Designing incentive compatible protocols for background data transfers

Costas Courcoubetis
Department of Engineering Systems and Design
Singapore University of Technology and Design
Singapore
costas@sutd.edu.sg

Antonis Dimakis, Michail Kanakakis
Department of Informatics
Athens University of Economics and Business
Athens, Greece
{dimakis, kanakakis}@aueb.gr

***Abstract*— We consider the design of transport protocols for background transfers with the aim of reducing the download delays of interactive (non-background) flows which share the same links. We demonstrate that Lower than Best Effort (LBE) protocols suffer from bandwidth starvation, a fact which not only creates adoption disincentives but can have a harmful effect on the interactive flows as well. Instead we propose a simple access control policy which limits the maximum number of simultaneously transmitting background flows. This maximum limit is dynamically adjusted in order to prevent bandwidth starvation thus eliminating the disincentives present in LBE protocols. Through simulations we show that our policy creates lower transfer delay for the interactive flows, as compared to that under LEDBAT [6] and TCP-LP [8] LBE protocols.**

I. Introduction

The incumbent TCP congestion control protocol was designed with the fairness philosophy to continuously provide equal allocation of bandwidth resources among competing flows. This fundamental fairness axiom is unaligned with the diverse needs and characteristics of various applications that arose with time. Background tasks such as peer-to-peer file-sharing, bulk updates, caching, and interactive tasks such as web browsing are all treated equally if they use TCP. This can have a negative impact on the task completion delay of interactive applications, since background applications could have equally refrained from transmitting during times when bandwidth is scarce, thus letting interactive applications such as web browsers complete downloads sooner. Indeed many ISPs operate outside the internet architecture by throttling file-sharing traffic using deep packet inspection (DPI) at network equipment [1]. Such network-centric solutions rest on DPI being able to counter header obfuscation tricks by peer-to-peer clients, and then again throttling may lead to unnecessary underutilization of network resources. For these reasons, a user-centric solution might be more preferable.

The shift of focus from the network to users raises two issues: 1) the users' incentives must be compatible with the behavior of these protocols; otherwise protocol adoption is compromised, and 2) it is more complex to predict overall system performance because control is exercised by multiple users.

The incentives can be economic, as is the case in [2] [3] [4] and [5], where congestion pricing is considered. Such pricing mechanisms require special monitoring equipment which may not be always available. Still even without pricing, owing to their tolerance to high transfer delays, background users are largely indifferent to which transfer protocol they use, provided their transfers complete in some finite time. Thus they might as well use protocols which yield to other traffic when they coexist on the same link. To this end, Less than Best Effort (LBE) protocols, namely LEDBAT [6] [7], TCP-LP [8] and TCP-nice [9], were developed recently to address this lower prioritization. The key premise is that although the value of background users is not significantly reduced when they use LBE protocols, interactive users are likely to see significant reductions in their transfer delay. Owing to this design goal, there are cases where LBE flows could obtain extremely low throughput even under light congestion, e.g., when they compete with persistent flows which use a more aggressive protocol than LBE such as TCP. Thus, LBE flows are vulnerable to *bandwidth starvation* which clearly presents a disincentive in their adoption since extremely high delays are likely to be intolerable. A motivating observation (made in Section II) for the work in this paper, is that bandwidth starvation has a negative impact on the delays of *interactive flows as well*.

Extending related literature (e.g. [10]) that analyzes the performance of LBE protocols over static traffic conditions, i.e., constant number of flows competing for the bottleneck capacity, we highlight situations where their adoption achieves negligible or even opposite results with respect to the aforementioned metric when the number of flows is allowed to vary. In socioeconomic terms, we claim that being excessively altruistic may end up to worst states for all involved stakeholders which makes the value of the adoption of such technologies questionable. Contrary to what one may have perhaps expected, if background applications use more aggressive protocols than LBE -that may complete with TCP more equally-, interactive users might see an even greater delay reduction. This effect is, as explained in Section II, due to the overly polite nature of LBE protocols which makes background flows achieve extremely low throughput during times of high congestion; subsequently a high number of them is accumulated and so by Little's law transfer delays are increased accordingly.

In Section III we propose a simple access control scheme meant to control streams of background TCP flows belonging to individual users, e.g., by being implemented in download manager software. It consists of limiting the maximum number of background flows currently transferring data. The maximum number of such "active" flows is automatically and continuously revised in order to minimize the impact on the interactive flows while keeping the number of "inactive", i.e., not yet transmitting background flows, bounded. In Section IV, we compare the performance of the access control scheme with other LBE protocols, namely two versions of LEDBAT of different "politeness" or aggressivity levels, and TCP-LP.

II. MOTIVATION

Consider a single bottleneck topology with capacity C, traversed by the following traffic mixture: a number of long-lived TCP flows transferring bulk data and two streams of dynamically arriving interactive and background flows, where the latter have switched from TCP to an LBE protocol. Each flow in the streams is associated with the transfer of a finite file size and the average loads of these streams, denoted as ρ_s and ρ_b respectively, are quantified as a portion of the available capacity C. The load composition described above, could correspond to a batch of software updates, web traffic, and chunks transfers of a file-sharing application; we use these terms pairwise interchangeably throughout the paper.

In this context, as time passes, the modulatory parameters of the LBE protocols that tune their aggressiveness lead them to gain arbitrary amounts of throughput as an aggregate, even if they all go through a period of starvation. Intuitively, this happens due to the fact that because of the always present long-lived TCP flows, each LBE flow obtains a negligible amount of bandwidth[1] and new entering flows accumulate in the network making their number unstable. But each such flow probes with a fixed rate the network for the right conditions to send data, and this traffic eventually becomes substantial increasing the delays of the interactive flows.

A similar effect may appear even in the absence of long-lived flows, when the LBE flows are routed over a two hops topology, each link with capacity C, competing with two independent streams of interactive cross traffic, each one with rate ρ_s. In this case, the presence of a single web flow at any of the links - an event not appearing with probability $(1-\rho_s)^2$ – should cause the background ones to completely yield. Thus the stability condition for the LBE flows is:

$$\rho_b \leq (1-\rho_s)^2, \tag{1}$$

instead of the traditional $\rho_b \leq (1-\rho_s)$ one would expect. If $(1-\rho_s)^2 \leq \rho_b \leq (1-\rho_s)$ then the number of LBE flows will become unstable and they will interfere with the interactive traffic as mentioned earlier. Hence LBE protocols may lead to inefficient capacity utilization by the background traffic. Moreover, the stability condition becomes exponentially stricter as further links are added on the respective path.

Our discussion suggests that low priority traffic should obtain a minimum average throughput that ensures its stability in the network. This prerequisite is also justified from the economics perspective: the extensively unfair treatment causes unacceptably long completion delays for background transfers which is commonly considered a disincentive for LBE adoption. But a predefined portion of capacity sharing, cannot be guaranteed by existing LBE protocols as it strongly depends both on the parameters determining their aggressiveness and on the traffic context.

Our core contribution is to combine the objectives of minimum delays for interactive flows while providing sufficient throughput to background transfers such that their arriving load is actually served during a relatively long time period. The proposed access control policy that follows, achieves these objectives utilizing information available at the network edges, without requiring any modifications on the current network or transport layer protocols.

To make our model more precise in terms of economics, we assume that flows can choose the protocol that is used to serve them. In particular we assume that flows (being controlled by users) can in turn choose between TCP and FAIRBAT (our new LBE proposal). There are two economic types of such flows. The "interactive" flows care for latency while the "background" flows obtain their maximum value from the service if each file is successfully transmitted over some large time interval with some very high probability. If the system offers this capability both under TCP and also though some other technology less aggressive to others, then they prefer the later one ("ε-altruists").

We say that an LBE technology is "incentive compatible" if interactive flows choose TCP and background flows choose the LBE technology. Our aim is to devise an LBE technology that creates the least negative externalities to the interactive traffic while keeping the above incentive compatibility properties.

III. IMPLEMENTATION

In this section we describe the heuristic for implementing FAIRBAT ("Fair Background Traffic" protocol), our proposed congestion control algorithm that is also "fair" to the background flows. We assume as before that the background information is divided into "chunks" (files with some length distribution) and there is an arrival rate of such chunks at the some network node with some specific destination. In the simplest case the network consists of a single link. A chunk becomes "active" if it starts transmitting in which case it turns into a flow.

Our approach consists in applying the suitable network access policy on background flows, while keeping the TCP-AIMD dynamics unchanged. The central idea is to control as to minimize the maximum number of active chunks, i.e. those that are allowed to transmit simultaneously, subject to the incentive-compatible requirement defined previously, that all arrivals are eventually served over a large enough time window. This is feasible if chunks do not enter the network (become active) when they arrive, but are temporarily queued being in an "inactive state" in a buffer located at the sender's side. Then,

Each flow still reserves a minimum throughput share, – at least a packet per round trip time – probing for intervals of congestion reduction that would imply its rate to increase.[1]

when any of the active ones completes its file transfer, the first chunk in the buffer queue is activated and starts transmitting. In this way, the inactive chunks enter the network sequentially in a FIFO fashion, while the maximum number of simultaneous transmissions is not violated. This idea is justified by the theory developed in [11] where it is shown that using wTCP for the background traffic aggregate and tuning w so that the background traffic queue becomes marginally stable, is a nearly optimal strategy for reducing the delay of the interactive traffic. It achieves a reduction of the average delay of the interactive flows that is within 17% of the minimum delay achieved by the optimal policy for serving the background traffic, which requires full information on the network state (number of active flows at the routers). Our implementation opens w TCP flows having the same effect as a single wTCP connection. If w is fractional, then our implementation will alternate opening an integer number of TCP connections equal to the floor and the ceiling of w.

We assume that the arriving background traffic chunks are stored in a buffer at the sender. Our goal is to design a practical algorithm for updating the number m of active chunks so that the queue at the sender buffer becomes marginally stable. An equivalent way to think about this is to find m such that the average bandwidth of the m TCP connections becomes equal to the load of the aggregate arriving background traffic. In theory this depends on many unobservable factors such as the load of the interactive traffic and the number of long-lived TCP connections competing on the same links. This motivates our approach to estimate m in an adaptive way by doing actual measurements.

The idea is simple: we observe the size of the chunk queue and if this has a tendency to grow we increase m; if it tends to shrink, then we decrease m. This update of m is done smoothly by reacting slowly to the queue dynamics. The goal is to keep the queue fluctuating around some high but constant level, where emptying is rare, and achieve this by varying m as little as possible over time (in the steady state). Keeping a constant aggressivity for the background aggregate has the best effect in reducing the delays of the interactive traffic as our theory in [11] suggests.

To this end, we fix a positive target level τ for the queue size and a time interval T for updating our control parameter m. Then for $t = 0, T, 2T, ...$ we update m as follows

$$m(t+1) = m(t) + \gamma\,(n(t) - \tau), \qquad (2)$$

where n is the queue size and $\gamma > 0$ is some sufficiently small coefficient to ensure convergence. Note that the target τ is used as a benchmark and its exact value has no impact on the limiting value of m. Thus, it is set arbitrarily but high enough, such that the queue size may temporarily fluctuate below it without reaching zero values. Emptying the buffer frequently would bias the adjustment process towards higher values of m. The choice of the value of γ causes a tradeoff between the convergence speed and the oscillations around the optimal value. In our implementation we address this tradeoff by starting with a high value and gradually decreasing it while convergence is achieved. Discussing such issues is out of scope in this paper as we mainly focus on the performance evaluation at convergence. We leave them for an extended presentation of our work.

IV. Simulation Results

In this section, we present the results of our simulations performed on ns-2 [12] for a comparative analysis between our proposed access control policy (FAIRBAT) and other LBE protocols. In all the following scenarios, their performance is compared with the relevant basis experiment, where all flows transmit over TCP and both types of short flows (background and web) enter the network when they arrive (the traditional implementation). When implementing FAIRBAT and other LBEs, the only change in the basis experiment is for handling the background traffic: in an LBE case, chunks enter according to the same arrival process but are served by the corresponding protocol, while over FAIRBAT they keep on using TCP but the controller on the number of active connections is applied. The two types of (short) flow arrivals (web and background) are Poisson and their rates will be explicitly mentioned in each subsection. Each such flow is associated with the transmission of a finite, exponentially distributed file size with mean 3Mbytes. For the topology, we consider a bottleneck uplink with capacity $C = 10Mbps$ and a network buffer of $B = 100$ packets, which are both duplicated in the two hops cases. Finally, we set a separate downlink for the acknowledgements, to avoid any secondary interactions between packets.

We evaluate LEDBAT using its default aggressivity parameterization and a modification (let it be defined as LEDBAT~a), tuning its parameters such that it becomes more competitive in throughput allocation. Our aim is to indicate that LBEs underperformance over unstable states is not a modulatory issue (poor parameter choices), but an inherent weakness of their design. We utilized the source coding for LEDBAT available in [13], and the default version of TCP-LP, not considering any further parameterization. Its encoding was included in ns-2.

In the next sections we show the weaknesses of the LBEs compared to FAIRBAT when (a) long-lived background flows are present, and (b) when the stability condition (1) is not satisfied. The first case is examined in the simplest topology of a single link, while the second needs a network setup and we analyze it for the simplest network, consisting of two links.

A. Single Bottleneck

We set the web flows arriving at rate 1/8 flows/sec and the rate of background flows at {1/12, 1/8, 1/6, 1/4.8, 1/4} flows/sec, resulting to loads of $\rho_s = 0.3$ and $\rho_b = \{0.2, 0.3, 0.4, 0.5, 0.6\}$ respectively, expressed in terms of capacity portion. Firstly, we consider the case of long-lived flows absence, where the stability condition is satisfied. In Fig. 1a we present the percentage improvement on the average delay of web flows, where the LBE protocols achieve their design objective. Actually, LEDBAT and TCP-LP outperform FAIRBAT for low loads, a fact which is interpreted by Fig. 1b where we depict the ratio of the average active chunk number compared to just opening TCP connections for each arriving chunk (our basis for comparisons). Note that all LBE active chunks apart from FAIRBAT cumulate up to a different level. This tendency is strictly increasing with ρ_b for both versions of LEDBAT, while it starts decreasing at $\rho_b = 0.4$ for TCP-LP, a fact which is explained as follows: over the complete range of load values, the LEDBAT dynamics as the load increases lead the flows to further limit their priority and hence decrease their aggressivity compared to TCP, leading to a higher relative number of active connections. For TCP-LP this behaviour is observed until $\rho_b = 0.4$, where the analysis of the experimental results suggests that connections reach their minimum transmission rate and can't backoff any more, while TCP connections keep reducing their average rates as the load increases further. Actually, in this region of higher loads FAIRBAT starts performing better than TCP-LP being able to adapt to the increasing load conditions.

Concerning LEDBAT~a, its higher aggressiveness (denoted by comparing its accumulation ratio with LEDBAT), results to lower gains for web traffic. Thus here, the modulatory parameters strongly affect its performance, contrary to the unstable states as it will be explained next.

The situation changes drastically when we add three long-lived flows to run in the background. Then, when using any LBE besides FAIRBAT, the number of active chunks becomes theoretically unstable, but practically it reaches a very large number since each connection becomes non-responsive and uses some fixed minimum bandwidth. In Fig. 2, we present these results, observing a clear gain by using FAIRBAT, especially for the higher loads. On the opposite, others' LBE impact is negligible or even negative. The main reason is that the corresponding flows become, even for low loads, unresponsive to congestion and accumulate in very large numbers contrary to their design philosophy. Therefore, the LBE altruism observed in Fig. 1 does not effectively work and this problem cannot be remedied by tuning differently their parameters.

Apart from the impact on average, another noteworthy issue is the way that this magnitude is shared among individual web flows. A metric to be utilized is the normalized sample variance of their delays, i.e., the relative value when using each protocol over our comparison basis. In Table I, we document our measurements. FAIRBAT achieves the lowest values, meaning that its improvement is more predictable and allocated in a more uniform way among the web flows.

In Fig. 3, we present the performance of FAIRBAT alone by varying the number k of long-lived flows. Due to space limitations we omit the comparative results for other LBEs, even though we have performed the relative experiments and they lie around zero (in agreement with Fig. 2a). Our improvement is proportional to ρ_b and inversely proportional to the factor k+1. The first correlation validates the reasonable conjecture that applying the access policy to higher capacity portions demanded by the background traffic, leads to more substantial results. For the latter, the presence of long lived flows is contrary to FAIRBAT objectives: the protocol becomes more aggressive (converges to more simultaneously active chunks) *to meet the incentive compatible property*, limiting therefor the positive impact on web flows.

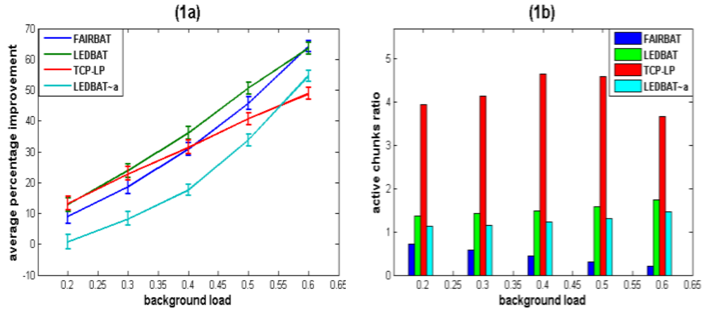

Fig. 1. *Comparison of LBEs in the absence of long-lived flows: Fig. 1a shows the percentage average delay improvement for the web traffic and Fig. 1b the average active connection (chunks) number ratio, compared to the case where all flows use TCP. In Fig 1a we observe that the LBEs improve the delays of the web flows and this effect becomes more prominent as the load of the background traffic increases. In Fig 1b we observe that both FAIRBAT and LEDBAT keep adapting for the whole loads range, while TCP-LP stops for loads larger than 0.4. This explains its performance deterioration in Fig 1a.*

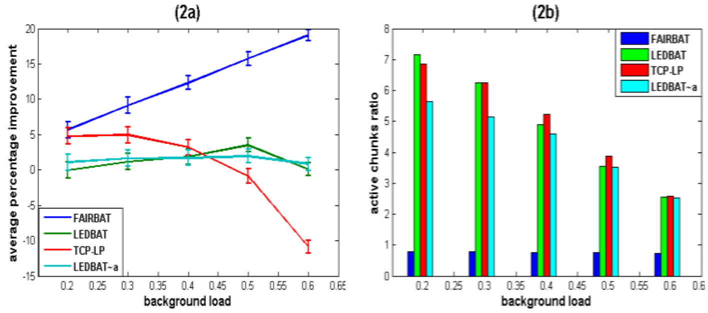

Fig. 2. *Comparison of LBEs in the presence of three long-lived flows, (in analogy to Fig. 1). In Fig 2a we observe a clear impact of improvement by using FAIRBAT, while the other LBEs either have no effects or perform worse than using TCP for the background traffic. The reason is that the system using any LBE besides FAIRBAT becomes unstable due to the presence of long-lived flows and each LBE flow ends up getting a fixed throughput, becoming unresponsive to congestion and hence to the load increase.*

Table I. Normalized sample variance of the web flows delays.

Protocol	Load of background flows ρ_b				
	0.2	*0.3*	*0.4*	*0.5*	*0.6*
FAIRBAT	0.8314	0.7173	0.6278	0.5763	0.4679
LEDBAT	0.9900	0.9706	0.9637	0.9407	1.131
TCP-LP	0.9327	0.9152	0.9727	1.0891	1.443
LEDBAT~a	0.9574	0.9672	0.9728	0.9728	1.067

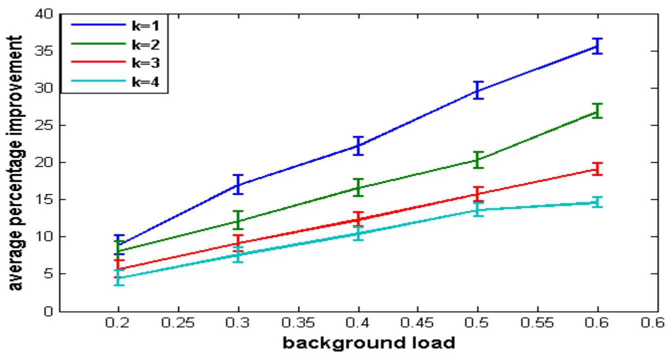

Fig. 3. *FAIRBAT: percentage average delay improvement for the web traffic compared to using TCP for the background flows, dimensioned on the number of long lived flows and the background traffic load. To meet the incentive compatibility property FAIRBAT becomes more aggressive when competing with more long lived background flows, reducing the good impact on web flows.*

B. *Two Links*

In this section, we consider the traffic composition over the two links topology, as depicted in Fig. 4.

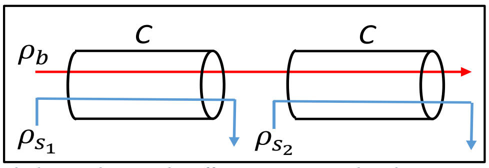

Fig. 4. *Two links topology and traffic mixture considered in section IV-B.*

In words, we fix a background stream traversing both links with load $\rho_b = 0.2$ and set two independent streams of web cross traffic with equal loads $\rho_{s_1} = \rho_{s_2} = \{0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7\}$. Our aim is to investigate how LBEs perform when the stability condition, defined by (1), is either satisfied or not.

In Fig. 5a, we present the relative results for the aggregate web traffic, because the impact on each individual stream is almost equal. The analysis is aligned with section IV-A. Within their stability region, LBEs prioritize web flows and achieve increasing improvement with congestion. This is consistent with their increasing number of active flows in Fig 5b, meaning that they efficiently further backoff as higher web loads enter the network without becoming unstable. On the contrary, when facing unstable conditions they get a small but fixed throughput and tend to accumulate less compared to TCP, whose throughput keeps reducing. Thus, in this case they fail to respond to congestion signals and consequently their impact is decelerated or limited, reaching even negative levels. Theoretically, the instability transition occurs for $\rho_s = 0.552$, but their actual accumulation behaviour denotes differentiations among protocols around this value, depending on their dynamics and protocol parameterization.

Here again, FAIRBAT outperforms all other LBEs especially for high web loads, while maintaining its increasing improvement throughout the whole web loads range. More precisely, its effect is inversely proportional to the factor $(1 - \rho_s)$, which is a reasonable result: although the background load is constant, its instantaneous aggressiveness control by the introduced access policy becomes more important at heavier congested environments.

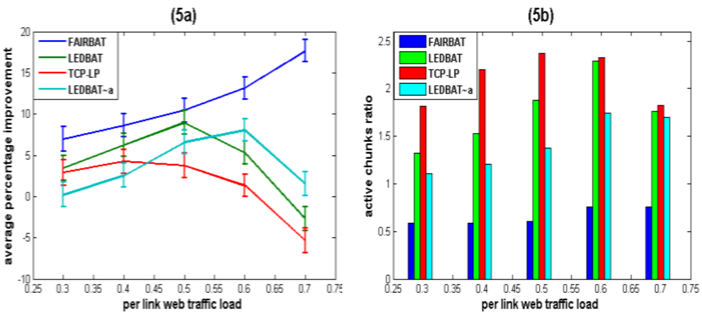

Fig. 5. *Comparison of LBEs under stability and instability conditions for prioritized traffic. In Fig 5a, we observe FAIRBAT achieving an increasing improvement for the whole web load range whereas the other LBEs deteriorate when the stability condition in (1) is violated. The LBEs performance, is consistent with their accumulation of active flows relative to using TCP in Fig. 5b, and aligned with the analysis in section IV-A.*

V. Concluding Remarks and Future Work

In this paper, we highlighted network conditions where existing LBEs face prolonged throughput starvation, leading to their unstable accumulation in the network. Their consequent unresponsive behaviour to congestion signals, results to poor or inverse than anticipated impact on web traffic delays, failing to meet their design objective. Instead, the proposed adaptive access control policy both satisfies the incentives-compatible requirement to serve the arriving stream during a large time interval and clearly outperforms other LBEs over unstable states. Combined with their close performance at stable situations, these properties make FAIRBAT a competitive alternative for background transfers.

Concerning our future work, we will investigate more complex topologies, including the case when the chunks do not share a common link with web traffic. In this scenario, the protocols dynamics are coupled by means of the long lived flows traversing the whole path, meaning that LBEs adoption affects only implicitly the interactive flows. Our aim is to verify that switching from TCP results to non-harmful outcomes, meaning that FAIRBAT has at least indifferent but not worse-off impact on the delay experienced by web flows.

References

[1] Max Planck Institute (March 18, 2008). "Glasnost: Results from tests for BitTorrent traffic blocking". [Online]. http://broadband.mpi-sws.org/transparency/results/ Retrieved April 3, 2011.

[2] Bob Briscoe, "A fairer, faster internet protocol," IEEE Spectrum, Dec. 2008, pp. 38–43, Dec.2008. [Online] Available: http://spectrum.ieee.org/telecom/standards/a-fairer-faster-internet-protocol

[3] Richard Gibbens and Frank P. Kelly, "Resource pricing and evolution of congestion control," Automatica, vol. 35, pp. 1969–1985, 1999.

[4] P. Key, L. Massoulie, and M. Vojnovic, "Farsighted users harness network time-diversity," in INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 4, 2005, pp. 2383–2394.

[5] C. Courcoubetis, A. Dimakis, "Pricing and Congestion Control for Volume-intensive Applications", 21st International Teletraffic Congress, Paris, 2008.

[6] Shalunov, Stanislav, Greg Hazel, Janardhan Iyengar, and Mirja Kuehlewind. "Low extra delay background transport (LEDBAT)." *draft-ietf-ledbat-congestion-04. txt* (2010).

[7] Rossi, Dario, Claudio Testa, Silvio Valenti, and Luca Muscariello. "LEDBAT: the new BitTorrent congestion control protocol." In *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pp. 1-6. IEEE, 2010.

[8] Kuzmanovic, Aleksandar, and Edward W. Knightly. "TCP-LP: low-priority service via end-point congestion control." *Networking, IEEE/ACM Transactions on* 14, no. 4 (2006): 739-752.

[9] Venkataramani, Arun, Ravi Kokku, and Mike Dahlin. "TCP Nice: A mechanism for background transfers." *ACM SIGOPS Operating Systems Review* 36, no. SI (2002): 329-343.

[10] Carofiglio, Giovanna, Luca Muscariello, Dario Rossi, and Claudio Testa. "A hands-on assessment of transport protocols with lower than best effort priority." In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pp. 8-15. IEEE, 2010.

[11] Courcoubetis Costas, and Antonis Dimakis. "Fair background data transfers of minimal delay impact." In *INFOCOM, 2012 Proceedings IEEE*, pp. 1053-1061. IEEE, 2012.

[12] The Network Simulator ns-2. [Online]: http://www.isi.edu/nsnam/ns/

[13] Dario Rossi: Software / Ledbat [Online] Available: http://perso.telecom-paristech.fr/~drossi/index.php?n=Software.LEDBAT.