Computer Networks 57 (2013) 1838–1852

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Rethinking the Low Extra Delay Background Transport (LEDBAT) Protocol

Giovanna Carofiglio [b], Luca Muscariello [c], Dario Rossi [a,*], Claudio Testa [a], Silvio Valenti [a,1]

[a] Télécom ParisTech, 46 rue Barrault, 75634 Paris, France
[b] Bell Labs, Alcatel-Lucent Villarceaux, Route de Villejust 91620 Nozay, France
[c] France Télécom R&D – Orange Labs, 38-40 rue du général Leclerc, 92794 Issy-les-Moulineaux, France

ARTICLE INFO

Article history:
Received 3 September 2012
Accepted 11 February 2013
Available online 25 March 2013

Keywords:
LEDBAT
Congestion control
Latecomer unfairness
BitTorrent

ABSTRACT

BitTorrent has recently introduced LEDBAT, a novel application-layer congestion control protocol for data exchange. The protocol design assumes that network bottlenecks are at the access of the network, and that thus user traffic competes creating self-induced congestion. To relieve this phenomenon, LEDBAT is designed to quickly infer when self-induced congestion is approaching (by detecting relative changes of the one-way delay in the transmission path), and to react promptly by reducing the sending rate prior to the congestion occurrence. Previous work has however shown LEDBAT to be affected by a latecomer advantage, where newly arriving connections can starve already existing flows. In this work, we propose modifications to the congestion window update mechanism of LEDBAT that solve this issue, thus guaranteeing intra-protocol fairness and efficiency. Closed-form expressions for the stationary throughput and queue occupancy are provided via a fluid model, whose accuracy is confirmed by means of ns2 packet level simulations. Our results show that the proposed change can effectively solve the latecomer issue, furthermore without affecting the other original LEDBAT goals.

1. Introduction

As recently pointed out in [9], "Internet delays now are as common as they are maddening". The root cause for these delays can be identified with the excess buffering inside a network, which is nicknamed "bufferbloat". Though this is nothing new [10], the situation got worse in the latest years due to mainly two facts: (i) TCP loss-based design, that forces the bottleneck buffer to fill before the sender reduces his rate and (ii) the fact that low uplink capacities of widely deployed ADSL and Cable modems can translate into significant queuing delay (up to few seconds [24]).

Evidently, BitTorrent engineers were well aware of this fact. Indeed, the popular peer-to-peer file sharing system with hundreds of millions of daily users worldwide, has recently developed a novel application-layer congestion control protocol for data exchange. The novel insight in the widely explored congestion control landscape is in this case the reasonable assumption that the bottleneck is most likely at the access of the network (e.g., at the ADSL modem line), which means that congestion is therefore typically *self-induced* by concurrent traffic sessions generated by the same user (e.g., BitTorrent transfers in parallel with Skype call and Web browsing). The new protocol, named LEDBAT after *Low Extra Delay Background Transport*, is designed to solve this issue and targets (i) efficient but (ii) low priority transfers. When LEDBAT flows have the exclusive use of the bottleneck resources, they fully exploit the available capacity. When instead other transfers – such as VoIP, gaming, Web or other TCP flows – are ongoing, LEDBAT flows back off to avoid harming the performance of interactive traffic. To attain the efficiency aim, LEDBAT

* Corresponding author.
E-mail address: dario.rossi@enst.fr (D. Rossi).
[1] Current affiliation: Google Inc.

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.02.020

flows need to create queuing, as otherwise the capacity would not be fully utilized. At the same time, due to the low-priority aim, the amount of extra queuing delay induced by LEDBAT flows should be small enough to avoid hurting the interactive traffic – hence the protocol name.

LEDBAT[2] has been defined as an IETF draft [37] (which focuses more on the algorithmic aspects) and as a BitTorrent Enhancement Proposal BEP-29 [32] (that instead focuses more on the UDP framing). and has recently become the BitTorrent default congestion control protocol, replacing thus TCP for the data transfer.

While previous research [33,34,8,7,4,13,36,39,40] on LEDBAT has shown its potential for P2P transfers it has also highlighted some serious fairness deficiencies. In more details, LEDBAT has a good interplay with BitTorrent-like transmissions of short-lived chunk-long flows [39,40]: hence, the definition of this protocol under a BEP document makes perfectly sense, as the performance improvement for BitTorrent is in this case coupled to a reduction of the self-induced bufferbloat. At the same time, LEDBAT is affected by a latecomer unfairness issue [34,8] that arises in (the not so uncommon) case of backlogged flows: under this conditions, latecomer flow takes over the bottleneck resource, starving the first-comers. As such, the normalization of an ill-defined protocol can have potentially dramatic effects: while the phenomenon hardly ever happens in P2P swarm like content delivery, it can severely impact the performance of backups, photo uploads and, more generally, long lived uploads of home users towards their virtual storage (e.g. DropBox, Google, Windows Live, Apple iCloud, etc.).

With respect to the normalization of an IETF congestion control algorithm, whose scope goes beyond a specific application, though popular it may be, fairness is a significant property that should always be taken into account for the design of any experimental or deploy-able protocol. The lack of fairness is an indication of poor convergence properties, e.g. the algorithm is unstable because of an unwise choice of the parameters or, more tricky, the algorithm cannot be stable for any choice of parameter. We show in this paper that LEDBAT falls in this second category.

The main contribution of this work is to propose a modification to the LEDBAT congestion control that, leaving untouched the design goals, solves the fairness issue – therefore avoiding unwanted effects at the application-layer altogether. Throughout this paper, we make use of several complementary techniques to study our proposal. First, we use passive measurements to gauge the popularity of LEDBAT transfers in the real Internet, and exploit an active testbed methodology to show the fairness issue in current BitTorrent. We propose a modification to the original LEDBAT protocol, to jointly achieve fairness and efficiency, and develop a fluid model to describe the system dynamics. An analytical solution of the model proves the soundness of our design, while numerical solutions allow us to grasp the transient phase as well. Finally, we use extensive `ns2` packet-level simulations to evaluate the LEDBAT performance under several scenarios. Since we need to ensure that our proposal works as expected under any circumstances, we include the general case of backlogged transfers (e.g., two concurrent low priority backups). At the same time, we need to ensure that LEDBAT does not harm the experience of BitTorrent users, hence we include P2P-like scenarios involving multiple-flows and a heterogeneous network environment as well.

The remainder of this paper is organized as follows. Related work and motivations are covered in Sections 2 and 3 respectively. The unfairness issue is introduced in Section 4, followed by our proposed modification in Section 5 along with the fluid model and its analytical solution. More complex network scenarios are tackled by means of packet-level simulations starting from Section 6, where we also compare the numerical solution of the fluid model with simulation results. We then proceed by studying the impact of the traffic model (e.g., backlogged vs. chunk-based transfers) in Section 7, the sensitivity of the protocol to parameter changes in Section 8. Lastly we evaluate the performance of the protocol in P2P-like settings from a single-peer as well as a whole swarm perspective in Section 9, before Section 10 concludes the paper.

2. Related work

Congestion control studies on the Internet date back to [21] and it is out-of-scope to provide a full review of the existing literature here. Still, a couple of references are worth citing as they share LEDBAT low-priority spirit [41,25,28,23]. For instance, TCP-LP [25] and NICE [41] share the same goal as LEDBAT aiming at implementing a *Lower-than Best Effort (LBE)* service for background transfers. In more detail, NICE extends the delay-based behavior, typical of TCP Vegas, with a multiplicative decrease reaction to early congestion, which is actually detected when the number of packets experiencing a large delay in an RTT exceeds a given threshold. On the other hand, TCP-LP enhances the loss-based behavior of TCP NewReno with an early congestion detection based on the distance of the instantaneous One-Way Delay from a weighted moving average calculated on all observations. In case of congestion, the protocol halves the rate and enters in a inference phase, during which, if further congestion is detected, the congestion window is set to zero and normal TCP NewReno behavior is restarted. At the same time, TCP-LP [25], NICE [41] differ from LEDBAT in that the latter aims at introducing a *bounded* extra delay: i.e., when queuing delay reaches a given target, the LEDBAT protocol slows down its transmission rate to ensure the queuing delay target is not exceeded. Notice that this is especially important for VoIP, gaming, and all other interactive delay-sensitive applications.

Related work has already tackled the intra-protocol fairness issue affecting delay-based congestion control algorithms. In particular, regarding TCP Vegas [6] which is the first example of such a family of techniques, the problem was pointed out [31] in relation to (i) to route changes and (ii) to persistent congestion when multiple concurrent Vegas flows compete at the same bottleneck. The latter is the same malfunctioning we spotted in the

[2] The protocol has been christened as LEDBAT in the IETF, and as uTP in the BitTorrent BEP community: to avoid ambiguity, we use its IETF name.