especially have started being deployed in both EU and US: indeed, infrastructural solutions to the bufferbloat such as scheduling (SFQ [McKenney 1990], DRR [Shreedhar and Varghese 1995]) and Active Queue Management (RED [Floyd and Jacobson 1993], CoDel [Nichols and Jacobson 2012]) techniques, whose adoption has been so far limited, are now becoming commonplace for operators worldwide to improve the quality of experience of their ADSL/Cable lines customers (e.g., the French ISP Free employs SFQ since 2005 [Bizon 2005] and US follows suit with CoDel under active development in DOCSIS modems [White 2014]).

While these trends on LPCC and AQM deployment are known facts, the impact on the dynamics of upper-layer LPCC schemes and lower-layer changes such as the replacement of simple DropTail buffer management policies with smarter adaptive policies is, however, not completely clear. Our previous simulative and experimental work [Gong et al. 2013b; Gong et al. 2014] has shown that a reprioritization effect may occur with LPCC flows getting a bandwidth share comparable with the TCP share. The goal of this paper is to shed light on this issue, by providing a systematic analysis from a control theoretic perspective of the dynamics of TCP and LEBDAT (taken as representative of the whole class of LPCC schemes) sharing an access bottleneck whose buffer adopts an AQM-RED scheme (taken as representative of the whole class of adaptive buffer management schemes).

The main contributions of this paper are as follow.

- We model the system as an Delay Differential Equation (DDE) that we study both as an open loop (Sec. 3) and as a closed loop (Sec. 4) system, significantly extending our preliminary work [Gong et al. 2013a].
- Using the open-loop model, where we fix an external loss probability p as if imposed by a generic AQM, we fully characterize the reprioritization regions (Sec. 4.1), and show that depending on how the parameters of LEBDAT and AQM-RED schemes are set, different behaviors are possible. Specifically, we derive a simple sufficient condition to avoid the reprioritization phenomenon; unfortunately, we show these settings to be highly impractical in real network scenarios.
- Using the closed-loop model, where we use RED to drive the queue size to a specific value, we then numerically characterize the stability of the linearized system around the equilibrium (Sec. 4.2). Specifically, after deriving

[2] `https://msdn.microsoft.com/en-us/library/aa363167.aspx`

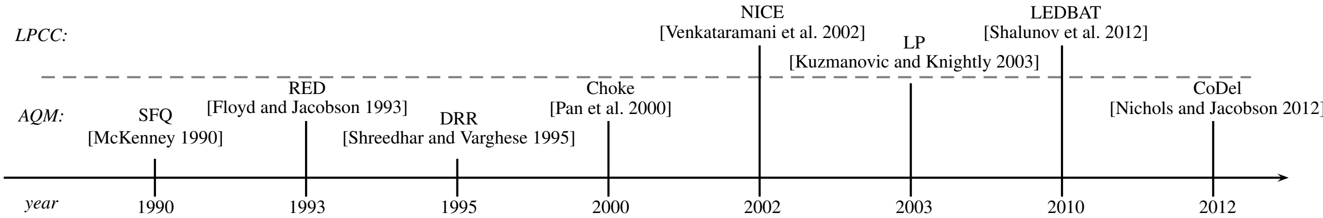

Fig. 1. Time-line of Active Queue Management (AQM) and Low Priority Congestion Control (LPCC) algorithms.

the boundaries of the stability region, we show that in the unstable region the system exhibits Hopf supercritical bifurcation [Hassard et al. 1981], i.e., an oscillatory behavior around an equilibrium is displayed.

—Using packet-level simulation (Sec. 5) we validate results of our model: we not only show a very good qualitative agreement but also introduce model refinements that reinforce the quantitative agreement, at the price of a treatable complexity (i.e., introducing a non-linearity by capping the minimum amount of LEDBAT packets in a RTT to 1, which happens in practice to avoid indefinite starvation).

—Finally, we discuss (Sec. 6) a very simple yet effective way to avoid such reprioritization phenomenon, which would already be deployable in practice, without requiring any further development but a minimum amount of explicit information exchange between the upper and lower layers.

Philosophically, our work shows that the independent evolution of upper and lower engineering species is a potential source of problems – namely the just mentioned reprioritization effect. We additionally demonstrate that, while operational points not requiring an interaction of upper- and lower-layer engineers to avoid such problems do exist, they are however of scarce practical interest. We therefore argue that a dialogue between species is needed for a clean solution. We are certainly not the first in pointing out problems rooted in layering [Crowcroft et al. 1992], yet we point out that the solution does not mandate complex cross-layer solution. As the reprioritization can be solved with the exchange of a minimum amount of information (i.e., a single bit, carried in lower-layer packet headers and set upon request of the upper layers) so that the evolution of both species can continue unperturbed.

Overall, the problem we address is not only timely (in reason of the above trends) and broadly applicable (as the effect we outline here holds for several combination of specimens), but our findings are fairly general (given to both the rigor in our methodology and the extent of the reprioritization) and accurate (as per validation with packet level simulation), and our solution is of practical interest (due to its extreme simplicity and the lack of cross-fertilization between species).

2. BACKGROUND

2.1 Related work

It would be extremely cumbersome to comprehensively retrace over 20 years of Internet research in these few pages. An historical viewpoint is sketched in [Gettys and Nichols 2012]: we extend this viewpoint by reporting in Fig. 1 a timeline of research in scheduling/AQM algorithms and LPCC protocols. The time-line clearly shows a temporal separation of these two research topics, which in our opinion helps understand why the AQM vs. LPCC interaction assessed in this paper was only barely previously exposed. In this section, we overview the related work separately considering (i) LEDBAT and other low priority protocols, (ii) the AQM vs. LPCC interaction, (iii) fluid modeling of TCP and AQM.

LPCC protocols. Protocols such as NICE [Venkataramani et al. 2002], LP [Kuzmanovic and Knightly 2003], 4CP [Liu et al. 2006] and [Key et al. 2004] share the same low-priority spirit of LEDBAT. We carried out a simulation-based comparison of NICE, LP and LEDBAT in [Carofiglio et al. 2010], showing that LEDBAT has the lowest level of priority. Some important differences among the above protocols are worth stressing. NICE [Venkataramani et al. 2002] extends the delay-based behavior of TCP Vegas with a multiplicative decrease reaction to early congestion (detected when the number of packets experiencing a large delay in an RTT exceeds a given threshold). Differently from LEDBAT, that reacts to instantaneous one-way delay (OWD) variations, NICE instead reacts to RTT variations, thus possibly reducing the sending window due to growing delay in the reverse path similar to TCP Vegas [Grieco and Mascolo 2004]. LP [Kuzmanovic and Knightly 2003] modifies the loss-based behavior of NewReno with an early congestion detection based on the distance of the OWD from a weighted moving average calculated on all observations. In case of congestion, the protocol halves the rate and enters an inference phase, during which, if further congestion is detected, the congestion window is set to one and normal NewReno behavior is restarted. This differs from LEDBAT, which aims at explicitly bounding the maximum delay introduced in the bottleneck queue, which is particularly important for VoIP, video conferencing, gaming and all other interactive delay-sensitive applications. Finally, a simulation based analysis of the impact of LEBDAT parameters on its behavior has been recently carried out in [Trang et al. 2014]. [Trang et al. 2014] focuses on a DropTail bottleneck-link shared by LEBDAT and TCP New Reno flows, proposing a set of LEBDAT parameters that minimize the overall LEBDAT bandwidth share.

Interaction of AQM and LPCC. To the best of our knowledge, aside our previous experimental [Gong et al. 2013b] and simulation-based [Gong et al. 2014] work, only [Schneider et al. 2010] mentions AQM and a LPCC (namely, LEDBAT) in the same paper. In one of the tests, the authors experiment with a home gateway that implements some (non-specified) AQM policy other than DropTail. When LEDBAT and TCP are both marked in the same "background class", the "TCP upstream traffic achieves a higher throughput than the LEDBAT flows but significantly lower than" under DropTail [Schneider et al. 2010]. This is explicitly mentioned in the LEDBAT RFC, stating that under AQM it is possible that "LEDBAT reverts to standard TCP behavior, rather than yield to other TCP flows" [Shalunov et al. 2012]. In our previous work [Gong et al. 2013b; Gong et al. 2014], we further show that this behavior is general and can arise from the interaction of any scheduling/AQM discipline and LPCC protocol shown in Fig. 1, using a twofold methodology including `ns2` simulation and experiments from both controlled testbed and wild Internet. The present work differs from [Gong et al. 2013b; Gong et al. 2014] in both its depth and methodology: indeed, we adopt a narrower but profound scope, selecting LEDBAT and RED as representative examples of the LPCC and scheduling/AQM design space that we then analytically model.

Fluid modeling. Fluid models have been extensively used in the literature to investigate the dynamics of TCP flows [Misra et al. 2000; Hollot et al. 2001; Liu et al. 2003; Marsan et al. 2005; Michiels et al. 2006; Mascolo 1999; De Cicco et al. 2011]. While several lines of research do exist (e.g., [Mascolo 1999; De Cicco et al. 2011] model the network as a time-delay linear system and TCP congestion control algorithms as Smith predictor controllers whose aim is to track an exogenous reference signal which models the TCP congestion window), the mainstream approach, closer to ours, is to model TCP window dynamics by means of either Delay Differential Equations (DDE) or Partial Differential Equations (PDE), [Misra et al. 2000; Hollot et al. 2001; Liu et al. 2003; Marsan et al. 2005]. We point out that, since generally a single dominant TCP flavor is modeled [Misra et al. 2000; Hollot et al. 2001; Mascolo 1999] (optionally including unresponsive background traffic [Liu et al. 2003] or short-lived connections [Marsan et al. 2005]), the novelty in this context lies in the definition of a fluid model of *heterogeneous* responsive sources, notably including LEDBAT. As our main innovation is not on the technique per se, but on its application to the study of a particular problem, we resort to classic models for TCP [Liu et al. 2003] and RED [Misra et al. 2000], that we extend to incorporate novel popular protocols such as LEDBAT. While the model presented in this work builds on our previous work [Gong et al. 2013a], this work introduces a significant amount of new control-theoretic material: notably, in Sec. 3 we provide closed-form solution for the equilibrium in open-loop and in Sec. 4 we provide a stability analysis and a characterization of the reprioritization phenomenon when closing the loop with RED.

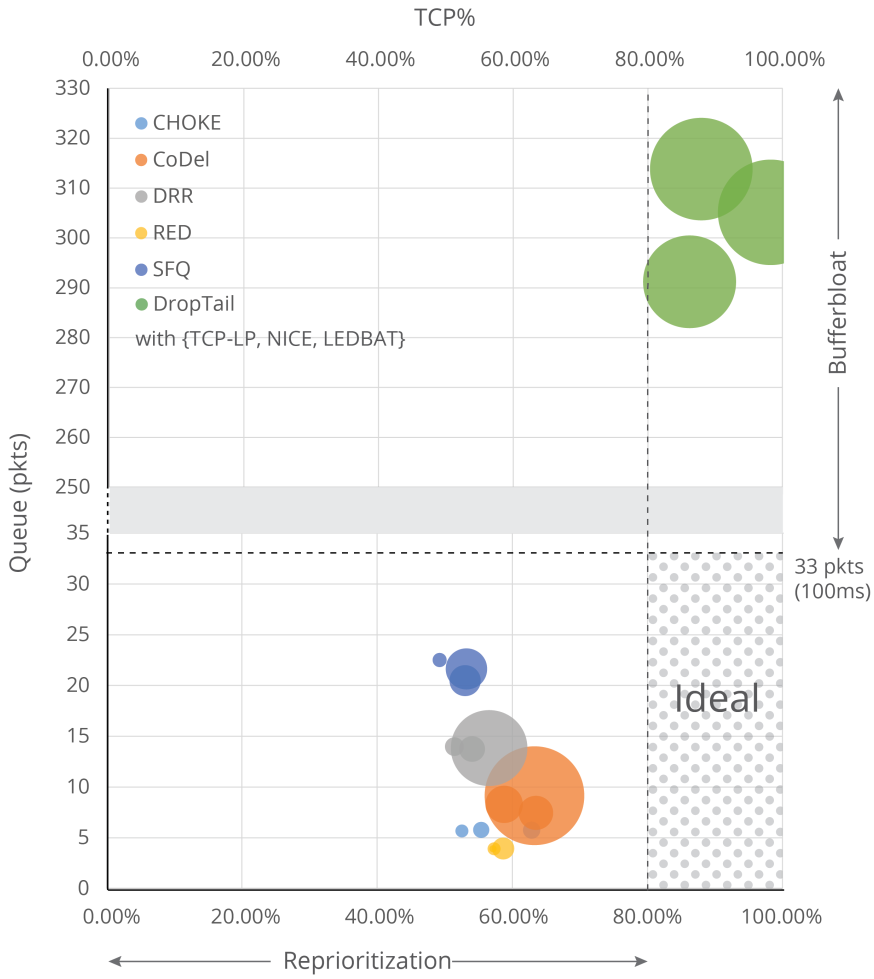

Fig. 2. Interaction of scheduling/AQM disciplines and LPCC protocols.

2.2 Motivations

We now better motivate the relevance of the present work. First, it could be objected that, given that the focus of this work is admittedly narrower than [Gong et al. 2013b; Gong et al. 2014], where we instead consider a larger basket of LPCC and AQM techniques, the value of the present work is similarly narrower. Yet, it is worth pointing out that the negative interplay of AQM and LPCC techniques is a general one, so that results gathered for any AQM+LPCC combination are fairly representative of all other AQM+LPCC pairs. To convince of this, we depict in Fig. 2 an original representation of simulations performed in [Gong et al. 2013b; Gong et al. 2014]: result are arranged as a bubble chart, where the center of the bubble represents the average (TCP share%, queue occupancy) for a specific AQM+LPCC pair, and the radius of the bubble represents the standard deviation over the different network parameters considered. It can be seen that, while under DropTail, TCP% monopolizes the bottleneck causing bufferbloat ("bufferbloat" zone in the y-axis), any combination of AQM+LPCC protocol results in a reduction of the queue length below the threshold

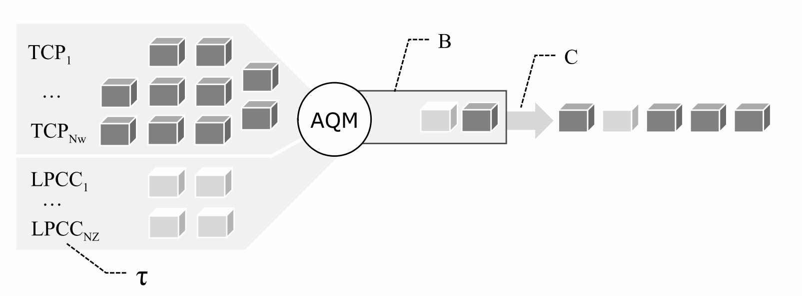

Fig. 3. Network scenario

considered to be harmful for interactive communications (100ms or 33 packets for the capacity considered in this example), but also on a dramatic low TCP share ("reprioritization" zone in the x-axis). It is worth stressing that whereas the actual values TCP share or queue size vary *quantitatively* across combination, reprioritization is a *qualitatively* general phenomenon. It is thus reasonable to develop a model for one such AQM+LPCC combination, to give both solid theoretic explanation of the observed phenomenon, as well as intuition and insights behind its root causes.

Second, we stress that this objective is far from being a sterile academic exercise, as mixtures of AQM+LPCC are already commonplace in the current Internet landscape. On the one hand, it has to be observed that LEDBAT is used by BitTorrent, the most prominent P2P applications: whereas downlink traffic is dominated by Video streaming applications and portals (e.g., Netflix and YouTube), BitTorrent is the top-1 application on uplink traffic, and can be credited for over one third of all upload traffic in North America, Latin America and Asia Pacific, as the latest report [Sandvine 2014] from the Canadian broadband management company Sandvine testifies. As claimed by Brahm Cohen, "LEDBAT is now the bulk of all BitTorrent traffic, [...] most consumer ISPs have seen the majority of their upload traffic switching to a UDP-based protocol" [B.Cohen], it is also confirmed by our own independent measurement in customer ISPs [Carofiglio et al. 2013]. On the other hand, it has also to be stressed that AQM adoption worldwide has started to change, with operators implementing scheduling policies in the upstream of the ADSL modem to improve the quality of user experience. For instance, in France, Free implements Stochastic Fair Queuing (SFQ) since 2005 [Bizon 2005] and Orange has started the deployment of Shortest Queue First (SQF) in 2010 [Benameur et al. 2013]). Similarly, US follow suit with new promising AQM techniques, such as CoDel [Nichols and Jacobson 2012], under active development in DOCSIS modems [White 2014].

Based on the above observations, we argue the RED+LEDBAT combination to be a reasonable modeling target. Indeed, though there is no agreement on a single specific AQM technique, due to the similar behavior of any AQM+LPCC combinations in Fig. 2, results will be still qualitatively relevant across combination. To simplify the analysis, we therefore decide to propose a new model for LEDBAT (which is by far the most used LPCC protocol in the Internet nowadays) and resort to known models of RED (which is by far the most popular AQM technique in the literature).

3. OPEN-LOOP MODEL

The goal of this section is to provide an open-loop model of the system composed of the LEDBAT and TCP sources that access a bottleneck whose queue is governed by a generic AQM controller.

In particular, the network scenario analyzed in this paper is shown in Fig. 3. N_W TCP long-lived flows and N_Z LEDBAT flows share a bottleneck with a fixed capacity C, whose access queue is of size B and implements a generic AQM scheme. In Sec. 3.1 we derive a mathematical model of the *open-loop* system, meaning that we consider the packet dropping probability $p(t)$ set by the AQM controller as given, i.e. it is assumed to be an input parameter of the model. This assumption is not fully realistic, since, in practice, packet loss probability $p(t)$ arises from queuing dynamics, which in turn are induced by the behavior of the sources. Nevertheless the analysis of this simplified system allows us to gather important preliminary information on the behavior of the system. In particular our study of the open-loop system focuses mainly on its equilibrium points and the analysis of their properties (see Sec. 3.2). Furthermore, we complement our analysis in section Sec. 3.3 by providing a clear physical interpretation of our findings. Finally, in Sec. 4 we will close the loop considering the effect of queuing dynamics on the loss probability for the specific case of RED.

3.1 The mathematical model

We denote the average TCP and LEDBAT window at time t as $W(t)$ and $Z(t)$ respectively. For TCP, we neglect the slow-start phase, which is instead only optional in LEDBAT. As such, we limitedly model the TCP congestion window dynamics in the congestion avoidance phase. At the reception of the $(n+1)$-th ack at time t_{n+1}, the TCP congestion window is updated as follows:

$$W(t_{n+1}) = \begin{cases} \frac{1}{2}W(t_n) & \text{on packet loss,} \\ W(t_n) + \frac{1}{W(t_n)} & \text{otherwise} \end{cases} \tag{1}$$

Similarly to TCP, LEDBAT reacts to losses by halving the congestion window, but it increases its congestion window at a rate that is proportional to the distance of the queuing delay $q(t)$ from the delay target τ and is always bounded by the TCP ramp-up in congestion avoidance:

$$Z(t_{n+1}) = \begin{cases} \frac{1}{2}Z(t_n) & \text{on packet loss,} \\ Z(t_n) + \frac{1}{Z(t_n)}\frac{\tau - q_d(t_n)}{\tau} & \text{otherwise} \end{cases} \tag{2}$$

with the current queuing delay $q_d(t)$ measured as:

$$q_d(t_n) = D(t_n) - D_{min} \tag{3}$$
$$D_{min} = \min_n D(t_n) \tag{4}$$

where $D(n)$ represents the instantaneous one-way delay (OWD) estimate, while the base delay D_{min} is the minimum observed OWD. The rationale is that, over a sufficiently large number of observations D_{min} accurately represents the fixed component of the delay (i.e., propagation delay plus negligible transmission delay, which should be the one found when queues are empty) so that the $D(t_n) - D_{min}$ difference represents the variable component of the delay (i.e., queuing delay plus negligible processing delay).

Notice that host synchronization over the Internet is known to be hard. As such, it is worth stressing that the OWD estimate $D(t_n)$ is affected by an unknown clock offset between the two endpoints, and is thus of no practical use. Conversely, the offset cancels in the difference operation in (3), which is only affected by clock drift – that is of much smaller magnitude and furthermore easier to correct [Cohen and Norberg 2010].

From (2), we gather that ramp-up is as fast as TCP only whenever the queue is empty (3), i.e., $\lim_{q_d \to 0} \frac{\tau - q_d}{\tau} = 1$. Furthermore, whenever the queuing delay hits the target τ, the congestion window settles since – when $q_d = \tau$ holds – it results $Z(t_{n+1}) = Z(t_n)$ for all n.

To analyze the interactions between sources and queue dynamics, we adopt a fluid flow modelling approach [Misra et al. 2000; Hollot et al. 2001; Liu et al. 2003; Marsan et al. 2005] in which the average dynamics of both sources and queues are described by nonlinear time-delay differential equations. In the following we denote by $W_i(t)$ the instantaneous congestion window at time t for connection i in the fluid system, by $R_i(t)$ the Round Trip Time (RTT) and by $p(t)$ the packet dropping probability set by the AQM algorithm. We consider the case of N_W TCP and N_Z LEDBAT connections sharing the same bottleneck, where flow-level congestion window evolution the TCP case is adopted from [Misra et al. 2000]:

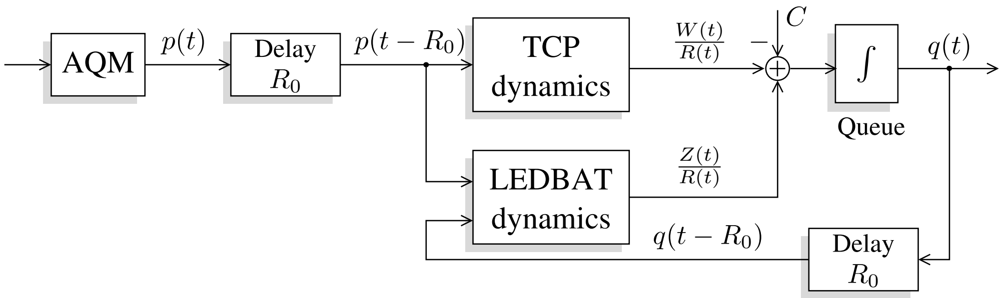

Fig. 4. Block diagram of the proposed model.

$$
\begin{aligned}
\frac{dW_i(t)}{dt} &= \frac{1}{R_i(t)} - \frac{W_i(t)W(t-R_i(t))}{2R_i(t-R_i(t))}p(t-R_i(t)) && (5)\\
\frac{dZ_i(t)}{dt} &= \frac{\tau - q(t-R_i(t))/C}{\tau}\frac{1}{R_i(t)} - \frac{Z_i(t)Z(t-R_i(t))}{2R_i(t-R_i(t))}p(t-R_i(t)) && (6)\\
\frac{dq(t)}{dt} &= \sum_{i=1}^{N_W}\frac{W_i(t)}{R_i(t)} + \sum_{i=1}^{N_Z}\frac{Z_i(t)}{R_i(t)} - C\mathbb{1}_{q(t)\geq 0} && (7)
\end{aligned}
$$

where the queuing delay is modeled as $q_d(t) = q(t)/C$ as suggested in [Hollot et al. 2001]. The RTT $R_i(t)$ is the superposition of two components: one constant component, i.e. the propagation delay T_i, and one time-varying, i.e. the queuing delay $q_d(t)$, thus giving $R_i(t) = T_i + q(t)/C$. For this reason the RTT cannot be considered to be constant since the component due to queuing delay can be predominant over the propagation delay – which is especially true in case of bufferbloat due to FIFO buffering. Conversely, in case AQM is used to avoid bufferbloat, it could be reasonable to assume the reverse $q(t)/C \ll T_i$ to hold.

We conclude this section by analyzing Fig. 4 that provides a representation of the dynamical system (5)-(6)-(7) in the form of a block diagram. The figure clearly shows that two control components interact to form the overall control system: (i) end-to-end control algorithms, namely TCP and LEDBAT, and (ii) the AQM controller.

The AQM control component sets a packet drop probability $p(t)$ that is first delayed and then fed in parallel to the TCP and LEDBAT blocks. Thus, we can argue that this control action acts equivalently on both the dynamics of the TCP and LEDBAT flows.

Concerning the end-to-end control algorithms it is easy to check that the only difference stems from the fact that LEDBAT dynamics is explicitly affected by the instantaneous queue length. With this regard it is interesting to notice that, from a control theoretic point of view, the term $-q(t-R_i(t))/(\tau C)$ in (6) plays the role of a "stabilizing" control signal since its sign is always negative or zero ($q(t) \geq 0$). Moreover, it is intuitive to anticipate that the value of the constant positive gain $1/(\tau C)$ plays an important role: the larger this constant, the larger will be the effect of the control signal $-q(t-R_i(t))/(\tau C)$ on the dynamics of the LEDBAT flow. The next sections will give a theoretical ground to this statement and show some important properties of the system.

3.2 Equilibrium and properties

In this section we compute the equilibrium of the proposed mathematical model, which allows us to derive some fundamental properties of the system. For the sake of simplicity we consider a bottleneck shared by an equal number of TCP and LEDBAT flows, i.e. $N_W = N_Z = N$. Moreover, we consider a homogeneous case, in which all the connections exhibit the same RTT (i.e., for both LEDBAT and TCP sources $T_i = T\ \forall i$). It is also worth noticing that the results presented below are valid regardless of the AQM control algorithm.

PROPOSITION 3.1. *The unique equilibrium (w, z, q) of (5)-(6)-(7) when $N = N_W = N_Z$, in the case of a generic AQM algorithm setting a steady state packet drop probability $p \in [0, 1]$, is given by:*

$$w = \sqrt{\frac{2}{p}} \tag{8}$$

$$z = \begin{cases} w & \sqrt{2} < w \le \frac{C}{2N}T \\ \frac{w}{2}\left[\sqrt{\left(\frac{Nw}{\tau C} - 2\right)^2 + 4\frac{T}{\tau}} - \frac{Nw}{\tau C}\right] & \frac{C}{2N}T < w \le (\tau + T)\frac{C}{N} \\ 0 & w > (\tau + T)\frac{C}{N} \end{cases} \tag{9}$$

$$q = \begin{cases} 0 & \sqrt{2} < w \le \frac{C}{2N}T \\ \tau C\left[1 - \frac{1}{4}\left(\sqrt{\left(\frac{Nw}{\tau C} - 2\right)^2 + 4\frac{T}{\tau}} - \frac{Nw}{\tau C}\right)^2\right] & \frac{C}{2N}T < w \le (\tau + T)\frac{C}{N} \\ Nw - TC & w > (\tau + T)\frac{C}{N} \end{cases} \tag{10}$$

The proof is reported in appendix.

Remark 3.2. At steady state w, z, and q vary as a function of p according to the three different operating zones[3] that are shown in Fig. 5:

(1) *No reprioritization* $0 \le p < 2N^2/(C^2(\tau + T)^2)$: in this zone $z = 0$ and thus the dynamics of N TCP flows accessing a bottleneck described in [Hollot et al. 2001] is recovered.
(2) *Reprioritization* $2N^2/(C^2(\tau + T)^2) \le p < 8N^2/(CT)^2$: in this zone $0 < z < w$ and $q > 0$, meaning that reprioritization is occurring since the LEDBAT flows are getting a non-negligible bandwidth share;
(3) *Underutilization* $8N^2/(CT)^2 \le p \le 1$: in this zone $w = z = \sqrt{2/p}$ and $q = 0$ and in this case the link is not fully utilized since $\sum w/R + \sum z/R = 2Nw/R < C$;

PROPOSITION 3.3. *A sufficient condition to avoid reprioritization in the case an equal number N of TCP and LEDBAT flows access a bottleneck link of capacity C, round-trip propagation delay T, with a queue governed by a generic AQM algorithm is that:*

$$\frac{C}{N}\frac{T + \tau}{\sqrt{2}} < 1. \tag{11}$$

PROOF. To prove this proposition we need to show that for any $p \in [0, 1]$ set by a generic AQM algorithm the LEDBAT congestion window z is zero at the equilibrium. From Remark 3.2 we have that $z = 0$, i.e. no reprioritization occurs, when the following condition holds:

$$0 < p < \frac{2N^2}{(\tau + T)^2 C^2}. \tag{12}$$

To conclude the proof we observe that when (11) holds the right extreme $2N^2/((\tau + T)^2 C^2)$ of (12) is always greater than 1, meaning that the only possible equilibrium is $z = 0$ regardless of the loss probability p set by the employed AQM control algorithm. □

Remark 3.4. Since (11) is a sufficient condition, when (11) does not hold, reprioritization could be still avoided by a particular AQM control law that sets a steady-state p such that (12) holds.

[3] In the following we consider that $CT/(2N) > \sqrt{2}$ otherwise the first zone would collapse to a point.

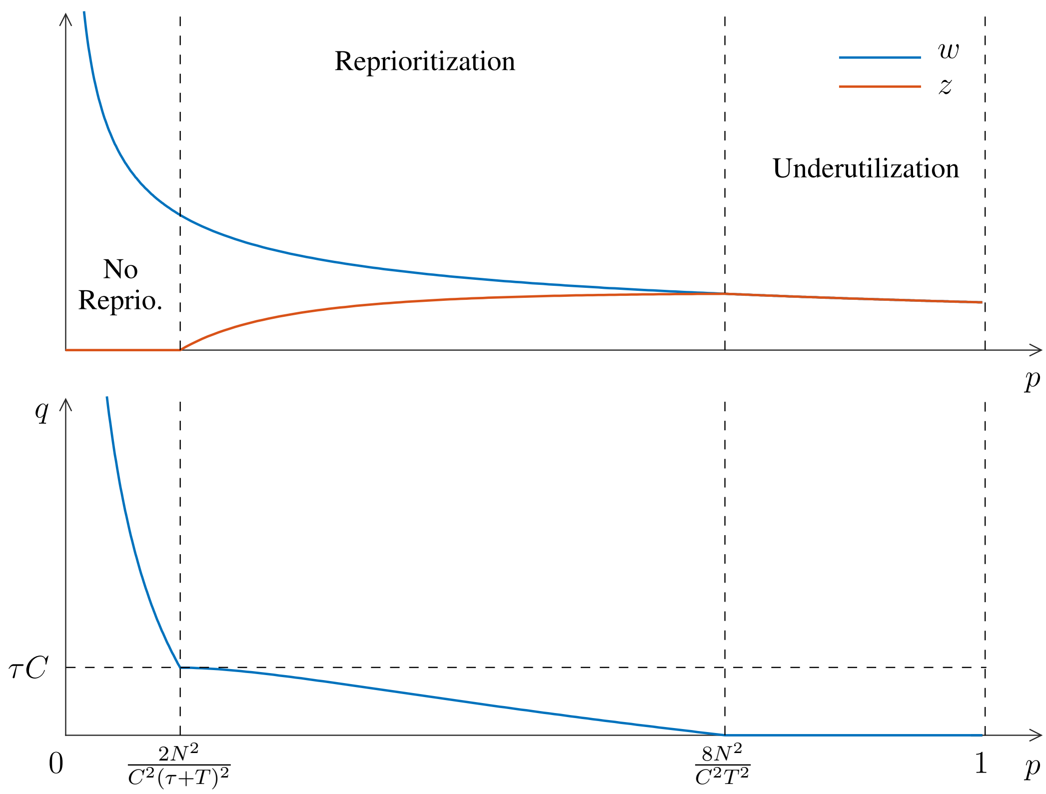

Fig. 5. LEDBAT window size z, TCP window size w, and queue size q at the equilibrium as function of $p \in [0, 1]$.

3.3 Discussion

In this section we analyze the tension that exists between the two main components of the overall system: (i) *end-to-end* control algorithms, implemented at the transport or application layer in the hosts; (ii) *AQM* controllers which are executed in the routers, i.e. in the network. Ideally one could aim at independently tuning the two control algorithms, while obtaining the following two properties at steady-state: (i) the LPCC flows only get a negligible share of the bottleneck when competing with TCP flows (reprioritization is avoided); (ii) the queuing delays are minimized (bufferbloat is avoided).

Thus, in this section we investigate whether, by only tuning the LEDBAT delay target τ, it is possible to satisfy the two properties mentioned above. Towards this end we consider Proposition 3.3 which provides a sufficient condition to avoid reprioritization. Thus, if we can show that condition (11) is general enough in practical scenarios and, when it holds, is able to also avoid bufferbloat, we would have satisfied both the properties by only tuning τ.

With this purpose in mind we rewrite (11) as follows:

$$0 < \tau < \sqrt{2}\frac{N}{C} - T \tag{13}$$

and we observe that if the inequality (13) is verified for some positive τ then reprioritization is avoided regardless of the specific AQM employed. Of course, satisfying (13) would need the a-priori knowledge, or a worst case estimate, of the number of concurrent TCP and LEDBAT flows N, the capacity of the bottleneck C, and the round-trip propagation delay of connections T. By considering a typical example we show that using (13) might be unfeasible because it could require setting a delay target τ that is either negative or too small.

To begin with, notice that for a typical ADSL whose upload data rate is 500Kbps, the transmission of a full MTU packet takes about 24 ms: initial versions of LEDBAT used to set τ = 25 ms, i.e., a packet worth of queuing. However, due to practical limitations (including timestamp precision in Windows OS, clock drift of several ppm in off-the-shelf PCs, etc.) this setting did not allow to fully exploit the link capacity, reason why the target was later increased to τ =100 ms. Thus, even though in principle one should set τ as low as possible, for practical issues τ has to be set to a value that is lower bounded at least by τ_{min} =25 ms (for which we already know that underutilization of the bottleneck occurs).

Therefore, if we require now that $\tau > \tau_{min}$, for (13) to be feasible we have to satisfy the following condition:

$$N > \frac{C}{\sqrt{2}}(T + \tau_{min}). \tag{14}$$

Now, we turn our attention to the case in whoch an ADSL whose uplink data-rate is 1 Mbps, which corresponds to $C = 100$pkt/s for packets of fixed size $P = 1250$B, is shared by flows whose round-trip propagation delay T is 50ms. In this case (14) would require at least $N = N_Z = N_W = 6$, i.e. a total of 12 flows sharing the bottleneck, a condition which might not hold in general. From this arguments we can say that it is in general not possible to tune τ independently from the AQM control law in order to both avoid reprioritization and bufferbloat.

We now take a different point of view and we analyze the formula of the TCP bandwidth share ρ at steady-state that is able to both quantitatively explain the interplay between AQM and LEDBAT algorithms and the reprioritization issue shown in Fig. 2. In particular we have:

$$\rho = \frac{w}{w+z} \tag{15}$$

By plugging the steady state w and z in the form of (24) we readily obtain:

$$\rho = \begin{cases} \frac{1}{1+\sqrt{1-\frac{q/C}{\tau}}} & 0 \leq q/C \leq \tau \\ 1 & q/C > \tau \end{cases} \tag{16}$$

It is straightforward to notice that $\rho \in [1/2, 1]$: $\rho = 1$ corresponds to the desirable case in which reprioritization is avoided and that is obtained when the steady state queuing delay q/C is greater than the LEDBAT target delay τ; $\rho = 1/2$ represents the non-desirable situation where LEDBAT and TCP get the same bandwidth share which is obtained when $q/C \to 0$. This discussion confirms what we have anticipated at the end of Sec. 3.1, i.e. that the control term $-q/(\tau C)$ that appears in the LEDBAT differential equation (6) is the main component driving the interplay between the two control algorithms acting on the system, i.e. AQM and end-to-end.

Interestingly, (16) is only function of the two parameters used by the AQM and the LEDBAT algorithms. The main parameter that can be used to tune LEDBAT is τ, whereas AQM algorithms trying to avoid bufferbloat strive to make the queuing delay $q_d = q/C$ small.

While a 100 ms target may be reasonable for an end-to-end protocol, such as in the case of LEDBAT [Shalunov et al. 2012], an AQM may be more precise in measuring the queue size and in adopting more aggressive dropping policy. This is in fact the case for two recently proposed AQM algorithms, namely CoDel and PIE: CoDel employs a target sojourn time of only 5 ms, whereas the default queuing delay target used by PIE is 20 ms. It is then reasonable to assume that in practical scenarios $q/C < \tau$ holds and consequently $\rho < 1$. This explains why the ideal interplay of AQM and LPCC that avoids bufferbloat and reprioritization shown in Fig. 2 is unfeasible.

Another important observation can be drawn by analyzing the absolute sensitivity of ρ with respect to τ or $q_d = q/C$, i.e. $\partial\rho/\partial q_d$ and $\partial\rho/\partial\tau$, in the region $q_d < \tau$ where reprioritization occurs:

$$\frac{\partial \rho}{\partial q_d}(\tau, q_d) = \frac{1}{2\tau(1+\sqrt{1-q_d/\tau})^2\sqrt{1-q_d/\tau}} \tag{17}$$

$$\frac{\partial \rho}{\partial \tau}(\tau, q_d) = \frac{q_d/\tau}{2\tau(1+\sqrt{1-q_d/\tau})^2\sqrt{1-q_d/\tau}} \tag{18}$$

Both the sensitivity functions show an asymptote at $q_d = \tau$ meaning that a sharp transition phase occurs when moving from the no-reprioritization ($q_d > \tau$) to the reprioritization zone ($q_d < \tau$) – yet another reason to make it impractical to rely on a single parameter selection τ to drive the overall system performance.

4. CLOSED-LOOP MODEL

In this section we focus on the properties of the closed-loop system that is obtained when a particular AQM controller, namely RED, is used. As already mentioned, RED has been chosen as a representative AQM algorithm mainly due to its popularity in the literature. It is worth noting that, even though the *quantitative* results derived in the following are specific to RED, the *qualitative* behavior of the system obtained by closing the loop with another AQM algorithm would be similar (see Fig. 2).

We start our analysis by deriving the model of the closed-loop system. To the purpose we take the model described by (5)-(6)-(7) and we plug the RED control law that sets the packet dropping probability $p(t)$ according to a static function $f : \mathbb{R}_+ \to [0, 1] \subset \mathbb{R}$ of the queue size $q(t)$ defined as follows:

$$f(q) = \begin{cases} 0 & q < q_{min} \\ \max_p \frac{q - q_{min}}{q_{max} - q_{min}} & q_{min} \leq q \leq q_{max} \\ 1 & q > q_{max} \end{cases} \tag{19}$$

where $\max_p \in [0, 1]$ is the maximum dropping probability when $q \in [q_{min}, q_{max}]$. In Sec. 3 we have shown that reprioritization is driven by two parameters, i.e. the steady state queuing delay q/C and the LEDBAT target τ (see (16)). For this reason we consider the system parameters space to be q_{min}, which relates to q/C, and τ. In particular, we analyze the closed-loop system and we explore the system parameters space to characterize: (i) the region of the system parameters where the reprioritization phenomenon is avoided, (ii) the stability region of the equilibrium.

4.1 Characterizing reprioritization

In order to characterize the reprioritization phenomenon in the parameters space, we have to compute the window size of the LEDBAT and the TCP flows at the equilibrium. Towards this end, we use Proposition 3.1 (Sec. 3) that characterizes the equilibrium of the open-loop system in closed form as function of the packet drop probability p set by a generic AQM controller.

We start by observing that the cases $q < q_{min}$ and $q > q_{max}$ are trivial and can be treated as follows. In the case $q < q_{min}$ RED would set a packet dropping probability equal to 0 that would make $z = 0$, i.e. LEDBAT is starved, and $w \to \infty$. However, since the congestion window w is always limited by the advertised window set by the flow control, in practice we would have $w = w_{max}$. The other limit case occurs when $q > q_{max}$ and RED would compute a packet dropping probability of 1, meaning that all packets are dropped, a situation of limited practical interest.

From the consideration made above, we limitedly focus on the case $q \in [q_{min}, q_{max}]$ where p is set proportional to the error $q - q_{min}$:

$$p(q) = k(q - q_{min}) \tag{20}$$

with the gain k defined as $k = \max_p / (q_{max} - q_{min})$. Thus, substitution of (20) in (10) yields the following equation has to be solved to get the steady-state queue length $q \in [q_{min}, q_{max}]$:

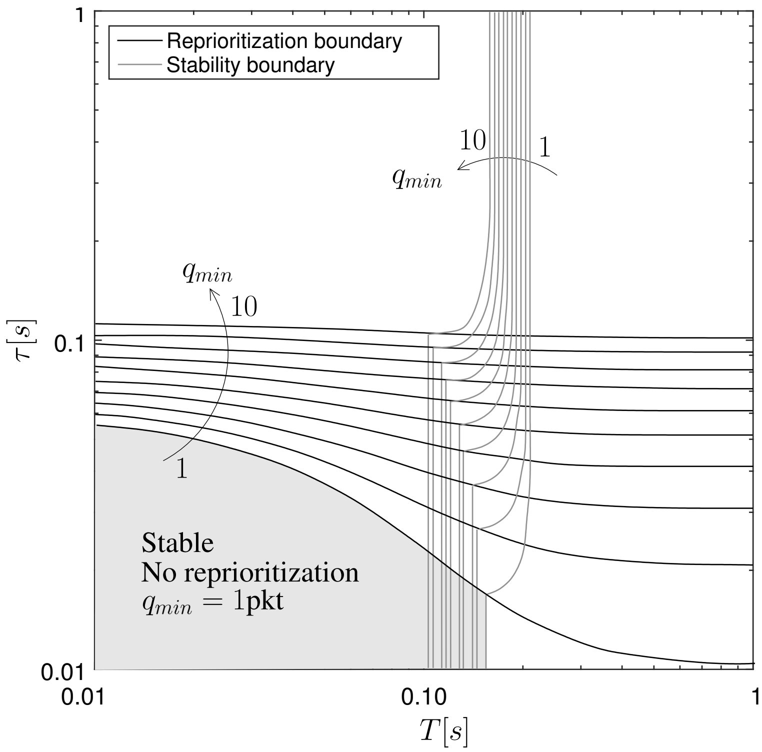

Fig. 6. Reprioritiazion and stability regions in the case of $N = 1$, $C = 1$Mbps, $q_{max} = 100$ pkt

$$\sqrt{1 - \frac{q}{\tau C}} = \frac{TC + q}{N}\sqrt{\frac{k}{2}(q - q_{min})} - 1 \tag{21}$$

Then, in order to get p we substitute the steady state queue length q solution of (21) in (19). Finally, plugging p in (8) and (9) we get the steady state window size of the TCP and LEDBAT flows respectively.
In order to characterize the local stability of the equilibrium we linearize the system around the equilibrium (w, z, q) to get the following linear time-delay system:

$$\dot{x}(t) = A_0 x(t) + A_R x(t - R) \tag{22}$$

where the state of the linearized system is $x(t) = [W(t) - w, Z(t) - z, q(t) - q]^T$, $R = T + q/C$ and the matrices A_0, and A_R are defined as follows:

$$A_0 = \begin{bmatrix} -\frac{\sqrt{p}}{\sqrt{2}R} & 0 & -\frac{1}{R^2C} \\ 0 & -\frac{\sqrt{p(1-\frac{q}{\tau C})}}{\sqrt{2}R} & -\frac{\tau+T}{\tau R^2 C} \\ \frac{N_W}{R} & \frac{N_Z}{R} & -\frac{N_W w + N_Z z}{R^2 C} \end{bmatrix}; A_R = \begin{bmatrix} -\frac{\sqrt{p}}{\sqrt{2}R} & 0 & \frac{1}{R^2C} - \frac{k}{pR} \\ 0 & -\frac{\sqrt{p(1-\frac{q}{\tau C})}}{\sqrt{2}R} & \frac{\tau - q/C}{\tau R^2 C} - k\frac{1-\frac{q}{\tau C}}{pR} \\ 0 & 0 & 0 \end{bmatrix} \tag{23}$$

It is well-known that the equilibrium of a non-linear time-delay system is locally stable if all the eigenvalues of the characteristic equation associated to (22) are in the left half-plane [Niculescu and Michiels 2007]. Since the equilibrium (w, z, q) cannot be found in closed form, we resort to numerically find the rightmost eigenvalue of (22) by using the

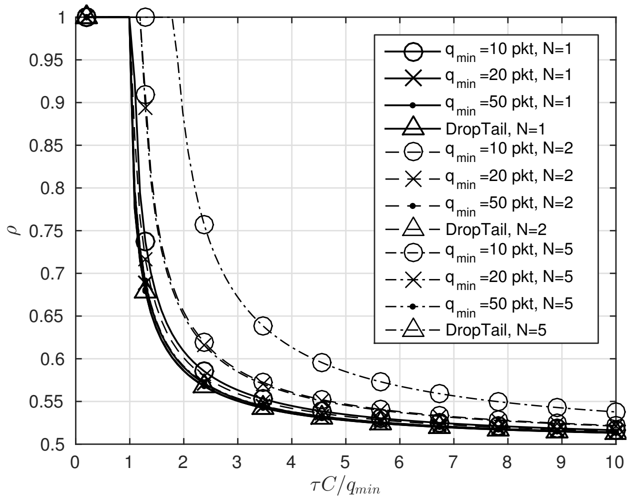

Fig. 7. TCP share ratio ρ at the equilibrium as a function of $\tau C/q_{min}$ for various flow number N, LEDBAT τ, and RED q_{min} settings.

tools described in [Michiels 2011].

Fig. 6 shows both reprioritization and stability regions in the parameters space (T, τ) in the case of a bottleneck with a capacity $C = 100$ pkt/s, propagation round trip time $T = 50$ ms maximum queue length $q_{max} = 100$ pkt which corresponds to a maximum queuing delay of 1 second. It is worth noting that such values are commonplace today, with modem buffers able to hold up to 4 seconds worth of data [Kreibich et al. 2010]. We have considered $q_{min} \in \{1, 2, \ldots, 10\}$ pkt to cover both AQM algorithms specifically designed to contain the queuing delay (CoDEL and PIE) and AQMs designed to contain queue length (f.i. RED, Choke [Pan et al. 2000]). Finally, for simplicity and without loss of generality, Fig. 6 shows the case of one LEDBAT with one concurrent TCP flow, i.e. $N = 1$.

Let us consider a pair (T^*, τ^*) and $q_{min} = q^*_{min}$: the equilibrium is stable if (T^*, τ^*) is at the left of stability boundary (dashed line) associated to q^*_{min}; similarly, it is simple to show that no reprioritization occurs, i.e. the LEDBAT bandwidth share at steady state is zero ($z = 0$) if (T^*, τ^*) is below the *reprioritization boundary* (solid line) associated to q^*_{min}. To give an example, Fig. 6 shows in gray the region where both stability of the equilibrium and reprioritization avoidance are obtained in the case of $q_{min} = 1$ pkt.

The figure shows that, as previously observed, the reprioritization phenomenon vanishes when $\tau < q/C$: in other words, when this condition holds all LEDBAT flows will by design yield to TCP and the system will behave as [Misra et al. 2000; Hollot et al. 2001]. At the same time, this scenario is unlikely to hold in practice. In fact, as previously observed, end-to-end congestion control protocols such as LEDBAT rely on noisy measures of queuing delay, so that they will not be able to guarantee protocol efficiency when $\tau \to 0$.

In the following we shall characterize reprioritization using a more fine grained approach as opposed the binary information that Fig. 6 conveys. In particular, we employ the TCP bandwidth share ρ given by (16) to measure the level of reprioritization, recalling that when ρ is close to 1 reprioritization is avoided.

We depict in Fig. 7 the TCP share ratio ρ at the equilibrium for varying user scenarios (i.e., number of TCP and LEDBAT flows N, LEDBAT target settings τ, and RED settings q_{min}).

Fig. 7 shows that under AQM the TCP share exhibits a sharp transition phase as soon as τ exceeds q_{min}, quickly dropping with an hyperbolic slope from a monopoly situation ($\rho \to 1$) to a fair share ($\rho^* \approx 0.58$ for $\tau = 2q$). Interestingly, [Carofiglio et al. 2010] shows that in the DropTail case, which can essentially be recovered from ours by letting $q_{min} \to q_{max} = B$, a sharp transition phase from TCP monopoly to a fair share happens whenever $\tau \to B$. This difference is rooted on the fact that RED dropping rates are strictly positive as soon as the queue size exceeds

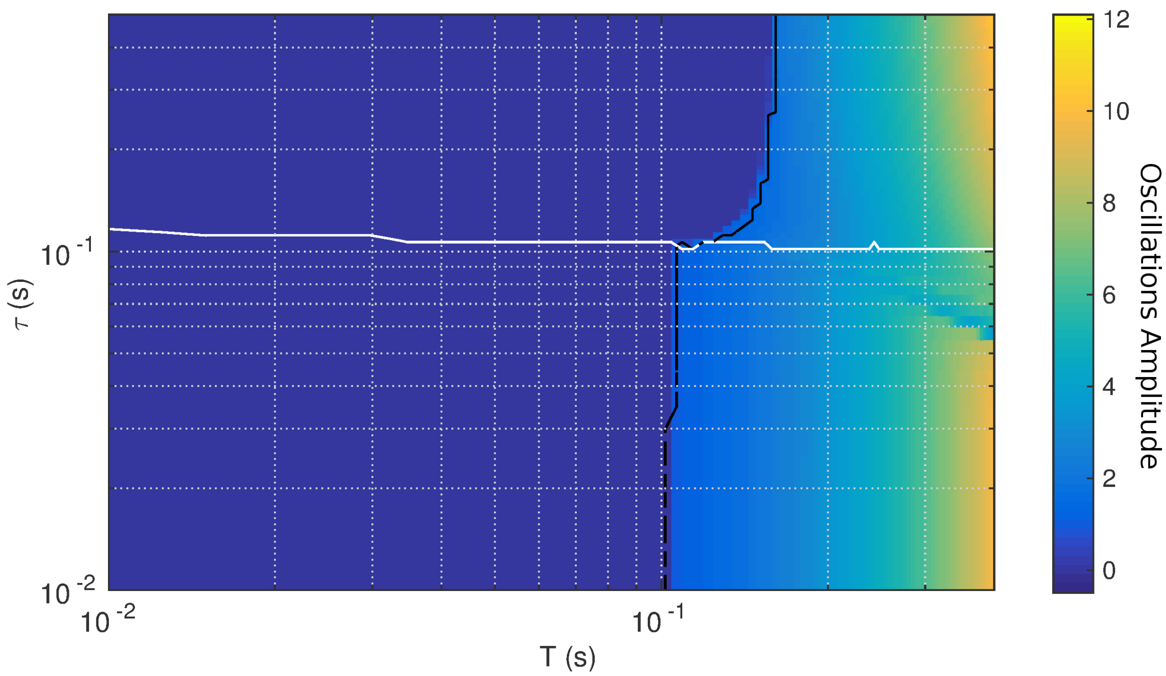

Fig. 8. Bifurcation analysis. $q_{min} = 10$ pkt, $q_{max} = 100$ pkt, $C = 1$Mbps, $N = 1$

q_{min}, whereas DropTail decisions have to wait until the queue exceeds B.

From Fig. 7 we also gather that different RED settings only minimally affect the reprioritization phenomenon. Trivially, since no dropping happens for $q < q_{min}$, this parameter plays the biggest role in determining the queue size at the equilibrium. Next comes the load factor, i.e., number of flows insisting on the bottleneck.

The impact of the LEDBAT target τ on the queue size has almost a step-like behavior, that can be explained taking into account the discussion about the absolute sensitivity of ρ with respect to τ (see (18)) or $q_d = q/C$ (see (17)) provided in Sec. 3.3.

4.2 Characterizing system dynamics

In this section we explore the parameter space with the aim of characterizing the dynamical behaviour of the system. To the purpose, let us consider again Fig. 6: the figure shows that the main parameter affecting the stability of the equilibrium is T, whereas the influence of q_{min} and τ is less important. Indeed, the adverse effect of time-delays in control loops is well-known and it affects a wide class of dynamical systems [Niculescu and Michiels 2007]. In particular, many time-delay systems display Hopf bifurcations when the time-delay T is varied: the linearized system is stable, i.e. all the eigenvalues are in the left-half plane, until T is equal to a critical value T_c for which a couple of purely imaginary eigenvalues appear; then, for $T > T_c$ the real-part of those eigenvalues increases. When a Hopf bifurcation takes place, a limit cycle, i.e. an isolated periodic orbit, branches from the fixed point and a self-sustained oscillatory dynamics is established for $T \geq T_c$ [Hassard et al. 1981].

Even though it is not the main focus of this work, in the following we shall characterize the properties of the dynamics of the system when T and τ are varied. In particular, we will show that, when the local stability of the equilibrium is lost, i.e. when passing from the stable to the unstable region shown in Fig. 6, a Hopf bifurcation occurs and the behaviour of the solutions at steady-state will change from being stationary to being periodic. With this purpose, we make T and τ vary and we numerically solve the time-delay differential equations to measure the amplitude of the

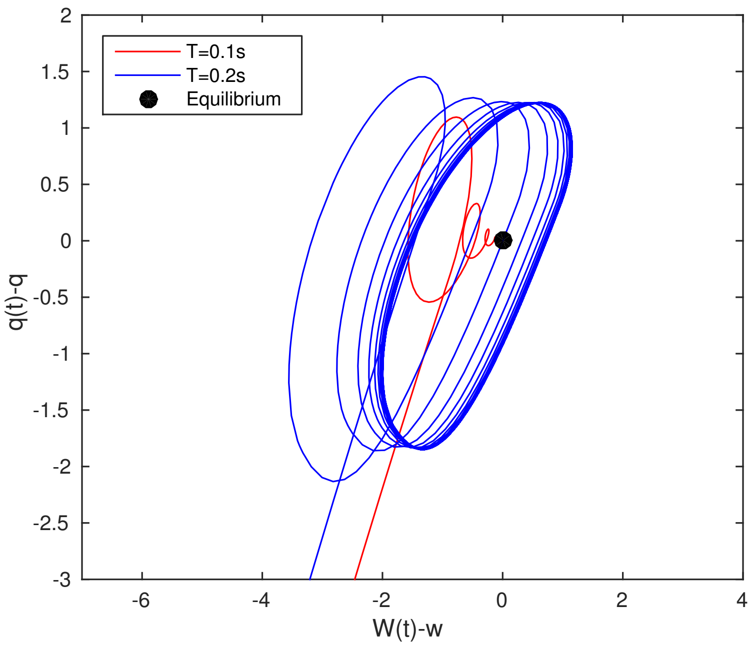

Fig. 9. Phase plots showing a stationary dynamics ($T = 0.1$ s) and a periodic dynamics ($T = 0.2$ s). $\tau = 0.2$ s, $q_{min} = 10$ pkt, $q_{max} = 100$ pkt, $C = 1$Mbps, $N = 1$

oscillations as a function of the two parameters. Fig. 8 shows a contour plot of the oscillation amplitude function of T and τ: as expected, in the stable region – the one at the left of the stability boundary show in black – the amplitude of the oscillations is zero, meaning that a stationary behaviour is obtained; as soon as we leave the stable region self-sustained oscillations appear whose amplitude gets larger and larger as we move away from the stable region. Fig. 9 shows the phase plots in the case of $T = 0.1$ s and $T = 0.2$ s when τ is fixed to 0.2 s: in the case of $T = 0.1$ s, since we are in the stable region, the equilibrium is reached asymptotically; on the other hand when $T = 0.2$ s, the trajectory describes a closed orbit and it does not converge to the equilibrium.

Let us now consider Fig. 10 that shows the dynamics of the system in each of the four regions shown in Fig. 6.

The two figures at the left are representative of the region where the system is stable and exhibits a stable response: the figure at the top-left corner shows the case in which reprioritization is obtained and the LEDBAT flow gets a non negligible share of the bottleneck bandwidth; on the other hand, the figure at the bottom-left corner shows the desirable case where the LEDBAT flow is starved by the TCP flow. The two figures at the right of Fig. 10 show the corresponding behaviour in the case the propagation delay T is larger than the critical value and, as expected considering the above analysis, a periodic response is obtained.

We conclude this section by showing the time-evolution of the system equations when $N_W = N_Z = 1$ in Fig. 11 under either DropTail (a) or RED (b,c,d) disciplines. Top plot shows the LEDBAT and TCP window congestion windows evolution, while queue $q(t)$ is reported in bottom plots.

In the DropTail case, we set $\tau = 0.1$ s and observe the same behavior shown via ns2 simulation in [Rossi et al. 2010]: i.e., LEDBAT yields to TCP as expected under DropTail. In the RED case, we set $q_{max} = B = 100$, $q_{min} = 10$ for the sake of illustration and let τ grow from 0.1 s (b) to 0.2 s (c) and 0.4 s (d). Notice that in case (b), RED drastically reduces the queue size and let the TCP rate to match the capacity after a short transient. Yet, when the target τ increases in (c) and (d), LEDBAT becomes increasingly aggressive under RED, and competes more fairly against TCP.

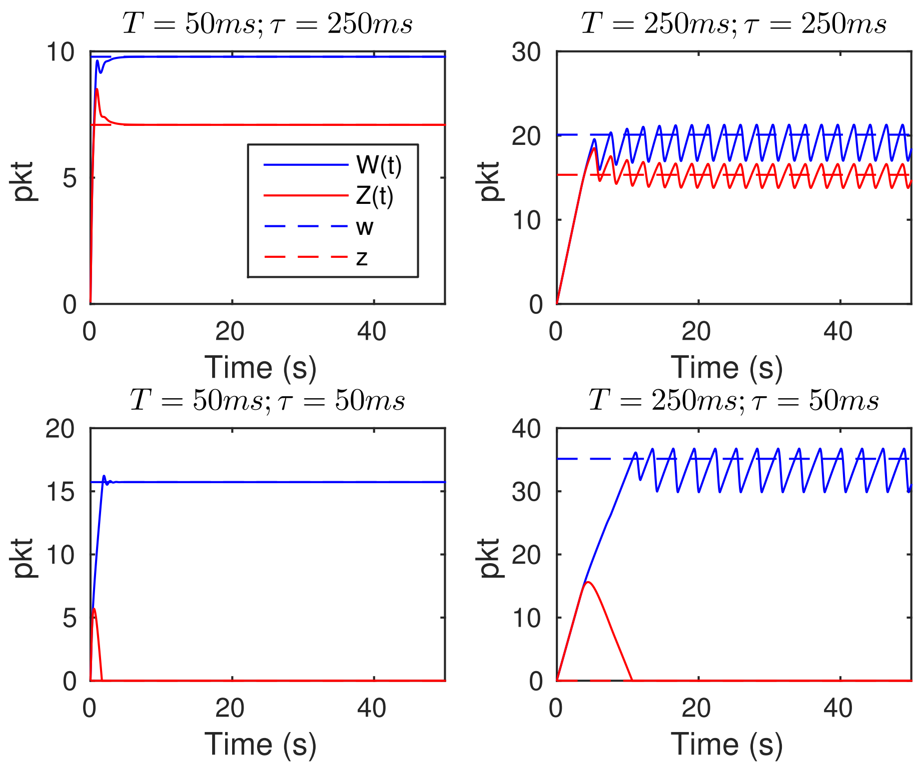

Fig. 10. LEDBAT and TCP dynamics window comparison in the four regions ($q_{min} = 10$pkt, $q_{max} = 100$pkt, $C = 1$Mbps, $N = 1$)

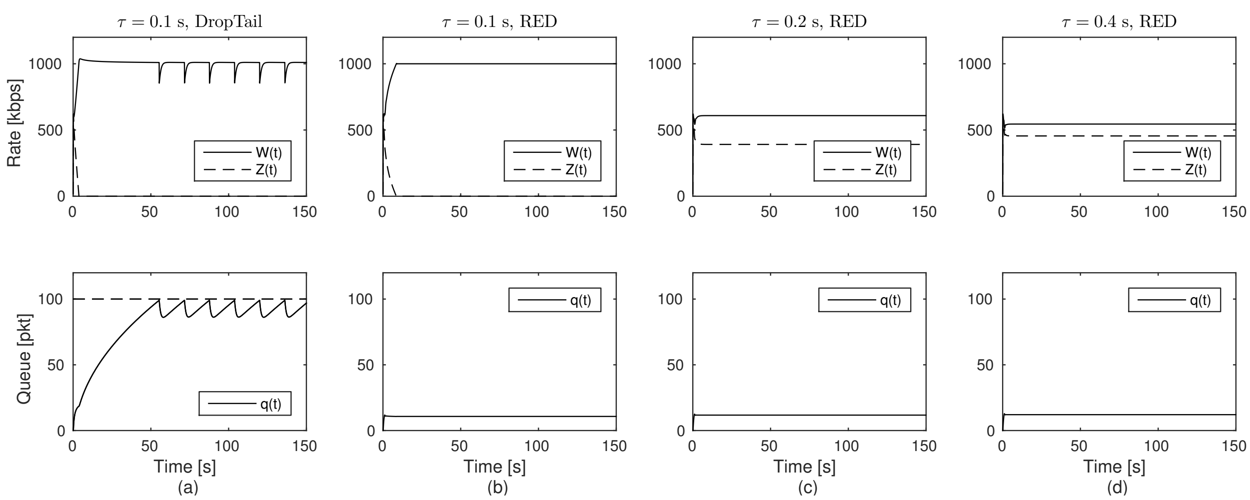

Fig. 11. Reprioritization phenomenon: Time evolution of $W(t)$, $Z(t)$ and $Q(t)$ under DropTail (a) and RED (b,c,d) for different values of τ.

To avoid cluttering the pictures, we do not report here the behavior of LEDBAT for increasing target τ under DropTail: from the simulation-based sensitivity analysis reported in [Carofiglio et al. 2010], it emerges that LEDBAT yields to TCP for a large range of $\tau < B/C$ values, and only whenever τ approaches (or exceeds) B/C LEDBAT behavior becomes loss-based as TCP.

5. VALIDATION

In this section we validate a subset of the numerical results obtained by integrating the time-delay differential equations modelling the system against those obtained from `ns2` simulator using our own implementation of LEDBAT, which is available as open source[4]. We additionally point out that we released all scripts and scenarios[5] to reproduce the simulation (as well as the experimental) results of [Gong et al. 2014]. In what follows, validation is performed on the most challenging (in terms of matching the simulation vs. fluid model results, to get conservative estimate of model accuracy) and relevant scenarios (in terms of practical relevance).

5.1 Scenario

We argue that the most challenging scenario, in terms of matching results gathered via simulation and fluid model, is the one with few number of flows. This is intuitive since in the case of multiple backlogged connections, statistical multiplexing will smooth out the impact of events, such as TCP retransmission time out, that would otherwise cause discontinuities in the case of few connections. At the same time, we also argue that the most practically relevant scenario is precisely one with a relatively small number of flows. Indeed, since the bottleneck is the user access, the number of concurrent connections will be likely small, even considering multiple applications/users in the household.

We consider both the Cloud and the P2P cases. In the Cloud case, it is easy to see that a small number of connections will be opened, at any given time, for a specific service. While considering a single user, even the server contacted will evolve over time (e.g., due to load balancing), this likely happens over time-scales that are much larger with respect to the short time-frames that we consider as "backlogged" data transfers (i.e., from tens of seconds to minutes) in this paper. Hence, the number of backlogged connections is upper-bounded by the number of Cloud services the user subscribes to, such as DropBox for data, GoogleMusic for music and Picasa for pictures/videos. Additionally, the number of simultaneous connections also depends on the on/off synchronization pattern toward the Cloud. As users are not continuously generating all kind of data at the same time, it thus reasonable to envision only a moderate number of concurrent backlogged connections per household, some of which may be lower priorities (e.g., pictures) over others (e.g., critical data, backup).

Consider next the P2P case, where it makes sense to consider file-sharing applications such as BitTorrent due to its popularity, and since it introduced LEDBAT in the first place precisely due to the bufferbloat problem. In BitTorrent, pipelining of piece requests at the application-level can cause multiple chunks to be transmitted consecutively over the same connection at transport-level. Since BitTorrent limits the number of concurrent slots to about[6] 4 per torrent, the number of concurrent connections will be again small. Moreover, BitTorrent peers periodically evaluate the throughput toward other peers every tens of seconds, and connections are maintained in case of good end-to-end throughput: coupled to pipelining, this entails that over the tens of seconds to minute timescale, connections can be considered backlogged.

5.2 Model validation against `ns2`simulations

From the above discussion, in the following we will limitedly consider an equal number $N = N_W = N_Z$ of flows, and let the total number of flows vary in $2N \in [2, 10]$ range. Unless otherwise stated, we consider homogeneous RTT delay settings with propagation delay $T = 50\,\text{ms}$ (to which we add a jittering component of 1 ms to avoid

[4]`http://www.enst.fr/~drossi/ledbat`
[5]`http://www.enst.fr/~drossi/dataset/ledbat+aqm`
[6]The limit actually increases with the square root of the uplink capacity

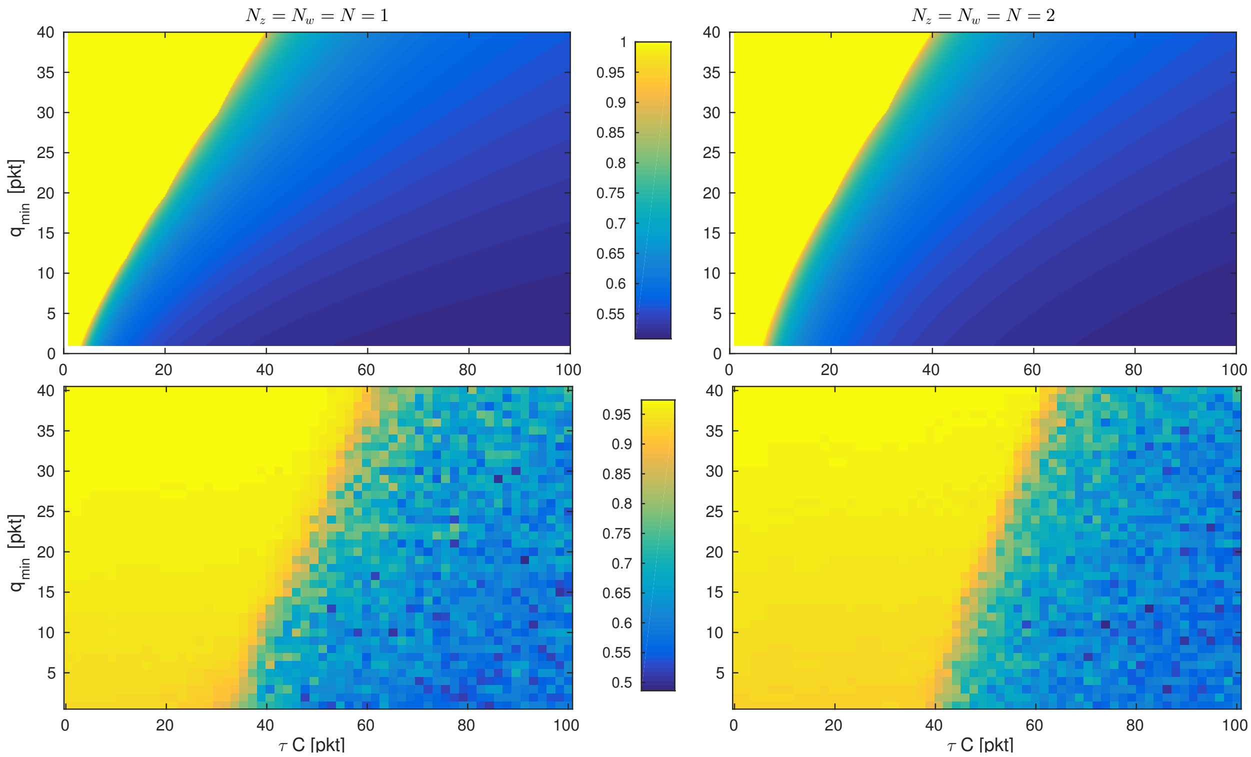

Fig. 12. Heatmap plots of $\rho(\tau, q_{min})$ in the case of either $N = 1$ (left) or $N = 2$ (right) obtained using the proposed model (top) or ns2 simulations (bottom)

synchronization of the congestion window dynamics). To precisely characterize system equilibrium properties, we will let the LEDBAT target τ vary – that in the uTorrent implementation of LEDBAT, this can be easily done by adjusting the `net.utp_target_delay` settings.

Without loss of generality, we consider a single access bottleneck and fix the capacity to C=1Mbps, typical range for ADSL/Cable access. The bottleneck buffer can accommodate up to $B = q_{max} = 100$ packets that, considering P=1250 Bytes sized packets for simplicity, corresponds to a maximum queuing delay of 1000 ms.

We point out that our goal is not to provide an exhaustive coverage of all scenarios considered in [Gong et al. 2014] (as scripts to reproduce experiments and simulations are available as pointed out before). Rather, our main aim here is to validate the most representative instance of our results – which is clearly represented by the TCP share ratio ρ that precisely quantifies the reprioritization.

As we have previously seen, q_{min} and τ have by far the biggest role in determining the TCP share curve, followed by the number N of flows in the bottleneck. Hence, in Fig. 12 we compare the TCP share ratio $\rho = w/(z+w)$ as a function of $\tau C \in [5, 100]$pkt and the RED minimum threshold $q_{mim} \in [5, 40]$pkt obtained either numerically (top figures) or via ns2 simulations (bottom figures) in the case of $N = 1$ (left) or $N = 2$ (right). The figure shows that the proposed model is able to qualitatively capture the reprioritization phenomenon that is observed with ns2 simulations (and experiments) across the considered parameter space $(\tau C, q_{min})$. In particular, by fixing $q_{min} = q^*_{min}$ and moving from the left to the right on the line parallel to the x-axis connecting $(0, q^*_{min})$ to $(\tau_{max} C, q^*_{min})$ an abrupt decrease of the TCP bandwidth share ρ occurs for any q^*_{min}. However, Fig. 12 also shows that the model overestimates the reprioritization: in fact, ns2 simulations provide a zone where $\rho \simeq 1$ (no reprioritization) that is larger than the one

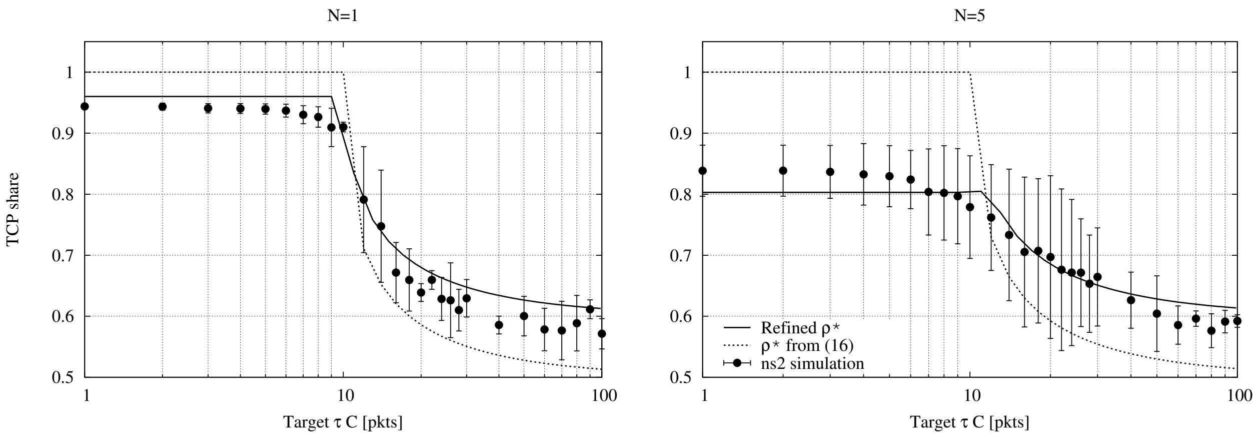

Fig. 13. Simulation validation of TCP share ratio ρ at steady-state as a function of τC for various traffic scenarios $N_W = N_Z = \{1, 5\}$.

obtained by using the model. We shall return shortly on this matter and propose a refinement of the model to address this issue. Finally, Fig. 12 shows that the effect of increasing the number N of concurrent flows on reprioritization is qualitatively the same, i.e. with an increased number of flow reprioritization is mitigated. This result also confirms the theoretical findings conveyed by Proposition 3.3.

5.3 Model refinement

We now return on the issue of the model underestimation of the TCP share that has been observed in Fig. 12 to provide a refined model. Towards this end we fix $q_{min} = 10$pkt, $q_{max} = B = 100$, $\max_p = 0.1$ and consider two traffic scenarios $N_W = N_Z = \{1, 5\}$. Fig. 13 compares average simulation results (solid point, with standard deviation bars over multiple runs) against the equilibrium given by (16) previously discussed (dotted line) and a slightly more accurate version (solid black line) that compensates two simplifications of the fluid model that we discuss next. Notice indeed that (16) captures reasonably well the essence of the reprioritization phenomenon. Still, two quantitative discrepancies arise.

First, it can be seen that for values of $\tau C > q_{min}$ the model underestimates the TCP share. This results from a known problem of the TCP model presented in [Liu et al. 2003] that this work extends: i.e., [Liu et al. 2003] is known to underestimate TCP congestion window with respect to simulation, which can be easily compensated by taking into account a multiplicative decrease factor of 1.5 (instead of 2) as in [Liu et al. 2003]. The refined equilibrium takes into account this correction, and is significantly more accurate when $\tau C > q_{min}$.

Second, recall that when $\tau C < q$, the model degenerates into a simpler one in which only TCP flows compete on the bottleneck, hence $\rho = 1$. In practice however, we know that LEDBAT will keep sending a minimum of 1 packet per RTT: this is done to continuously measure the queuing delay, at very low frequency and intrusiveness. LEDBAT does this in order to promptly react to queuing delay reduction and effectively utilize the spare capacity as soon as the link becomes free again. Hence, in case $\tau C < q$, our model of LEBDAT window dynamics can be, in principle, easily refined to account for this effect by introducing a non-linearity (i.e, capping from below window) at $z = 1$. As a result, the capacity available for TCP is reduced proportionally to the number of LEDBAT flows, i.e.,

$$\rho < 1 - \frac{N}{\frac{CTp}{P} + q}$$

Observe that, the new equilibrium point taking into account this second correction is significantly more accurate with respect to (16) when $\tau C < q_{min}$. However this is obtained at the price of having to handle the additional complexity brought by the additional non-linearity. Yet, we argue that such level of detail can be better captured with `ns2` simulations, and that quantifying the *exact* level of reprioritization is less relevant for practical purposes – i.e., as users will likely be interested in knowing whether their non-critical bulky transfers are indeed lower-priority with respect to critical continuous backups, or if they compete on a roughly equal basis.

6. DISCUSSION

Recent evolutions on the upper species (i.e., Internet applications) and lower species (i.e., Internet infrastructure) seem to suggest that AQM and Low priority congestion control (LPCC) protocols will have to coexist: indeed, popular applications are developing delay-based congestion control protocols such as BitTorrent/LEDBAT on the one hand, operators are starting to deploy AQM/scheduling on the user access uplink on the other hand. As such, it is imperative to find solutions to the negative AQM/LPCC interplay we have shown in this paper. While a general solution is hard to find, as testified by the current standpoint after over 20 years of research, a patch to this specific problem may be within reach.

Some might argue that small buffers would be enough to solve bufferbloat altogether. Yet, there are several reasons why this simple solution is not sufficient. First, in presence of too small buffers, it would be difficult for TCP and other congestion control to fully saturate the capacity, causing an undesirable efficiency loss. Second, deciding a buffer size is a matter of concern per se: consider indeed WiFi links, where the capacity may fluctuates widely over time, so that no single buffer size can at the same time (i) be large enough to support TCP congestion control and (ii) rule out bufferbloat in a fast-to-slow transition from 54Mbps to 2Mbps. Finally, jeopardization of relative priorities are not solved by small buffers as we show in [Gong et al. 2013b].

An ideal solution should achieve two goals: (i) meet quality of service constraints while (ii) respecting relative levels of priorities among protocols. Quality of service constraints clearly translate into upper-bounding the queuing delay, that we know is used by protocols to enforce their relative priorities. Since even a single TCP flow may bufferbloat the others, the solution *needs AQM*, as otherwise the quality of service constraints would be violated. At the same time, to avoid the LPCC reprioritization phenomenon, we argue that classification capabilities will be needed in AQM to account for flows' *explicitly advertised* level of priority.

Although in the more general case classification has failed to be adopted (IP TOS field, DiffServ, etc.), and the ability to claim *higher* priority could be easily gamed, in a hybrid AQM vs. LPCC world it makes sense for flows to claim a *lower* priority: indeed, generally malicious sources would tend to exploit resources by claiming high priority, so if the source is marking its own packets as low priority, for the scheduler there is no reason not to follow the application desire by prioritizing dropping of low-priority packets. We believe that this subtle difference can make an important practical difference in terms of deployability.

A simple way would be to let application exploit IP TOS field. While the overloading of the IP TOS can be troublesome within an operator network, this is not an issue in the home. Indeed, the usefulness of the IP TOS is not end-to-end but merely meant as a low-priority signal to the box in the user home, where the bottleneck and contention arise. Hence, IP TOS could be leveraged by the ISP Customer Premise Equipment (CPE) in the user home to apply differential treatment to best-effort and low-priority traffic (e.g., different AQM loss profiles, different scheduling weights), after which the end-user IP TOS value is no longer useful and can be rewritten by the CPE (or at the DSLAM, or BRAS, etc.) in the network of an operator using DiffServ if needed.

We further stress that the firmware governing home-routers and WiFi APs is generally based on some variants of the Linux kernel, possibly open-source as in the OpenWrt or CeroWrt cases. We point out that the above solution is therefore already implementable without any additional development effort – e.g., using strict priority queuing or shaping. In the Linux traffic control (`tc`) suite, this can be achieved with the `PRIO` queuing discipline (`qdisc`) that implements non-shaping container for a configurable number of classes which are dequeued in order. This first solution allows for easy prioritization of traffic, where lower classes are only able to send if higher ones have no packets available. A second solution offered by Linux `tc` is represented by the `CBQ qdisc` that offers shaping and finer-grained prioritization capabilities.

This solution is simple, as it requires a minimum amount of information exchange precisely at the interface between layers, with the upper-layers requiring a marking in the lower-layer packet header. This also means that evolution of species can still happen independently, without requiring heavy cross-layer or hybrid solution – which would require much more involved exchanges between species that are not necessarily akin. This simplicity is, we believe, the root of its appeal, as this also increases the chances of its deployment, making it of high practical relevance for the problem under consideration.

7. CONCLUSIONS

In this work we analyzed the interdependency phenomenon that occurs when heterogeneous protocols, namely best-effort and low priority congestion control (LPCC) protocols, share a bottleneck governed by an Active Queue Management (AQM) algorithm.
In previous works we have shown by means of simulations and Internet experiments [Gong et al. 2014] that a negative interplay exists: in particular, the relative level of priority of the congestion control protocols is reset, a phenomenon that we call *reprioritization*, meaning that the protocols compete on a roughly equal basis.
With the purpose of analyzing this phenomenon we have proposed a DDE model that captures the dynamics of an overall system composed of: (i) TCP flows, which are representative of best-effort congestion control algorithm; (ii) LEDBAT flows, which is a prominent example of LPCC algorithms; (iii) an AQM control algorithm that is executed at the bottleneck queue.
By analyzing dynamical properties of the system around its equilibrium points we have been able to provide an explanation to the reprioritization phenomenon and confirmed the generality of this issue. In fact, even though we provide a sufficient condition that allows to independently tune the LEDBAT parameter τ to avoid reprioritization regardless of the AQM employed, we show that such a condition is of scarce practical interest due to the difficulty of accurately measuring the delays at the end systems. Moreover, model predictions have been validated against `ns2` simulations when RED is used as AQM algorithm.

Due to the increasing deployment of both low-priority congestion control and AQM techniques that today are employed to fight bufferbloat, the problem discussed in this paper may be of significant practical relevance. As we believe that it may be desirable for end-users (or end-user applications) to autonomously and coarsely set their relative level of priorities, we have proposed simple yet effective system-level design and practices that give a solution to the issue that we have analyzed in this work.

Advertisement

This work was carried out at LINCS `http://www.lincs.fr`. The research leading to these results has received funding from the European Union under the FP7 Grant Agreement n. 318627 (Integrated Project "mPlane"). This research has been also partially supported by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 608533 (NECOMA). The opinions expressed in this paper do not necessarily reflect the opinions of the European Union, or of any of its members.

APPENDIX

Proof of Proposition 3.1

PROOF. We denote with (w, z, q) the equilibrium components of the steady state. By imposing $dW/dt = 0$ in (5) we readily obtain:

$$w = \sqrt{\frac{2}{p}}.$$

It has to be noted that since $p \in [0, 1]$ the steady-state value of the TCP window is lower bounded by $\sqrt{2}$. By imposing $dZ/dt = 0$ in (6) we get:

$$\frac{1}{\tau R}(\tau - q/C) - \frac{z^2}{2R}p = 0,$$

which gives:

$$z = \begin{cases} w\sqrt{1 - \frac{q}{\tau C}} & 0 \leq q \leq \tau C \\ 0 & q > \tau C \end{cases} \tag{24}$$

Let us begin by focusing on the case $0 \leq q \leq \tau c$. By letting $\dot{q} = 0$ in (10) we obtain:

$$Nw + Nw\sqrt{1 - \frac{q}{\tau C}} = TC + q \tag{25}$$

We now make the change of variables:

$$x(q) = \sqrt{1 - \frac{q}{\tau C}} \tag{26}$$

that is a monotonically decreasing mapping from the set $[0, \tau C]$ to the set $[0, 1]$. By plugging (26) into (25) we get:

$$x^2 + \frac{Nw}{\tau C}x + \frac{Nw}{\tau C} - \frac{T}{\tau} - 1 = 0 \tag{27}$$

By solving (27) we get a unique positive solution:

$$x = \frac{1}{2}\left(\sqrt{\left(\frac{Nw}{\tau C} - 2\right)^2 + 4\frac{T}{\tau}} - \frac{Nw}{\tau C}\right) \tag{28}$$

The solution (28) is meaningful only if it belongs to the domain $[0, 1]$. Let us start by looking at the upper bound of the solution by plugging $x = 1$ into (28), which corresponds to the case $q = 0$:

$$\sqrt{\left(\frac{Nw}{\tau C} - 2\right)^2 + 4\frac{T}{\tau}} - \frac{Nw}{\tau C} = 2. \tag{29}$$

After a little algebra we obtain:

$$w_{min} = \frac{TC}{2N} \tag{30}$$

To get a physical interpretation of w_{min} we have to consider that for this value of w the queue is zero and $z(w_{min}) = w$, i.e. the N TCP and N LEDBAT flow obtain exactly the same per flow rate equal to w_{min}/T. It has also be noticed that the sum of the rates of the LEDBAT flows and TCP flows is exactly equal to the capacity of the link:

$$N\frac{w_{min}}{T + \frac{q}{C}} + N\frac{w_{min}}{T + \frac{q}{C}} = \frac{2N}{T}w_{min} = C \tag{31}$$

Let us now consider the other extreme case $x = 0$ corresponding to $q = \tau c$. In this case by substituting $x = 0$ in (28) we get:

$$w_{max} = \frac{C}{N}(T + \tau) \tag{32}$$

Thus, we conclude that the interval $0 < q < \tau C$ maps to the interval $TC/(2N) < w < C(T + \tau)/N$. By considering (28) and (26) we obtain the equilibrium for the queue when $TC/(2N) < w < C(T + \tau)/N$:

$$q = \tau C \left[1 - \frac{1}{4} \left(\sqrt{\left(\frac{Nw}{\tau C} - 2 \right)^2 + 4\frac{T}{\tau}} - \frac{Nw}{\tau C} \right)^2 \right] \tag{33}$$

It is important to notice that $z = 0$ when $w = w_{max}$, meaning that in this case the whole link capacity is allocated to the N TCP flows. Finally, by plugging (33) in (24) we get the LEDBAT window size at steady-state when $TC/(2N) < w < C(T + \tau)/N$:

$$z = \frac{w}{2} \left[\sqrt{\left(\frac{Nw}{\tau C} - 2 \right)^2 + 4\frac{T}{\tau}} - \frac{Nw}{\tau C} \right]. \tag{34}$$

Let us now consider the case $q \geq \tau C$ that occurs when $w \geq C(T + \tau)/N$. In this case $z = 0$ and, as a consequence, $q = Nw - TC > 0$.

To conclude the proof we have to consider the case $w < w_{min} = TC/(2N)$. It is easy to check that $q = 0$ and thus by plugging this value in (24) we obtain $z = w$. $\square$

REFERENCES

B.COHEN. How has bittorrent as a protocol evolved over time. http://www.quora.com/BitTorrent-protocol-company.

BENAMEUR, N., GUILLEMIN, F., AND MUSCARIELLO, L. 2013. Latency reduction in home access gateways with shortest queue first. In *Proc. ISOC Workshop on Reducing Internet Latency*.

BIZON, M. 2005. [nouveaux firmwares freebox v3 & v4] aduf - historique firmware, 01/07/2005. http://88.191.250.12/viewtopic.php?t=164746&view=previous.

CAROFIGLIO, G., MUSCARIELLO, L., ROSSI, D., AND TESTA, C. 2010. A hands-on Assessment of Transport Protocols with Lower than Best Effort Priority. In *IEEE LCN*.

CAROFIGLIO, G., MUSCARIELLO, L., ROSSI, D., TESTA, C., AND VALENTI, S. 2013. Rethinking low extra delay backtround transport protocols. *Elsevier Computer Networks*, 1838–1852.

COHEN, B. AND NORBERG, A. 2010. Correcting for clock drift in uTP and LEDBAT. In *9th USENIX International Workshop on Peer-to-Peer Systems (IPTPS'10)*.

CROWCROFT, J., WAKEMAN, I., WANG, Z., AND SIROVICA, D. 1992. Is layering, harmful? *IEEE Network 6,* 1, 20–24.

DE CICCO, L., MASCOLO, S., AND NICULESCU, S.-I. 2011. Robust stability analysis of smith predictor-based congestion control algorithms for computer networks. *Automatica 47,* 8, 1685–1692.

FLOYD, S. AND JACOBSON, V. 1993. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking 1,* 4, 397–413.

GETTYS, J. AND NICHOLS, K. 2012. Bufferbloat: dark buffers in the internet. *Communications of the ACM 55,* 1, 57–65.

GONG, Y., ROSSI, D., AND LEONARDI, E. 2013a. Modeling the interdependency of low-priority congestion control and active queue management. In *The 25th International Teletraffic Congress (ITC25)*.

GONG, Y., ROSSI, D., TESTA, C., VALENTI, S., AND TAHT, D. 2013b. Fighting the bufferbloat: on the coexistence of aqm and low priority congestion control. In *IEEE INFOCOM Workshop on Traffic Measurement and Analysis (TMA'13)*. Turin, Italy.

GONG, Y., ROSSI, D., TESTA, C., VALENTI, S., AND THT, M. 2014. Fighting the bufferbloat: On the coexistence of aqm and low priority congestion control. *Computer Networks 65,* 0, 255 – 267.

GRIECO, L. A. AND MASCOLO, S. 2004. Performance Evaluation and Comparison of Westwood+, New Reno, and Vegas TCP Congestion Control. *ACM SIGCOMM Computer Communication Review 34,* 2, 25–38.

HASSARD, B. D., KAZARINOFF, N. D., AND WAN, Y.-H. 1981. *Theory and applications of Hopf bifurcation*. Vol. 41. CUP Archive.

HOLLOT, C., MISRA, V., TOWSLEY, D., AND GONG, W. 2001. A control theoretic analysis of RED. In *IEEE INFOCOM'01*. Vol. 3. 1510–1519.

JIANG, H., WANG, Y., LEE, K., AND RHEE, I. 2012. Tackling bufferbloat in 3G/4G networks. In *ACM IMC*.

KEY, P., MASSOULIÉ, L., AND WANG, B. 2004. Emulating low-priority transport at the application layer: a background transfer service. In *ACM SIGMETRICS*. New York City, NY.

KREIBICH, C., WEAVER, N., NECHAEV, B., AND PAXSON, V. 2010. Netalyzr: Illuminating the edge network. In *ACM SIGCOMM Internet Measurement Conference (IMC'10)*. 246–259.

KUZMANOVIC, A. AND KNIGHTLY, E. 2003. TCP-LP: A distributed algorithm for low priority data transfer. In *IEEE INFOCOM*.

LIU, S., VOJNOVIC, M., AND GUNAWARDENA, D. 2006. 4cp: Competitive and considerate congestion control protocol. In *ACM SIGCOMM*.

LIU, Y., PRESTI, F. L., MISRA, V., TOWSLEY, D., AND GU, Y. 2003. Fluid models and solutions for large-scale IP networks. In *ACM SIGMETRICS*. Vol. 31. ACM, 91–101.

MARSAN, M. A., GARETTO, M., GIACCONE, P., LEONARDI, E., SCHIATTARELLA, E., AND TARELLO, A. 2005. Using partial differential equations to model TCP mice and elephants in large IP networks. *IEEE/ACM Transactions on Networking, 13,* 6, 1289–1301.

MASCOLO, S. 1999. Congestion control in high-speed communication networks using the Smith principle. *Special Issue on "Control methods for communication networks" Automatica 35,* 12, 1921–1935.

MCKENNEY, P. 1990. Stochastic fairness queueing. In *IEEE INFOCOM*.

MICHIELS, W. 2011. Spectrum-based stability analysis and stabilisation of systems described by delay differential algebraic equations. *IET control theory & applications 5,* 16, 1829–1842.

MICHIELS, W., MELCHOR-AGUILAR, D., AND NICULESCU, S.-I. 2006. Stability analysis of some classes of TCP/AQM networks. *International Journal of Control 79,* 9, 1136–1144.

MISRA, V., GONG, W., AND TOWSLEY, D. 2000. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. *ACM SIGCOMM 30,* 4, 151–160.

NICHOLS, K. AND JACOBSON, V. 2012. Controlling queue delay. *Commun. ACM 55,* 7, 42–50.

NICULESCU, S.-I. AND MICHIELS, W. 2007. *Stability and stabilization of time-delay systems. An eigenvalue-based approach*. SIAM: Philadelphia, USA.

PAN, R., PRABHAKAR, B., AND PSOUNIS, K. 2000. Choke - a stateless active queue management scheme for approximating fair bandwidth allocation. In *IEEE INFOCOM*.

ROSSI, D., TESTA, C., VALENTI, S., AND MUSCARIELLO, L. 2010. LEDBAT: the new BitTorrent congestion control protocol. In *IEEE ICCCN*.

SANDVINE. 2014. Global Internet Phenomena Report (1H 2014). `https://www.sandvine.com/downloads/general/global-internet-phenomena/2014/1h-2014-global-internet-phenomena-report.pdf`.

SCHNEIDER, J., WAGNER, J., WINTER, R., AND H.J.KOLBE. 2010. Out of my way – evaluating low extra delay background transport in an ADSL access network. In *ITC22*.

SHALUNOV, S., HAZEL, G., IYENGAR, J., AND KUEHLEWIN, M. 2012. Low Extra Delay Background Transport (LEDBAT). IETF RFC 6817.

SHREEDHAR, M. AND VARGHESE, G. 1995. Efficient fair queueing using deficit round robin. In *ACM SIGCOMM*.

TRANG, S. Q. V., KUHN, N., LOCHIN, E., BAUDOIN, C., DUBOIS, E., AND GELARD, P. 2014. On the existence of optimal ledbat parameters. In *IEEE International Conference on Communications (ICC)*. 1216–1221.

VENKATARAMANI, A., KOKKU, R., AND DAHLIN, M. 2002. TCP Nice: A mechanism for background transfers. In *USENIX OSDI*.

WHITE, G. 2014. Active queue management in docsis 3.x cable modems. Tech. rep., CableLabs. May.