# Congestion Control for Background Data Transfers with Minimal Delay Impact

Costas Courcoubetis, Antonis Dimakis, and Michalis Kanakakis

***Abstract*—Congestion control protocols for background data are commonly conceived and designed to emulate low priority traffic which yields to TCP flows. In the presence of even a few very long TCP flows, this behavior can cause bandwidth starvation and hence the accumulation of large numbers of background data flows for prolonged periods of time, which may ultimately have an adverse effect on the download delays of delay-sensitive TCP flows. In this paper we look at the fundamental problem of designing congestion control protocols for background traffic with the minimum impact on short TCP flows *while achieving a certain desired average throughput over time*. The corresponding optimal policy under various assumptions on the available information is obtained analytically. We give tight bounds of the distance between TCP-based background transfer protocols and the optimal policy, and identify the range of system parameters for which more sophisticated congestion control makes a noticeable difference. Based on these results, we propose an access control algorithm for systems where control on aggregates of background flows can be exercised, as in file servers. Simulations of simple network topologies suggest that this type of access control performs better than protocols emulating low priority over a wide range of parameters.**

***Index Terms*—Background data, bandwidth sharing, congestion control, delay-sensitive flows.**

## I. Introduction

A key element of the success of the internet architecture is the ability to accommodate current and future needs of very diverse applications. Connection rates differ by many orders of magnitude, while file transfer sizes vary by more than ten orders of magnitude. Nevertheless this is achieved using only a handful of transport protocols, mainly Transmission Control Protocol (TCP) and its variants, which in essence allocate network bandwidth to flows continuously so as to achieve fair sharing at all times. Indeed TCP 'fairness' or 'friendliness' [1] has become a common prescription for congestion control algorithms which intends to ensure equal sharing between flows.

But there are problems when all internet flows use the same protocol as applications do not equally value download delay. Web browsing and media streaming are delay-sensitive tasks where low web-page download delay and low initial playback latency (see [2]) is desirable, respectively. On the other hand, background data transfers such as large batch software or database updates are indifferent to small temporal variations of their bandwidth share, provided the data volume downloaded over a long time period is sufficient. It is well known from scheduling theory (see [3]) that shorter jobs or jobs with tighter deadlines should be assigned higher priority. Hence using TCP as the common transport protocol creates unnecessary delays to short and delay-sensitive flows.

C. Courcoubetis is with the Singapore University of Technology and Design. E-mail: costas@sutd.edu.sg

A. Dimakis and M. Kanakakis are with the Deparment of Informatics, Athens University of Economics and Business. E-mail: {dimakis,kanakakis}@aueb.gr

A possible solution, violating the end-to-end principle of the internet architecture, is for the internet service providers (ISPs) to intervene and throttle the bandwidth assigned to background data leaving more space for delay-sensitive traffic, or offering some form of prioritization. But this is not in many cases an efficient solution, since the ISPs cannot have the necessary information on how much throttling is necessary, and for which flows [4]. Also, unjustified throttling of traffic can have serious side effects for the ISP business, e.g., legal actions taken by disaffected end users [5].

Recognizing this, internet engineers have developed end-to-end 'less-than-best-effort' (LBE) congestion control protocols for background data transfers, such as TCP-LP [6], TCP-nice [7], uTorrent transport protocol [8], LEDBAT [9]. These are typically designed to emulate a low priority transport class which yields to TCP traffic, but this behavior can have a serious drawback under the presence of 'long' TCP flows, i.e., persistent or extremely long-lasting and always active flows, as we explain next, motivating our approach. In principle, during the time in which long TCP flows compete with *ideal* low priority flows, the latter suffer from bandwidth starvation and so their number grows arbitrarily as new low priority flows continue to arrive[1]. Since prioritization in reality is less than ideal (as demonstrated in the simulations of Section IV below), an excessive number of (nonideal) low priority flows will actually result. As each LBE flow transmits at least one packet per round-trip-time, the impact of a large number of such flows on the download delays of TCP flows can be significant. In Section IV, LBE protocols are shown to cause more performance degradation than TCP even (see Fig. 5b), in contrast to their original intent. Another problem which appears even if no long flows are present, is the shrinkage of the (flow-level) stability region due to paths spreading over multiple links: a low priority flow will send packets only at times where *all links* in the path are uncongested, and this fraction of time decreases fast with the number of links (see [10]). Again, the resulting excessive number of accumulated low priority flows may affect TCP flows, as in the simulations of Section IV-C.

The above discussion suggests the need to engineer protocols that more aggresively compete with long TCP flows but at the same time have a protective effect on *shorter* TCP flows, which happen to be most of the delay-sensitive flows. In this paper we consider algorithms which minimize the negative effects on short TCP flows while guaranteeing some minimum throughput to background flows, such that their number does not become excessive. These guarantees can be *implicit*, i.e., maintain stability of background flows, or *explicit*, i.e., achieve a given fraction of the excess capacity, possibly higher than the one required for stability. An example of an explicit guarantee is to provide the

[1]Background flow generation is in many cases not elastic since it is automatically generated by machines.

same average throughput as TCP in the long run, as in [11]. This achieves 'incentive compatibility' with the social planner of the ecosystem: it makes originators of the background flows indifferent between adopting a new protocol instead of TCP, while reducing the average delays of delay-sensitive flows compared to the case where the background flows use TCP.

*Main results*

To state our main results it helps to think of the case of a single bottleneck link with capacity $C$ where the internet traffic passing through it (see Fig. 1) is abstracted from all its unnecessary details and assumed to be comprised by

- *TCP traffic - which we do not control*: it consists of i) the long (TCP) flows discussed above, and ii) a stream of TCP flow arrivals, referred to as 'short', with each being a few orders of magnitude shorter[2] than long flows, and which consume an average fraction $\rho$ of link capacity in the long run, hence leaving an excess capacity of $C(1-\rho)$ (on average in the long run) to the rest of the flows.
- *Traffic that we control*: these are the 'Controlled Background Flows (CBFs)', i.e., flows carrying background data whose congestion control protocol we optimize. Each CBF originates at some edge of the network and models either the transfer of a single file of a very large size compared to the sizes of short flows, or a stream of file-transfer requests with sizes comparable to short flows. A natural application of this model is to systems where a level of aggregation is possible, such as file-sharing peers or content servers.

Before we give a summary of the key policies resulting from our analysis as well as our key findings, we note that no application-level information nor any size-based differentiation mechanism is explicitly employed: the differentiation is implicitly performed by exploiting the different timescales in which the numbers of flows of each type evolve.

***The optimal full information policy:*** it minimizes the average short flow (download) delay subject to the CBF traffic obtaining a given fraction (implicit or explicit) $f$ of the excess capacity $C(1-\rho)$. It is of a simple threshold type on the number of short flows in the system. At any time, if the number of short flows is above the threshold, all the capacity goes to the TCP flows (short and long), else it is allocated to the CBF traffic. (See Theorem 1.) Theorem 2 implies the negative impact on the short flows can be arbitrarily large if $f \to 1$: since both long and short flows use TCP, the CBFs cannot squeeze out the former without harming the latter type.

The policy assumes full information on the number of flows passing through the link at each time and so it cannot be implemented by a distributed controller using end-to-end information. Nevertheless it serves as a benchmark (lower bound on download delay) to compare with other more practical polices.

***The optimal policy implementable by congestion feedback:*** it solves the same optimization problem but restricted to a set of policies that use only information available at the edges of the network by reacting to congestion. It has a simple form (see Theorem 3): if link congestion is above some level, the controller of the CBF traffic sends no data; else it sends at a high enough rate to keep congestion at this constant level. This policy performs asymptotically as the optimal full information policy for $\rho \to 1$ (Theorem 4), and numerical analysis shows that it is within few percents of the latter even for smaller values of $\rho$.

If there is a explicit target for the average throughput, our results suggest a simple adaptive algorithm: use the optimal implementable congestion controller while slowly adapt the congestion threshold of the algorithm to achieve the desired throughput, as proposed in [11] and [12].

***Performance of TCP-based congestion controllers.*** A simple policy which directly extends TCP's notion of fairness is $w$TCP: at any time a bandwidth share equivalent to $w$ individual TCP connections is obtained, where the value of $w$ is chosen to achieve the required long-run average share $f$ of the excess capacity. Then the resulting short flow delay is not larger than $20.7\%$ of the optimal under the full information policy (Theorem 5). As the number of long flows increases, the distance from the optimal quickly vanishes and hence the details of the congestion controller become insignificant (e.g., see Fig. 3).

In the case of a CBF comprised *by an arriving stream of file transfer requests*, the following simple form of access control is shown (Theorem 7) to be equivalent to $w$TCP: each arriving file, instead of being immediately transmitted, it is added to a queue from which at most $w$ files are served at any time using TCP. The value of $w$ is chosen so as to produce a critically loaded queue, i.e., long but not unstable.

For values of $f$ near zero or one, $w$TCP's performance converges to the performance of the optimal full information policy (Theorem 5). Hence either when the number of long flows is large or $f$ is extreme, the use of access control described previously is nearly optimal. In contrast, not imposing access control, so that each arriving background file opens a new TCP connection, could more than double the delay of short flows (see Fig. 4).

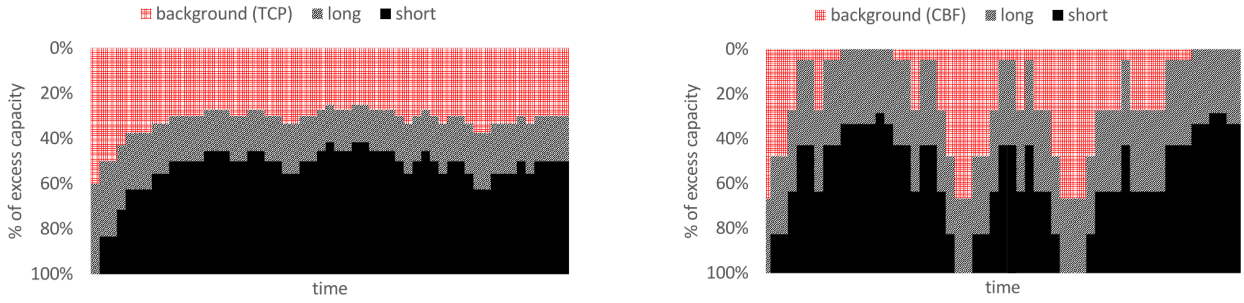
The rest of the paper is organized as follows. In Section II we introduce our system model of short flow arrivals at a single bottleneck link, and establish the optimality of threshold policies under the full information assumption. Bounds and formulas for the minimum delay are given in the case where the offered load of short flows is high. In Section II-C we obtain the optimal policy within the class of policies implementable by congestion feedback, and establish their optimality as $\rho \to 1$. In Section II-D we assess the delay incurred by $w$TCP and compare it with the optimal under full information. In Section III we consider a model in which background flows arrive dynamically and propose an access control policy which limits the maximum number of active background flows. In Section IV we compare the performance of the access control policy with the TCP-LP and LEDBAT LBE protocols. Finally, in Section V we give further justification of our model assumptions as well as discuss a possible extension of our methodology. All proofs are contained in the appendices.

*Related work*

The subject of fairness between different internet flows is intrinsically linked to congestion control and has been studied extensively over the past decades under different perspectives, e.g., see [13] and references therein. The utility-based approach pioneered in [14], [15] paved the way for designing new congestion control algorithms for heterogeneous applications [16] and different notions of fairness [17]–[19].

In [10], [18], [20] the effect of congestion control on the number of ongoing file transfers and download delays is studied.

[2]For example, if long flow sizes are of the order of GB, the sizes of short flow are in the MB range. See also the discussion in Section II-A.




(a) Background flows use TCP.

(b) Background flows use a less aggresive CBF protocol.

Fig. 1. Illustration of bandwidth sharing model: a capacity $C$ link is used by CBF, long TCP flows and a faster varying number of short TCP flows which occupy a fraction $\rho$ of link capacity. As new short flows arrive in the system or old ones complete their transfers, the link capacity is reallocated between all ongoing transfers. Since short and long flows both use TCP, they all obtain equal shares at all times. Background and long flows compete for the excess capacity ($C(1-\rho)$ on average) left over by short flows. In (a) TCP is used for the transport of background files, while in (b) a CBF protocol obtains a lower share during times where more short flows exist in the system. This brings a decrease in the download delays of short flows while the background flows still occupy the same average fraction $f$ of excess capacity as in (a).

We take a similar viewpoint by considering a model where flow-level dynamics are described by a Markovian process, and ignore congestion window dynamics and packet-level effects.

Deb et al. [17] consider a flow-level model of a large system with many long and short flows. They consider the optimization of congestion controllers of all flows -background, long and short- by maximizing a social welfare function which includes the average utility obtained by background traffic and the delay caused to short flows. Since we assume that part of the traffic, namely long and short, uses TCP for its transport and cannot be optimized, the optimal policies differ considerably from the ones in [17].

A model where part of the traffic cannot be optimized is considered in [11] where the notion of *farsighted congestion controllers* for CBF flows is introduced, using a static optimization problem without flow-level dynamics and not involving delays. These controllers implicitly attempt to inflict less delay to short flows but without compromising their average throughput. In this paper we show that these controllers are optimal within the class of policies implemented by state feedback, as they have the same structure as the optimal policy.

## II. Bandwidth sharing for background flows

### A. Basic model

Consider a link of capacity $C$ shared by a set of CBFs, long TCP flows, and a dynamically arriving stream of short TCP flows. The latter concern transfers of files with independent and exponentially distributed file sizes, of mean $\mu^{-1}$, and arrive at the link according to a Poisson process with rate $\lambda$ arrivals per unit time. Because the size of long TCP flows is assumed to be much greater than $\mu^{-1}$, the rate at which flows of this size arrive in the system is orders of magnitude less than $\lambda$. This implies that the timescales in which the number of short and long flows vary are widely disparate. Thus we assume these two timescales are separate and so from the point of view of the short flow timescale, the number of long flows can be taken to be constant and equal to $k$, and they have always data to send. The effect of the long flow dynamics, in the slower timescale, is briefly discussed in Section V.

Here and in the next sections we seek to optimize the bandwidth sharing policies used by the CBFs in order to minimize the delay impact on the short flows. The precise number of CBFs does not matter as we will be optimizing the aggregate behavior; we could as well think of optimizing a single CBF. We do not consider the details of congestion window dynamics, and only model the bandwidth shares after they converge. This simplifying assumption (also used in e.g., [10], [18], [20]) is accurate when the sizes of the short flows are sufficiently larger than the bandwidth delay product, so that download delays are determined mostly by equilibrium shares arising from congestion avoidance rather than the slow start phase. In practice this occurs for a fraction of the TCP flows, which are also the flows that mostly benefit from our control algorithms as seen in Section IV-B. In particular, all TCP flows (whether short or long) are assumed to receive an equal bandwidth share, $x_n$, when the number of active short flows is $n$. Instead of considering the bandwidth share $C-(n+k)x_n$ of CBFs as the controlling variable one can equivalently –as we do– use $x_n$. Thus we assume that $x_n$ can take any value such that the capacity limit is respected, i.e., $x_n \leq C/(k+n)$.

This number $n$ of short flows evolves according to a Markov chain with state space $\{0,1,\dots,\}$ and transition rates:

$$n \to \begin{cases} n+1\,, & \text{with rate } \lambda\,, n \geq 0\,, \\ n-1\,, & \text{with rate } \mu n x_n\,, n \geq 1\,. \end{cases} \tag{1}$$

The *load* $\rho$ is the average fraction $\rho$ of link capacity consumed by short flows, i.e., $\rho = (\lambda/\mu)/C$. Clearly, if $\rho \geq 1$ the Markov chain is not positive recurrent regardless of the choice of $x_n$'s; thus from now on $\rho < 1$ is assumed to always hold. The average capacity $C(1-\rho)$ left over by short flows in the long run, is the *excess capacity* and is consumed (again in the long run) in its entirety by the other flows, i.e., the $k$ long TCP and CBFs.

Now, the choice of $(x_n, n=0,1,\dots)$ determines how capacity is shared between TCP flows and CBFs, since at state $n$ the TCP flows use bandwidth $(k+n)x_n$ while the CBFs consume the remaining $C-(k+n)x_n$. Let $(\pi_n, n=0,1,\dots)$ be the stationary distribution of the Markov chain when it does exist. Then the *average download delay* (or just, delay) experienced by short flows is $\lambda^{-1}\sum_{n=0}^{\infty} n\pi_n$, by Little's law.

For ease of reference, Table I lists the basic model parameters.

TABLE I
BASIC MODEL PARAMETERS AND DEFINITIONS

| | |
|---|---|
| $C$ | link capacity |
| $k$ | number of long TCP flows |
| $\rho$ | load of short (TCP) flows |
| $C(1-\rho)$ | excess capacity |
| $f$ | fraction of the excess capacity consumed by CBF |
| $\lambda$ | arrival rate of short flows |
| $1/\mu$ | average file size of short flows |
| $x_n$ | bandwidth share of each TCP flow when $n$ short flows are present |
| $\pi_n$ | stationary probability of the number of short flows to be $n$ |
| $N_*(k,f,\rho)$ | average number of short flows under the optimal full information policy, see (2). |

### B. Optimal sharing under full information

The problem we solve in this section is the following: *what is the optimal sharing policy* $(x_n, n = 0, 1, \ldots)$ *such that the average delay experienced by short flows is minimized, under the constraint that the CBFs get a fraction* $f$ *of the excess capacity?* Insofar as we only deal with the delay impact on short flows, we do not consider how the $f$ fraction is attributed between CBFs; this path is followed in [12] using a utility maximization framework.

Thus we arrive at the following optimization problem:

$$N_*(k,f,\rho) = \min \sum_{n=0}^{\infty} n\pi_n \tag{2}$$

$$\text{such that: } \lambda\pi_{n-1} = \mu n x_n \pi_n \,, n = 1, 2, \ldots \tag{3}$$

$$\sum_{n=0}^{\infty} \pi_n = 1 \tag{4}$$

$$x_n \le \frac{C}{k+n} \,, n = 0, 1, \ldots \tag{5}$$

$$\sum_{n=0}^{\infty} x_n \pi_n = \frac{C(1-\rho)(1-f)}{k} \tag{6}$$

$$\text{over } x_n \ge 0, 0 \le \pi_n \le 1 \,, n = 0, 1, \ldots \tag{7}$$

Equalities (3) are the local balance equations corresponding to (1), (5) is the link capacity constraint, and (6) is equivalent to the constraint that the CBFs attain the target fraction $f$.

The following theorem states that the optimal policy has a structure of a threshold policy on the number of short flows.

**Theorem 1** (Structure of the optimal policy)**.** *The optimal sharing policy* $(x_n, n = 0, 1, \ldots)$ *satisfies: If* $(1-\rho)^k \le f$ *then*[3]

$$x_n = \begin{cases} 0 \,, & \text{for each } n \le n_* \,, \\ \frac{C}{k+n} & \text{for each } n \ge n_* + 2 \end{cases} \tag{8}$$

*for some finite nonnegative integer* $n_*$.

*If* $(1-\rho)^k > f$*, the CBFs get their target share while* $n = 0$*, so they do not interfere with short flows, i.e., they behave as low priority traffic. More specifically,*

$$x_n = \begin{cases} \frac{C\left[(1-\rho)^k - f\right]}{k(1-\rho)^k} \,, & \text{for } n = 0 \,, \\ \frac{C}{k+n} \,, & \text{for each } n \ge 1 \,. \end{cases} \tag{9}$$

In words, CBFs get the entire capacity at times where the number of flows is no more than $n_*$, while they get zero bandwidth in states strictly greater than $n_* + 1$. Although (8) does not specify $x_{n_*+1}$, it is determined by the requirement that CBFs obtain their target throughput $C(1-\rho)f$ (equivalently (6)), i.e.,

$$\pi_{n_*} C + \pi_{n_*+1} \left[C - (k + n_* + 1) x_{n_*+1}\right] = C(1-\rho)f \,, \tag{10}$$

as CBF flows send traffic only in states $n_*, n_* + 1$. Finally we note that the policy in Theorem 1 is not only optimal with respect to the mean but also with respect to all higher moments[4]; this suggests that the delay distribution has a small tail as well.

Interestingly, the optimal threshold $n_*$ is determined by considering an associated loss system, as described in the following proposition.

**Proposition 1** (Determination of optimal threshold)**.** *The optimal threshold* $n_*$ *satisfies*

$$E\left(n_* + k + 1, k, \frac{1-\rho}{\rho}\right) \le f < E\left(n_* + k + 2, k, \frac{1-\rho}{\rho}\right) \,, \tag{11}$$

*where* $E(m,q,r) = \dfrac{\binom{m-1}{q} r^q}{\sum_{i=0}^{q} \binom{m-1}{i} r^i} \,, m > q \,,$
*is the Engset formula of blocking probability for a loss system with* $q$ *circuits and* $m$ *independent users, each offering traffic equal to* $r$ *Erlangs.*

The mimimum average number of short flows $N_*(k,f,\rho)$ is obtained by the following theorem, which holds for any (not necessarily optimal) threshold policy:

**Theorem 2** (Performance of threshold policies)**.** *Consider any threshold policy with*

$$x_n = \begin{cases} 0 \,, & \text{for each } n \le n_0 \,, \\ \frac{C}{k+n} & \text{for each } n > n_0 \end{cases} \tag{12}$$

*for finite nonnegative integer* $n_0$*. The average number of short flows under this policy at stationarity is*

$$N_{n_0} = \frac{(k+1)\rho}{1-\rho} + n_0 E\left(n_0 + k + 1, k, \frac{1-\rho}{\rho}\right) \,. \tag{13}$$

*In particular, under the optimal policy,* $N_{n_*} \le N_*(k,f,\rho) < N_{n_*+1}$.

As $\rho$ approaches 1, the associated loss system in Proposition 1 is closely approximated by a standard Erlang loss system. This simplifies the determination of both $n_*$ and $N_*(k,f,\rho)$:

**Corollary 1.** *As* $\rho \to 1$ *the optimal threshold* $n_*$ *satisfies* $n_*(1-\rho) \to a_f$*, where* $a_f$ *is the unique solution of* $B(k, a_f) = f$*, and* $B(k,a) = \frac{a^k}{k!}\left(\sum_{i=0}^{k} \frac{a^i}{i!}\right)^{-1}$ *is the call blocking probability given by the Erlang B formula for a system with* $k$ *circuits under a load of* $a$ *Erlangs.*

[3]From Lemma 1 in Appendix A for $n_0 = 0$, $(1-\rho)^{k+1}$ is the proportion of time there are no short flows in the system. Thus $(1-\rho)^k < f$ is equivalent for the CBF target throughput to be greater than the maximum throughput $C(1-\rho)^{k+1}$ that can be obtained while no short flows are in the system. In this case, low priority cannot be the optimal policy for CBF.

[4]This follows from the proof of Lemma 4 in Appendix A which actually establishes domination with respect to convex stochastic order (see [21]) rather than just the mean.

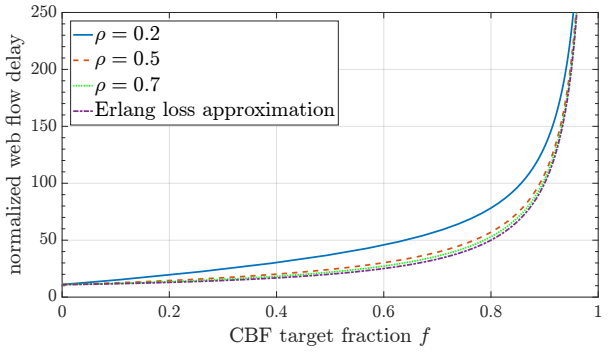



Fig. 2. Minimum delay of short flows (normalized by $1/(\mu-\lambda)$, i.e., the delay in the absence of background and long TCP traffic) under the optimal policy for CBFs, as a function of the fraction $f$ of the excess capacity consumed by CBFs. The link is used also by $k = 10$ long flows. The Erlang loss approximation given in Corollary 1 is close to the minimum delay for $\rho \geq .5$.

*Moreover, the average number of short flows $N_*(k, f, \rho)$ satisfies $N_*(k, f, \rho)(1-\rho) \to k + 1 + a_f f$, as $\rho \to 1$.*

In Fig. 2, $N_*(10, f, \rho)$ is plotted against the target fraction $f$ under various load levels $\rho$, after being normalized by $\rho/(1-\rho)$ (the average number of short flows if background flows were absent). The solid curve is the approximation provided by the Erlang loss system in Corollary 1, which is fairly accurate for $\rho > 0.5$. Notice the sharp increase to $+\infty$ as $f \to 1$: it is inevitable in their competition with long flows for the excess capacity, for CBFs to interfere with short flows as TCP is used by both short and long flows. Larger fractions $f$ of the excess capacity require higher levels of interference.

Suppose one does not use a fixed set of transition rates $(x_n, n = 0, 1, 2, \ldots)$ but is allowed to switch between different policies on a very slow timescale. Can this 'policy switching' result into lower delay? Notice that any policy used in such an optimal mixture must be itself optimal for some level of target excess capacity $f$, i.e., of the form (8); otherwise one could reduce delay by using (8) for target $f$. Thus any optimal mixture of policies can be represented by a probability measure $\Phi$ on the set of target fractions, i.e., the set $[0, 1]$. Let $N(f) = k+1+fa_f$ be the limit (as $\rho \to 1$) of the rescaled average number of short flows from Corollary 1. Then the following holds:

**Proposition 2.** *$N$ is a convex function.*

Applying Jensen's inequality to $N$ yields

$$N\left(\int \phi\Phi(d\phi)\right) \leq \int N(\phi)\Phi(d\phi)\,,$$

and so policy switching has worse delay than the optimal policy (8) of the same target $f = \int \phi\Phi(d\phi)$, at least when $\rho$ is sufficiently close to 1. Fig. 2 suggests that this might not be true only at the limit but it may hold for any value of $\rho$.

### C. Optimal sharing within a class of policies implementable by congestion feedback

Translating a threshold policy into an end-to-end congestion control algorithm is a challenging task because the number $n$ of ongoing short TCP flows is not directly observable, and so it must be inferred through some indirect way. The natural way to do this is through some end-to-end observable measure of congestion such as packet loss and/or delay, which varies monotonically with $n$.

Suppose there exist utility (i.e., increasing and concave) functions $u, v$ for which the maximization problem

$$\begin{aligned} \max \quad & (n+k)u(x) + v(y) \\ \text{such that } & (n+k)x + y \leq C\,, \\ & \text{over } x, y \geq 0\,, \end{aligned} \tag{14}$$

attains its optimum at $x = x_n, y = C - (n+k)x_n$ for every $n > 0$. When such a representation of $(x_n, n = 1, 2, \ldots)$ is possible then $x_n = (u')^{-1}(\lambda_n)$, where $\lambda_n$ is the shadow price of the link capacity constraint. A key insight from [15] is that, in a relaxation of (14), $\lambda_n$ can be interpreted as the rate of congestion indicators fedback by the link to the end users. There it is also shown how the utility functions $u, v$ can be used as a basis for the design of end-to-end algorithms which use the congestion signals sent by the network to attain the optimal solution of (14), i.e., $x_n$, at equilibrium. For this reason, whenever a policy $(x_n, n = 1, 2, \ldots)$ is represented as above, we say that it is *implementable (by congestion feedback)*.

Note in particular that (14) implies that $\lambda_n$ is increasing with $n$, i.e., more congestion signals are sent as $n$ increases since the link is more congested. This implies that $x_n$ is decreasing for implementable policies, but clearly this is not the case for the optimal policy (8). In practical terms, the basic problem with threshold policies is that whenever $n \leq n_*$ and the CBFs need to consume the entire link capacity, the congestion indicator rate must increase considerably in order for TCP flows to drop their congestion windows significantly. As a result, subsequent upcrossings of $n_*$ are difficult to detect on the basis of such congestion indicators alone.

Hence we restrict the search for an optimal policy within the class of implementable policies where we have the following result:

**Theorem 3** (Structure of the optimal implementable policy)**.** *The optimal implementable policy $(x_n, n = 0, 1, \ldots)$ satisfies*

$$x_n = \begin{cases} x_{n-1}\,, & \text{if } n \leq n_*\,, \\ \frac{C}{k+n} & \text{if } n > n_* \end{cases}\,, n = 1, 2, \ldots \tag{15}$$

*for some finite nonnegative integer $n_*$.*

*The policy in* (15) *is represented by* (14) *by choosing $u$ to be any utility function and $v(y) = u'(x_{n_*})y$ for all $y \geq 0$.*

The value $x_{n_*} = \cdots = x_0$ is not determined in closed form but by the requirement that CBFs get throughput $C(1-\rho)f$. There is a unique value $x_{n_*}$ achieving this since the CBF throughput is strictly decreasing in $x_{n_*}$. The implementation of such a policy does not require the knowledge of $u'(x_{n_*})$ as one could start with a utility $v(y) = py$ and adapt $p$ in order for CBF to achieve the target fraction. This approach is followed in [11] where (15) arises in a utility maximization context without short flow arrivals.

Another consequence of implementability is that any such policy is amenable to distributed implementation: it can be effected by each CBF using its own congestion controller, instead of having a single algorithm controlling the aggregate. This is because in the case of $L$ CBFs indexed by $l = 1, \ldots, L$, implementability implies a representation similar to (14) is possible where $y$ and $v(y)$ are replaced by $y_1, \ldots, y_L$ and $\sum_l v_l(y_l)$ respectively. The utility function $v_l$ corresponds to the congestion

controller of the $l$-th CBF. For example, (15) can be represented by $v_l(y_l) = u'(x_{n_*})y_l$. This representation is not unique and different allocations of $y_1, \dots, y_l$ may arise for the same value of $\sum_l y_l$: the choice of $v_l$'s should consider fairness within CBFs which is an interesting problem of further research (see also [12]).

Now, how (15) performs compared to the optimal (8) within the class of all policies (not necessarily implementable)? The following theorem states that the optimal implementable policy has optimal delay scaling as $\rho \to 1$.

**Theorem 4** (Asymptotic optimality of implementable policies)**.** *Let $M_*(k, f, \rho)$ be the average number of short flows under the optimal implementable policy. Then $\lim_{\rho\uparrow 1} M_*(k, f, \rho)/N_*(k, f, \rho) = 1$ for every $k \geq 0, 0 < f < 1$.*

In Fig. 4 the ratio $M_*(1, f, \rho)/N_*(1, f, \rho)$ is plotted for $\rho = 0.5$ and $\rho = 0.9$ The delay of the optimal implementable policy can be up to 22% higher than the optimal for $\rho = 0.5$. For $\rho = 0.9$ the delays of the two algorithms are practically indistinguishable.

### *D. A weighted TCP sharing policy (wTCP)*

In this section we consider an implementable policy which is easier to implement than (15) and can be thought of as a weighted variant of TCP; thus we call it *weighted TCP (wTCP)*. It is appealing because the delay is not much larger than the optimal under full information, when $\rho$ is close to 1.

Under $w$TCP the aggregate of CBFs now takes, at all times, a fixed proportion $w$ of a TCP flow's instantaneous bandwidth. Thus, the CBFs collectively behave as a set of $w$ TCP flows, and $x_n = C/(k + w + n)$ at each state $n \geq 1$. It is easy to see that $w$TCP is implementable by taking $v(y) = wu(y/w)$ in (14). Indeed it has been widely considered in the past (see e.g., [4], [16], [22]) and simple implementations exist through modifying the additive increase and multiplicative decrease parameters of TCP.

If the target CBF ratio is set to $f$, $w$ must satisfy $w/(k+w) = f$. Thus, by Theorem 2 for $n_0 = 0$ and $k + w$ replacing $k$, the resulting average number of short flows is

$$N_{\rm w}(k, f, \rho) = \frac{(k + w + 1)\rho}{1 - \rho} = \frac{\rho}{1 - \rho}\left(k + 1 + \frac{kf}{1 - f}\right). \quad (16)$$

This is related to the optimal $N_*(k, f, \rho)$ as follows:

**Theorem 5** ($w$TCP performance relative to optimal)**.** *The following hold:*

1)
$$\lim_{\rho\to 1} \frac{N_w(1, f, \rho) - N_*(1, f, \rho)}{N_*(1, f, \rho)} = \frac{f(1 - f)}{1 + (1 - f)^2} \leq \frac{3 - 2\sqrt{2}}{2\sqrt{2} - 2} \approx 20.7\% . \quad (17)$$
*The upper bound is tight and is achieved at $f = 2 - \sqrt{2} \approx 60\%$.*

2) *Let $k \geq 2$. Then for every $0 \leq f < 1$:*
$$\lim_{\rho\to 1} \frac{N_w(k, f, \rho) - N_*(k, f, \rho)}{N_*(k, f, \rho)} \leq b_k(f) , \quad (18)$$
$$\text{where } b_k(f) = \frac{B(k - 1, a_f) - f}{1 - [B(k - 1, a_f) - f]} , \quad (19)$$
*with $\sup_{0\leq f<1} b_k(f) < \infty$ decreasing to zero as $k \to \infty$.*

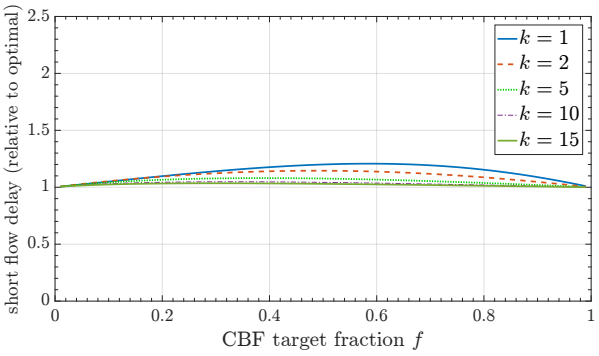



Fig. 3. The delay of short flows caused under the $w$TCP policy for CBFs, normalized by the delay achieved under the optimal full information policy for each level of CBF target fraction $f$. Their maximum difference is 20.7% attained when one long TCP exists in the system. The difference converges to zero as the number $k$ of long flows increases. $w$TCP performs close to optimal for small and large values of $f$.

The theorem states that the relative difference between the delay of $w$TCP and the optimal is bounded. For $k = 1$ the maximum difference is about 20.7%. The second part of Theorem 5 says that the maximum difference goes to zero for large $k$. We stress that this does not need to be the case for *any* CBF policy: there is a policy mixture, of the form considered in the end of section II-B, where the difference grows unboundedly. An example of this is an 'on-off' CBF which half of the time it behaves according to $w$TCP with $w_{\rm on} > 0$, and the remaining time it has $w_{\rm off} = 0$. A similar calculation as above shows that the target ratio $f$ is achieved for $w_{\rm on} = 2kf/(1 - 2f)$. Thus the average number of short flows explodes as $f \to 1/2$, while $N_*(k, 1/2, \rho) < \infty$.

Note that since $b_k(0) = \lim_{f\to 1} b_k(f) = 0$ for $k \geq 2$, $w$TCP is close to the optimal for high and low values of $f$. Fig. 3 depicts the relative difference as a function of $f$ where it is seen to be decreasing in $k$ for most values of $f$. Thus, for practical purposes, $w$TCP appears to be close to the optimal for intermediate values of $f$, even for not so large $k$, e.g., for $k = 5$ the worst difference is 8%.

## III. Dynamically arriving background flows

In the previous sections we used a system model with a fixed number of background flows. This is justified when there is an infinite amount of background data readily available for transfer. In this section we consider a link model as the one in Section II-A but where the CBF is comprised by a stream of 'micro-flows' of finite duration, arriving according to a Poisson process with rate $\lambda_b$. Each micro-flow is associated with the download of a file with an exponentially distributed size with mean $1/\mu_b$. Also the file sizes are assumed to be independent across different flows. If the CBF offered rate $\lambda_b/\mu_b$ is less than the excess capacity and no amount of flow is lost, the fraction of excess capacity consumed by the CBF (or equivalently by its micro-flows) is $f = \lambda_b/[\mu_b C(1 - \rho)]$. We also define the load of the CBF to be $\rho_b = \frac{\lambda_b}{C\mu_b}$.

We allow policies to depend on both the number of short flows and micro-flows, so the state-space is comprised by vectors of the form $(n, m)$ where $n$ is number of short flows and $m$ the number of micro-flows present in the system. As we will see below, the delay of short flows is minimized when the number of micro-flows is a critically stable process, thus we conveniently allow

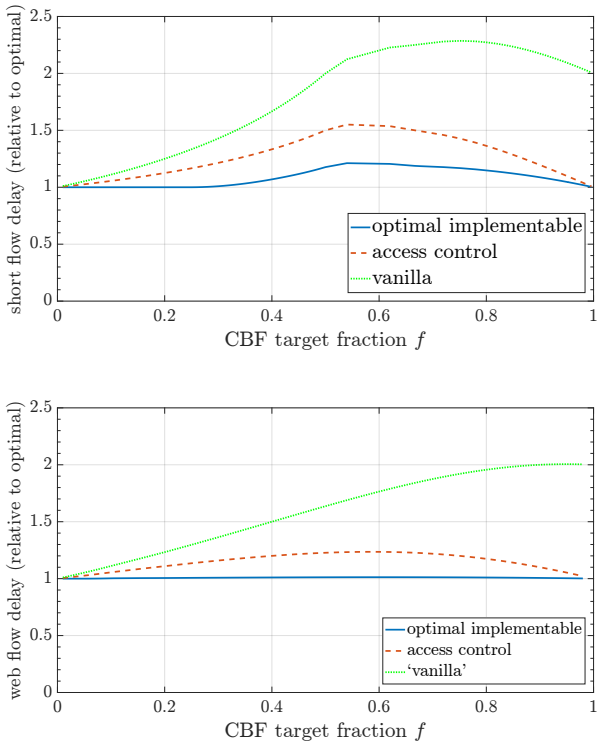



Fig. 4. The short flow delay caused under different CBF policies, at moderate ($\rho = 0.5$ above) and high ($\rho = 0.9$ below) loads, when $k = 1$. The optimal implementable policy practically coincides with the optimal under full information at $\rho = 0.9$. The 'vanilla' policy doubles the delay at $f = 1$ compared to the other two policies which are optimal at $f = 1$.

the possibility $m = \infty$ and consider the extended state-space $\mathcal{S} = \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$.

A policy is specified by the bandwidth $x_{n,m}$ allocated to each TCP flow at every state $(n, m) \in \mathcal{S}$. We will consider policies which satisfy the Feller condition, $x_{n,m} \to x_{n,\infty}$ as $m \to \infty$ for every $n$. This means when the number of micro-flows is high, each TCP flow consumes a well-defined amount. Also notice that when $m = 0$, the link bandwidth is consumed by TCP flows, i.e., necessarily $x_{n,0} = C/(n + k)$ for all $n \geq 0$.

Fixing any such policy yields a Markov process $((N_t, M_t), t \geq 0)$ which evolves according to the transition rates:

$$(n,m) \longrightarrow \begin{cases} (n+1,m) & \text{with rate } \lambda \\ (n-1,m) & \mu n x_{n,m}\ , \text{ if } n > 0 \\ (n,m+1) & \lambda_b\ , \text{ if } 0 \leq m < \infty \\ (n,m-1) & \mu_b\left[C - (n+k)nx_{n,m}\right]\ , \text{ if } 0 < m < \infty \end{cases} \tag{20}$$

### A. Optimal sharing under dynamic arrivals of micro-flows

In this section we solve the following problem which is analogous to to (2)-(7) in Section II-B:

$$N_{**}(k,\rho) = \min \sum_{(n,m)\in\mathcal{S}} n\pi_{n,m} \tag{21}$$

$$\text{such that: } (\pi_{n,m}, (n,m) \in \mathcal{S}) \tag{22}$$

$$\text{is a stationary distribution of (20)}$$

$$x_{n,m} \leq \frac{C}{k+n}\,, n \geq 0,\ 1 \leq m \leq \infty \tag{23}$$

$$x_{n,0} = \frac{C}{k+n}\,, n \geq 0 \tag{24}$$

$$\sum_{(n,m)\in\mathcal{S}} \left[C - (n+k)x_{n,m}\right]\pi_{n,m} = \frac{\lambda_b}{\mu_b}\ , \tag{25}$$

over $x_{n,m} \geq 0, 0 \leq \pi_{n,m} \leq 1, (n,m) \in \mathcal{S}$. Note that the summations in (21),(25) include the points at infinity $m = \infty$. The constraint (25) says that the CBF throughput equals its load, i.e., no amount of flow is lost.

It turns out that the minimum delay is the same with that achieved in the case of a CBF with always data to send, i.e., the optimal value (21) coincides with (2).

**Theorem 6.** *$N_{**}(k,\rho) = N_*\left(k, \frac{\rho_b}{1-\rho}, \rho\right)$ and an optimal policy for (21) is $x_{n,m} = x_n$ for every $n \geq 0, m > 0$ including $m = \infty$, where $(x_n, n \in \mathbb{N})$ is the optimal policy (8) for $f = \frac{\rho_b}{1-\rho}$.*

For the same reasons outlined in subsection II-C it is not easy to implement the optimal policy. Hence we consider a suboptimal but simple policy which controls the access of micro-flows which we describe next.

### B. An access control policy for micro-flows

In Theorem 6 is shown that the optimal policy in the model with arrivals achieves the same delay for the short flows as the optimal one *without* arrivals. In this section we show that a similar result (Theorem 7 below) holds for $w$TCP: the delay of the simple access control policy defined below, coincides with the delay of $w$TCP in a model without arrivals. In particular, this and Theorem 6 imply that the delay induced to the short flows by the access control policy is never more than 20.7% from the optimum (in the case of arrivals).

Consider a CBF policy controlling the access of micro-flows into the network in which no more than $M$ *active* micro-flows are allowed to transmit[5], for some constant $M > 0$. Once an (active) micro-flow completes its download, a previously inactive flow (provided there is one) becomes now active. Hence the number of active micro-flows carried by that CBF is $\min(m, M)$ when there is a total of $m$ micro-flows (both active and inactive). Each micro-flow once active it uses TCP for its transmission. Thus, since the link capacity is divided equally between all TCP and the active micro-flows, we have

$$x_{n,m} = \frac{C}{n + k + \min(m, M)}\ , \text{ at any state } (n,m) \in \mathcal{S}\ . \tag{26}$$

The number of active micro-flows is always at or below $M$, so the CBF obtains at most a $M/(k+M)$ fraction of the excess capacity as the result of its competition with the $k$ long flows which also use TCP. Choosing $M$ to be too low may result to a throughput which is strictly lower than the load $C\rho_b$ brought by the CBF. In this case the number of micro-flows will increase arbitrarily without though causing an arbitrary degradation to the delay of short flows. This is because the micro-flows are kept *outside* of the network until they get to transmit. Choosing $M$ to be too high will result to a stable number of micro-flows, and so their throughput equals $C\rho_b$, but at the cost of a higher delay caused to short flows. This delay may be uneccesarily high if stability holds for even lower values of $M$. Thus $M$ should be chosen such that the number of micro-flows is barely stable. Because the number of micro-flows will be much larger than $M$, most of the time there will be exactly $M$ micro-flows transmitting. Hence the CBF will behave as a set of $M$ TCP flows.

[5]The number $M$ is CBF specific and in general it differs across CBFs.

The above discussion is formalized in the following result:

**Theorem 7.** *Under the policy* (26)*, where* $M$ *satisfies*

$$\frac{\rho_b}{1-\rho} = \frac{M}{k+M}\,, \tag{27}$$

*the average number of short flows is the same as under a single* $w$*TCP flow with weight* $M$ *(or equivalently, the same under a CBF comprised by* $M$ *TCP flows), i.e.,* $N_w\left(k, \frac{M}{M+k}, \rho\right)$ *as defined in* (16)*.*

In practice, (27) can be made to approximately hold by slowly adjusting $M$ such that long micro-flows queues are maintained. The longer the queues are the closer $M$ gets to the critical value (27).

## IV. Access control versus Less than Best Effort protocols

Consider a file server which transmits background content, e.g., a server distributing software updates after requests sent by users of an application. All requests should be served eventually but in such a way so delay sensitive flows in the network see a mimimal disruption in their download delays. Theorem 7 in Section III-B suggests that the access control policy which simply limits the maximum number $M$ of active connections is not far (less than 20.7% away) from the minimum delay achieved by any control policy that the server could use.

In this section we compare the performance of such a policy with the alternative of using LBE protocols such as LEDBAT and TCP-LP, using simulation. As Theorem 5 implies access control is optimal for $f$ near 0 or 1, $k \geq 1$ and high $\rho$, we expect this policy to perform better than LBE for these values.

Before we proceed we first describe the implementation of the access control policy used in the simulations. As explained in Section III-B the access control policy should pick the least $M$ such that (27) holds. Since the link and traffic parameters are not known by the server, $M$ is calculated adaptively such that the average sending rate (in e.g., Mbps) matches the arrival load $C\rho_b$. Both the load and sending rate are estimated using moving averages with sufficiently long memory such that the effect of background micro-flow arrivals and departures is averaged out. Since the sending rate is increasing in $M$, the latter is increased or decreased such that the sending rate tracks the arrival load estimate. Thus $M$ changes on a slower timescale than the dynamics of arrivals and departures.

We next compare the performance of the access control policy with that of other LBE protocols in the case of a single bottleneck link and also in the case of simple networks. Contrary to what one might have guessed, the LBE protocols can be quite intrusive especially under a high load, the presence of long TCP flows, or multiple traversed links. Access control on the other hand, leaves flows outside of the network when congested, and so performs better in high loads.

### A. Experimental setup

The simulations are performed in ns2 [23] where TCP flows use Reno, TCP-LP flows use the implementation provided in ns2, and LEDBAT flows use the implementation in [24]. The latter flows use either the default 25ms target delay or a more 'aggressive' 60ms value (labeled as LEDBAT-60 below). For comparison, we also consider a 'vanilla' server policy where no form of access control or LBE protocol is used, i.e., each request upon arriving at the server it initiates a file transfer which uses TCP.

All links have $C = 10$Mbps capacity, a 80ms buffer and 25ms propagation delay. Flow arrivals, whether background or not, follow independent Poisson processes, while the sizes of all files (again, background or not) are exponentially distributed with mean 3MB.

We consider three linear network topologies described next.

### B. Single link

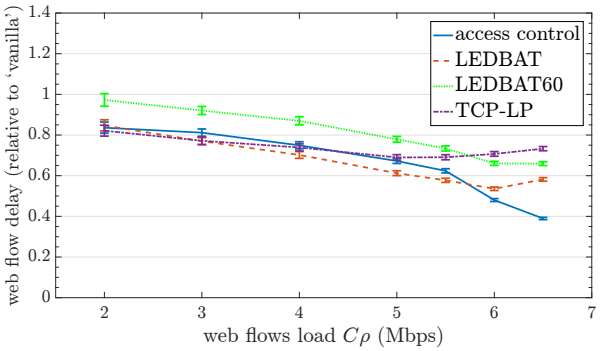
The rate $C\rho_b$ of background traffic brought to the server by the requests is 3Mbps. In the top row of Fig. 5 the delay of short flows is depicted for various levels of load $\rho$. The delay is normalized by the delay of the 'vanilla' policy for each level of $\rho$. In Fig. 5a, where no long flows exist, i.e., $k = 0$, the delay under access control is comparable to LBE protocols. In high loads CBF performs significantly better: for $\rho = 0.5$ the difference is 20%. As expected, the delay of LBE protocols is always less than the 'vanilla' policy. Interestingly, this is not the case if even a single long flow is added as in Fig. 5b: above $\rho = 0.6$ LBE protocols cause worse delays than the 'vanilla' policy.

To see what happens, consider the normalized average number of *active* micro-flows when no long flows are present, depicted in Fig. 5c. (Again, the normalization is done with respect to the number of active micro-flows under the 'vanilla' policy for the same load $\rho$.) Since the LBE flows behave as 'low priority' traffic yielding to competing TCP flows, more LBE flows are squeezed out of the link as it becomes more congested. As a result a greater number of active background downloads is observed –except for TCP-LP above $\rho = 0.5$–.

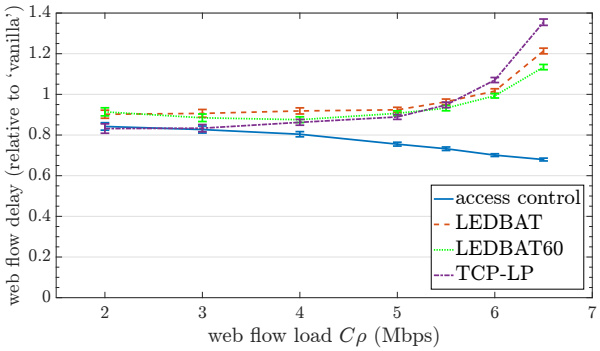
Under the presence of a single long flow, in Fig. 5d, the number of background micro-flows (relative to 'vanilla') has a decreasing trend. This means the LBE flows do not yield as much as when $k = 0$, as the load increases. This is because their absolute number has increased considerably and it has reached a point where the LBE flow throughput cannot be compressed further since each such flow sends at least one packet per round-trip-time (provided no loss occurs). Thus, aggregate LBE traffic essentially stops behaving as low priority and competes more equally with TCP. This has a damaging effect on the delay of short flows as the traffic composition now contains more aggressive flows.

On the other hand, the delay caused by access control continues to decrease (relative to 'vanilla') even after the addition of the long flow. This is because the number of active micro-flows does not increase as much as the total number of micro-flows in the system. Hence the congestion of micro-flows does not spill over to short flows.

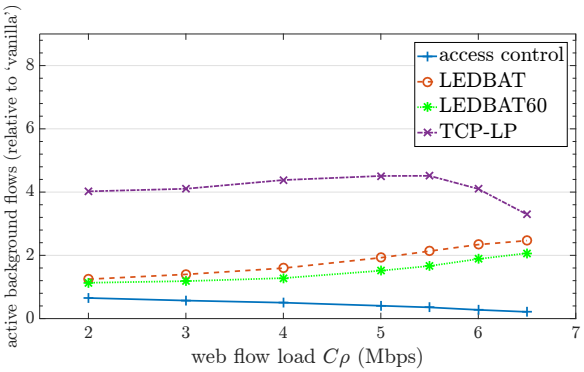
In Fig. 6 the delay experienced by short flows with respect to their size is shown, where $k = 1$. For $\rho = 0.2$, access control has similar performance to LEDBAT except in very short files (less than 100kB). For $\rho = 0.6$, access control performs significantly better or similar to LBE protocols over all flow sizes except short ones (less than 100kB), where the latter might outperform access control. In particular LEDBAT always resulted in smaller delay for flow sizes less than 100kB. But LBE protocols can be very harmful even compared to vanilla when $\rho = 0.6$ for sizes bigger than 1MB. The access control policy performs always better than the 'vanilla policy for all flow sizes.



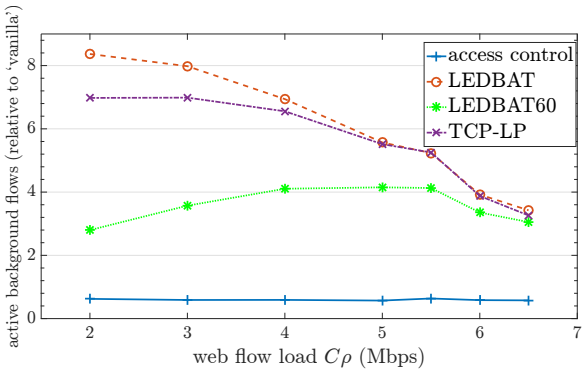

(a) Normalized delay, no long flows present ($k = 0$)




(b) Normalized delay, $k = 1$




(c) Normalized active micro-flows, $k = 0$




(d) Normalized active micro-flows, $k = 1$

Fig. 5. *Single link network:* A comparison of access control and LBE protocols for a CBF comprised by a stream of dynamically arriving micro-flows. The top row depicts the short flow delay for each protocol normalized by the delay caused by the 'vanilla' policy for the same level of load. The presence of a single long flow (in (b)) causes LBE protocols to perform worse than the 'vanilla' policy at high loads. The bottom row depicts the average number of active micro-flows normalized by the corresponding number under the 'vanilla' policy for the same level of load.

### C. Two link network

Here we consider a network with two links of equal capacity $C = 10$Mbps without any long flows ($k = 0$), in which $\rho_b = 0.3$ again but now the background flows traverse both links. In each link there is also a separate stream of short flows of load $\rho$.

Fig. 7 depicts the short flow delay and number of active micro-flows for different load levels $\rho$. In Fig. 7a the delay under access control is similar to LEDBAT for loads less than $\rho = 0.5$. Above this value a similar effect to that when long flows are present occurs: in Fig. 7b the number of LEDBAT flows decreases relative to the 'vanilla' policy, i.e., they become more aggressive. The reason is that starvation effects are expected to occur for loads $\rho_b > (1-\rho)^2$, i.e., $\rho > 0.46$, as truly low priority flows would be able to push data through the two links only at times where no short flows are present, i.e., a proportion $(1-\rho)^2$ of time (see [10]). This explains the peak in the relative number of LEDBAT flows after $\rho = 0.5$ shown in Fig. 7b. Starvation in practice means the number of background flows will increase significantly as in the case with long flows. Again since the LEDBAT flow throughput cannot be compressed below a certain lower limit, LEDBAT stops behaving as low priority traffic and increases delay significantly. The delay reduction due to LEDBAT relative to the 'vanilla' policy is only 10% at 95% total load.

This is contrasted with access control where the reduction is 28% for the same load. As in the single-link case, this is because under access control the number of background flows does not affect congestion because only a fraction of micro-flows is active.

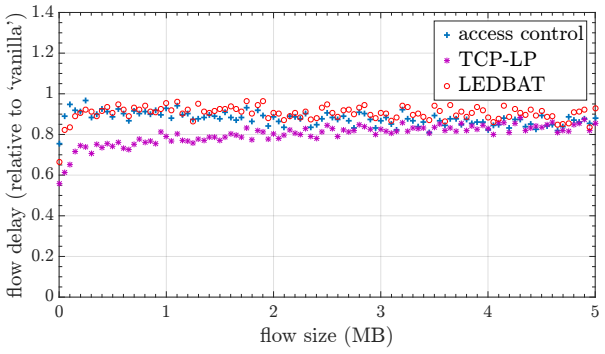
### D. Three links

Here we consider the topology in Fig. 8 where there are four routes labelled `routeX`, where `X` $\in \{1, 123, 2, 23\}$ with `route1` spanning only link 1, `route123` spanning links 1,2,3 etc. `route1` is only used by a stream of short flows with load $\rho_1$. On each of `route123`, `route2`, and `route23` there is a CBF (carrying its own stream of micro-flows) with load $\rho_{123} = 0.2, \rho_2 = 0.1, \rho_{23} = 0.1$, respectively. These three routes also carry streams of short flows with the same load as the CBFs on the same routes.

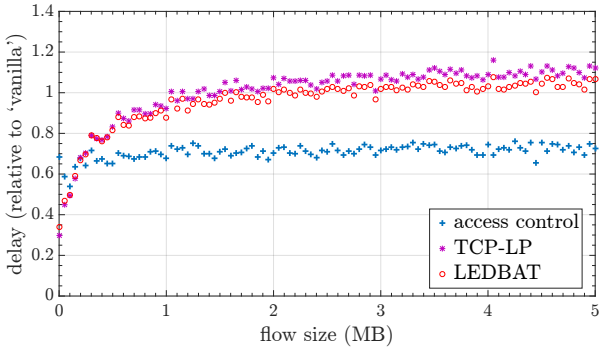
Fig. 9 depicts the normalized delay of short flows on each of the four routes for the choices $\rho_1 = 0.3$ and $\rho_1 = 0.5$. Even though link 1 is not excessively loaded (70%), the LBE protocols do not seem to bring any significant delay savings compared to the 'vanilla' policy. In fact TCP-LP results to higher delays. Access control causes less delays for the short flows on any route, except `route123` and `route23` where it is close to LEDBAT which has the lowest delay there. The difference between access control and the LBE protocols becomes more significant for $\rho = 0.5$ over the routes that have highly loaded links, i.e., `route1` and `route123`.

## V. Discussion

The timescale separation assumption (in Section II-A) that the number $k$ of long TCP flows is constant and they never stop sending, is essential in obtaining the results of this paper. However as $k$ changes in reality, we expect $k = 0$ to occur if the total offered load is below capacity. But then why would not low

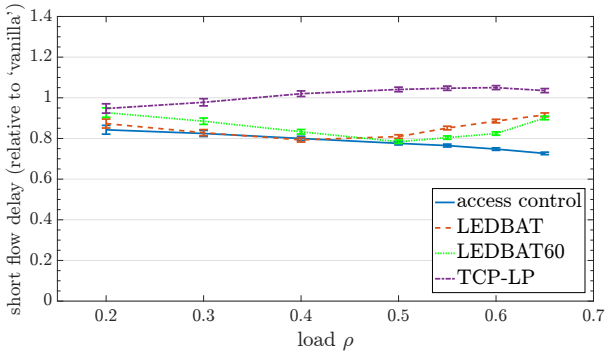



(a) $\rho = 0.2$

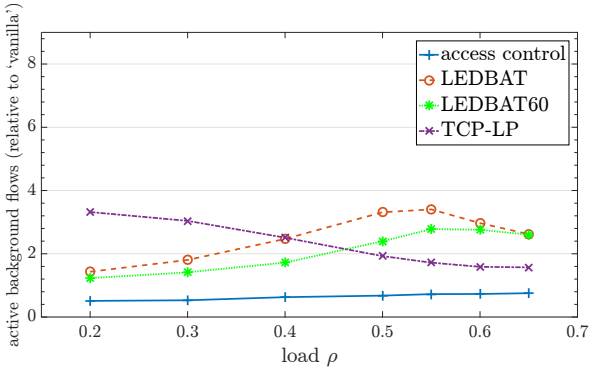



(b) $\rho = 0.6$

Fig. 6. A comparison with 'vanilla' of access control and LBE protocols when $k = 1$ for a CBF comprised by a stream of dynamically arriving micro-flows. Short flows are classified into 100 bins of multiples of 50kB according to their sizes. Each point shows the average delay within the respective bin staring from $[0, 50kB)$, divided by the average delay of the same bin under the 'vanilla' policy. (We have omitted the results for LEDBAT60 because they are very similar to LEDBAT.)




(a)




(b)

Fig. 7. *Two link network*: comparison with 'vanilla' of access control LBE protocols for a CBF comprised by a stream of dynamically arriving micro-flows for short flow file sizes of 3MB. LBE starvation effects occur at $\rho \geq 0.46$ even without the presence of long flows. (a) Short flow delay for each protocol normalized by the delay caused by the 'vanilla' policy for the same level of load. (b) The average number of active micro-flows normalized by the corresponding number under the 'vanilla' policy for the same level of load.

priority CBFs be optimal, as they can send only during $k = 0$ and no short flows are in the system (i.e., $n = 0$ in the notation of Section II-A) and thus not interfere at all with short flows? Such an ideal low priority policy is indeed optimal if the evolution of $k$ is considered. But as we saw in the simulations of the LBE protocols in Section IV, ideal prioritization is not feasible, especially in highly loaded links. In particular, LBE protocols obtain a nonnegligible throughput if $k = 1$, whereas under ideal priority it ought to be zero. In this paper we choose to accept that for each $k$, CBFs unavoidably obtain a nonzero throughput (either by design or not) and so consider controllers which explicitly minimize their impact for each level of obtained throughput.

In the experiments of Section IV we saw that this more pragmatic design goal yields simple controllers which cause lower delays to short flows than LBE protocols, in most cases and especially under high load and if flow sizes are not too short.

Observe that the optimal parameter value for each controller ($n^*$ in (8) and (15), $w$ in (16), and $M$ in (27)) cannot be computed in closed form during operation, as $k$ and the fraction $f$ of excess capacity obtained by background flows are not known in general. Instead, as the parameter is a monotonic function of throughput, one could adaptively search for the appropriate value at which the resulting throughput is the desired one. This suggests that a good CBF controller consists of the i) *fast timescale congestion control* that deals with how the protocol responds to congestion in the fast timescale that determines the instantaneous capacity share (which can be based on the policies of this paper), and ii) *slow timescale feedback control*, that looks at the average throughput obtained over this timescale and adapts the parameter supplied to the fast timescale congestion controller.

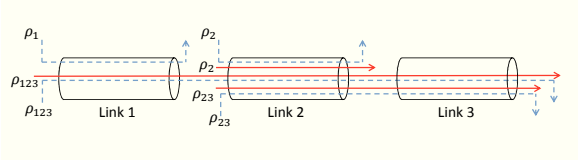



Fig. 8. The three link network simulated in Section IV-D under two different loads ($\rho_1 = 0.3$ and $0.5$) for the short flows traversing only link 1.

Finally, one possible extension of this work is to explore system objectives which include the performance of background flows, such as sums of utilities as in [11], [12], utility rates as in [17], or their delay, in addition to the delay of short flows. It is likely that the policies in this paper may serve as building blocks for the fast timescale congestion control part in these more general controllers.

## ACKNOWLEDGMENT

The work of A. Dimakis was supported by European Union (European Social Fund - ESF) and Greek national funds through the Operational Program Education and Lifelong Learning of the National Strategic Reference Framework through the Research Funding Program Thales - Investing in knowledge society through the European Social Fund.

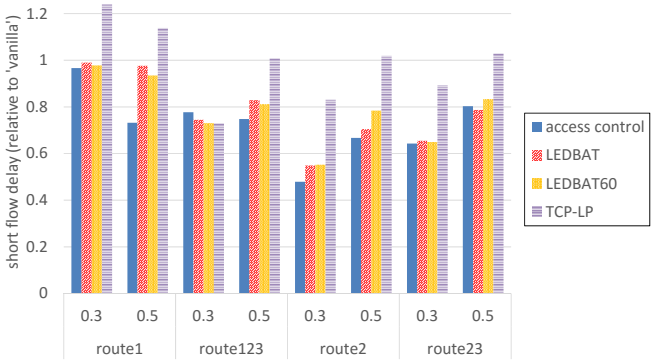



Fig. 9. *Three link network:* A comparison with 'vanilla' of access control and LBE protocols for a CBF comprised by an arriving stream of micro-flows. The short flow delay (normalized by the delay caused by the 'vanilla' policy) for each route and each value for $\rho_1 \in \{0.3, 0.5\}$. The access control policy performs better than LBE when the (short flow) load in link 1 is increased to $\rho_1 = 0.5$. It does not cause significantly worse delay to short flows not passing through link 1 (i.e., on `route2` and `route23`).

## APPENDIX

### A. Auxilliary results

Here we establish a series of lemmas along with Theorem 8 which lies at the core of two main results, Theorems 1 and 3.

**Lemma 1.** *For the threshold policy* (2)*, if* $\rho < 1$ *the stationary distribution of the number of short flows* $n$ *is given by*

$$\pi_n = \begin{cases} \frac{\binom{n+k}{k}(1-\rho)^{k+1}\rho^n}{\sum_{i=0}^{k}\binom{n_0+k}{i}(1-\rho)^i\rho^{n_0+k-i}}\,, & n \geq n_0\,, \\ 0\,, & n < n_0\,. \end{cases} \tag{28}$$

*In particular,* $\pi_n = P(X = n | X \geq n_0)$*, where* $X$ *is the sum of* $k+1$ *independent geometric random variables, each with 'success' probability* $1-\rho$*.*

*Proof:* Let $X$ be the sum of $k+1$ independent geometric distributions each with 'success' probability $1-\rho$. Its distribution is readily shown to satisfy the detailed balance equations

$$P(X = n)\rho = P(X = n+1)\frac{n+1}{n+1+k}\,, n = 0, 1, \ldots, \tag{29}$$

which correspond to a system with the threshold set to zero. Since the Markov chain is reversible, the stationary distribution for the case $n_0 > 0$ is $\pi_n = P(X = n | X \geq n_0)$, i.e.,

$$\pi_n = \frac{\binom{n+k}{k}(1-\rho)^{k+1}\rho^n}{\sum_{i=0}^{k}\binom{n_0+k}{i}(1-\rho)^i\rho^{n_0+k-i}}\,, n = n_0, n_0+1, \ldots \tag{30}$$

and 0 for $n < n_0$, where we have used the fact that the event $X \geq n_0$ corresponds to the occurence of at most $k$ 'successes' in a sequence of $n_0 + k$ Bernoulli trials. ∎

**Lemma 2.** *If* $(1-\rho)^k \leq f$ *and* $(x_n, n = 0, 1, \ldots)$ *defined as in Theorem 1, then* $x_0 = 0$*.*

*Proof:* Assume first that $x_n = \frac{C}{k+n}$ for all $n \geq 1$. Then the stationary distribution is given by Lemma 1 for $n_0 = 0$, and

$$\pi_0 x_0 + \sum_{n=1}^{\infty}\pi_n x_n = (1-\rho)^{k+1}x_0 + \frac{C(1-\rho) - C(1-\rho)^{k+1}}{k}\,. \tag{31}$$

Plugging this into (6) yields $f = (1-\rho)^k(1 - kx_0/C)$ which it does not hold unless $x_0 = 0$.

Now assume that there exists $n \geq 1$ for which the inequality in (5) is strict. If $x_0 > 0$ then we could decrease $x_0$ and increase $x_n$ such that (6) remains true. Notice though, that the increase of $x_n$ does decrease the average number of short flows, while the increase of $x_0$ does not have any effect whatsoever. Thus, the optimal allocation $x_0$ must be zero. ∎

**Lemma 3.** *For each* $n \geq 0$ *and* $x_n, \pi_n$ *which satisfy* (3)-(7)*,*

*define* $\bar{\pi}_n = x_n\pi_n\frac{k}{C(1-\rho)(1-f)}\,, y_{n+1} = \frac{\bar{\pi}_n}{\bar{\pi}_{n+1}}\,.$ (32)

*The following hold:*

$$\sum_{n=0}^{\infty}\bar{\pi}_n = 1\,, \tag{33}$$

$$y_{n+1} \leq \frac{n+1}{\rho(k+n)}\,, n \geq 0\,, \tag{34}$$

$$\sum_{n=0}^{\infty} n\bar{\pi}_n = \frac{\rho k}{(1-\rho)(1-f)}\,, \tag{35}$$

$$0 \leq \bar{\pi}_n \leq 1\,, n = 0, 1, \ldots \tag{36}$$

*Conversely, for any* $\bar{\pi}_n, y_{n+1}, n = 0, 1, \ldots$ *satisfying* (33)-(36) *there exist unique* $\pi_n, x_n, n = 0, 1, \ldots$ *for which* (3)-(7) *and* (32) *hold.*

*Proof:* Summing (32) over $n = 0, 1, \ldots$ and using (6) yields (33). (34) follows by mutliplying both sides of (5) by $\pi_n$ and utilizing (3),(32). Since $\rho < 1$, the average rate of departures must equal $\lambda$, that is,

$$\sum_{n=0}^{\infty}\mu n x_n\pi_n = \lambda \Leftrightarrow \sum_{n=0}^{\infty} n\bar{\pi}_n = \frac{\rho k}{(1-\rho)(1-f)}\,, \tag{37}$$

which proves (35).

To show the converse part, define

$$\pi_n = \frac{(n+1)(1-\rho)(1-f)}{\rho k}\bar{\pi}_{n+1}$$

$$\text{and } x_n = \begin{cases} \frac{C\rho y_{n+1}}{n+1}\,, & \text{if } \bar{\pi}_{n+1} > 0\,, \\ 0\,, & \text{if } \bar{\pi}_{n+1} = 0\,, \end{cases} \text{ for each } n = 0, 1, \ldots \tag{38}$$

Now, (3) follows by noting that $\frac{ny_{n+1}}{n+1} = \frac{\pi_{n-1}}{\pi_n}$ and substitution in the definition of $x_n$ above. Also, (4) follows by (35). (5) follows by (34) and the definition of $x_n$ above, while (6) follows directly from the definition of $\pi_n, x_n$.

This deals with existence; to establish uniqueness note that any collection of $\pi_n, x_n, n = 0, 1, \ldots$ which satisfy (32) and (3) defines $\pi_n$ uniquely by the latter equation. Thus, $x_n$ is defined uniquely by (32) for every $n \geq 1$, while $x_0$ is determined by (6). ∎

Observe that the second equation in (32) and (33) imply that $(\bar{\pi}_n, n \geq 0)$ can be interpreted as the stationary distribution of a birth-death chain with unit birth rate and death rate $y_n$ in state $n \geq 1$.

**Lemma 4.** *Let* $(y_{n+1}, \bar{\pi}_n, n \geq 0)$ *and* $(y'_{n+1}, \bar{\pi}'_n, n \geq 0)$ *be two sets of death rates and their corresponding stationary distributions, so both sets satisfy* (33)-(36) *and the second equation in* (32)*.*

*If $y_{m+1} < y'_{m+1}, y_n = y'_n$ for all $n \notin \{m, m+1\}$ for some $m \geq 1$, then $\sum_n n^2 \bar{\pi}'_n \leq \sum_n n^2 \bar{\pi}_n$.*

*Proof:* We show that $(\bar{\pi}_n, n \geq 0)$ dominates $(\bar{\pi}', n \geq 0)$ in the convex stochastic order which by definition (e.g., see 3.A.1 in [21]) then implies $\sum_n n^2 \bar{\pi}'_n \leq \sum_n n^2 \bar{\pi}_n$.

First note that $\text{sgn}(\bar{\pi}_n - \bar{\pi}'_n) = \text{sgn}(\bar{\pi}_0 - \bar{\pi}'_0)$[6] for all $n < m$ since $y_n = y'_n$ in that range. Similarly $\text{sgn}(\bar{\pi}_n - \bar{\pi}'_n) = \text{sgn}(\bar{\pi}_{m+1} - \bar{\pi}'_{m+1})$ for all $n > m$ is true. Now

$$\begin{aligned} \text{sgn}(\bar{\pi}_m - \bar{\pi}'_m) &= \text{sgn}\left(\bar{\pi}_{m-1} y_m^{-1} - \bar{\pi}'_{m-1} {y'_m}^{-1}\right) \\ &\leq \text{sgn}\left(\bar{\pi}_{m-1} - \bar{\pi}'_{m-1}\right) , \end{aligned} \tag{39}$$

since $y'_m \leq y_m$ which follows from the assumption that $(\bar{\pi}_n)$ and $(\bar{\pi}'_n)$ have the same mean and $y'_{m+1} \geq y_{m+1}$. In turn this implies $\text{sgn}(\bar{\pi}_m - \bar{\pi}'_m) \leq \text{sgn}(\bar{\pi}_{m+1} - \bar{\pi}'_{m+1})$. Thus, $\bar{\pi}_n - \bar{\pi}'_n$ can change sign at most twice as $n$ goes from 0 to $\infty$. It is easy to see that the distributions $(\bar{\pi}_n)$ and $(\bar{\pi}'_n)$ are not stochastically ordered so Theorem 1.A.12 in [21] implies that $\bar{\pi}_n - \bar{\pi}'_n$ cannot change sign only once. Thus there are exactly two sign changes and so by Theorem 3.A.57, $(\bar{\pi}_n)$ dominates $(\bar{\pi}'_n)$ in the convex stochastic order. ∎

**Theorem 8.** *Consider the optimization problem:*

$$\textit{Minimize } \sum_{n=0}^{\infty} n^2 \bar{\pi}_n \textit{ over } \bar{\pi}_n, y_n, n \geq 0$$
$$\textit{such that (33)-(36) and the second equation in (32) hold.} \tag{40}$$

*The optimal solution satisfies:*

$$y_n = \begin{cases} \frac{n}{\rho(k+n-1)} & n > m \\ 0 & n < m \end{cases}, n = 1, 2, \dots , \tag{41}$$

*for some $m \geq 1$.*

*Under the additional constraints*

$$\frac{y_{n+1}}{n+1} \leq \frac{y_n}{n}, n = 1, 2, \dots , \tag{42}$$

*the optimal solution satisfies*

$$y_n = \begin{cases} \frac{n}{\rho(k+n-1)} & n \geq m \\ \frac{n y_{n-1}}{n-1} & 1 \leq n < m \end{cases} , \tag{43}$$

*for some $m \geq 1$.*

*Proof:* Let $(y_n, n \geq 1)$ be the optimal solution of (40) and suppose $y_{m+1} < (m+1)/(\rho(k+m))$ and $y_m > 0$ for some $m \geq 1$. Then there exist transition rates $(y'_n, n \geq 1)$ with $y_{m+1} < y'_{m+1} \leq (m+1)/(\rho(k+m))$, $y_m > y'_m \geq 0$ $y'_n = y_n$ for all $n \notin \{m, m+1\}$ and for which the corresponding stationary distribution $(\bar{\pi}'_n)$ still satisfies (35). Then Lemma 4 implies that $y$ is not optimal and we arrive at a contradiction. Thus either $y_{n+1} = (n+1)/(\rho(k+n))$ or $y_n = 0$ for all $n \geq 1$, which in turn implies (41).

If (42) is required then the same reasoning implies that $y_{n+1} = (n+1)/(\rho(k+n))$ or $y_n/n = y_{n+1}/(n+1)$ holds for all $n \geq 1$. Notice that if $y_n = n/(\rho(k+n-1))$ then $(n+1)y_n/n = (n+1)/(\rho(k+n-1)) > (n+1)/(\rho(k+n)) \geq y_{n+1}$, and so $y_{n+1} = (n+1)/(\rho(k+n))$ must be true. This proves (43). ∎

[6] $\text{sgn}(x)$ denotes the sign function: it takes the values -1,0,1 if $x < 0$, $x = 0$, or $x > 0$ respectively.

### B. Proof of Theorem 1

Consider the case $(1-\rho)^k > f$ first. Since by Lemma 1 for $n_0 = 0$, $(1-\rho)^{k+1}$ is the proportion of time there are no short flows in the system, CBFs do not affect the dynamics of the Markov chain as they can obtain the target fraction $f$ during the absence of short flows. Hence the average number of short flows is minimized by setting $x_n$ at its maximum value $\frac{C}{k+n}$ for each $n \geq 1$. (Notice also that when $x_0$ is as given in (9), the average throughput constraint (6) is satisfied.)

Now consider the case $(1-\rho)^k \leq f$. The key idea is to consider an alternative representation of the optimization problem (2)-(7) over the variables $\bar{\pi}_n, y_n, n \geq 0$ defined by (32). By Lemma (3) the constraints (3)-(7) correspond to (22)-(25) under the new representation. Also multiplying both sides of (3) by $n-1$ and summing over $n = 1, 2, \dots$ yields

$$\lambda \sum_{n=1}^{\infty} (n-1)\pi_{n-1} = \frac{\mu C(1-\rho)(1-f)}{k} \sum_{n=1}^{\infty} n^2 \bar{\pi}_n - \lambda , \tag{44}$$

where we have used (35) in Lemma (3). Thus, the minimization of the objective in (2) is equivalent to that of the second moment of the distribution $\bar{\pi}_n, n = 0, 1, \dots$ under the new representation.

The optimal solution for $y_n, n \geq 1$ for this problem is given by (41) in Theorem 8. To translate into a policy for the initial problem, use Lemma 3 to obtain unique $x_n, n \geq 0$ given by (38) in terms of the $y_n, n \geq 1$. This yields (8) by defining $n_* = m-2$, where $m$ is as in Theorem 8.

### C. Proof of Proposition 1

The proof follows by the observation that the threshold policy with the threshold set at $n_0$ with $x_{n_0} = 0$ obtains a fraction $f = \frac{\pi_{n_0}}{1-\rho} = E\left(n_0 + k + 1, k, \frac{1-\rho}{\rho}\right)$ of the excess capacity, by Lemma 1. See [12] for more details.

### D. Proof of Theorem 2

It follows by direct calculations using the stationary distribution of short flows in Lemma 1. See [12] for more details.

### E. Proof of Corollary 1

Consider a sequence of threshold policies indexed by $\rho$ for which the threshold level $n_\rho$ satisfies $n_\rho(1-\rho) \to a$ as $\rho \to 1$. Since the number of users $n_\rho + k + 1$, in the loss system described in Proposition 1, grows with $\rho$ while the total load converges to $a$, the call arrival process is approximated by a Poisson process with rate $a$. Hence the blocking probability is approximated by the Erlang B formula, i.e., $E\left(n_\rho + k, k, \frac{1-\rho}{\rho}\right) \to B(k, a)$, as $\rho \to 1$.

Observe that since $B(k, 0) = 0, B(k, +\infty) = 1$, and $B(k, a)$ is increasing in $a$, there is a unique $a_f$ for which $B(k, a_f) = f$ holds. Thus, $n_*(1-\rho) \to a_f$ as $\rho \to 1$, and both the lower and upper bounds in Proposition 1 converge to $B(k, a_f)$.

### F. Proof of Proposition 2

The second derivative of $N$ is

$$N''(f) = \left(\frac{B}{B'}\right)^2 \left[\left(\rho^{-1} - 1\right)^2 + \frac{\rho^{-2}}{k} + B\left(\rho^{-1} - 1\right)\right] , \tag{45}$$

where $B = B(k, a_f) = f, B'' = \frac{\partial B}{\partial a}(k, a)\big|_{a=a_f}, \rho = a_f/k$. For $\rho \leq 1$, (45) is nonnegative.

To deal with $\rho > 1$ first note that $N''(f) > 0$ is equivalent to

$$B \leq \frac{k(\rho-1)^2+1}{k\rho(\rho-1)}\,. \tag{46}$$

But this inequality follows by noticing the expression on the right hand side is greater than the upper bound of $B$,

$$B \leq \frac{k(\rho-1)^2+2\rho+(\rho-1)\sqrt{4k\rho+k^2(1-\rho)^2}}{k\rho(\rho-1)+2\rho+\rho\sqrt{4k\rho+k^2(1-\rho)^2}}\,, \tag{47}$$

shown by Harel [25].

### G. Proof of Theorem 3

Lemma 3 allows one to consider the equivalent problem (40), where the additional constraint $x_n \geq x_{n+1}, n \geq 0$ implied by implementability is equivalent to (42) in the light of (38). Now Theorem 8 characterizes the optimal policy which is just (15), by (38) and the identification $n_* = m-1$.

To show that this policy is implementable, let $x(n), y(n)$ be the optimum solution of (14) for each $n \geq 1$; we show $x(n) = x_n$ for each $n$. First note that the definition of $v$ implies $\lambda_n \geq u'(x_{n_*})$ and so $x(n) \leq x_{n_*}$ for each $n$. $x(n) < x_{n_*}$ then $\lambda_n = u'(x(n)) > u'(x_{n_*})$ so $y(n) = 0$, i.e., $x(n) = C/(n+k)$. But $x_{n_*} > x_n = C/(n+k) \geq x(n)$ for all $n > n_*$, so $x(n) = C/(n+k) = x_n$ for each $n > n_*$. Now assume $n \leq n_*$. If $\lambda_n > u'(x_{n_*})$ then $x(n) = C/(n+k) \geq x_{n_*}$ which cannot hold since $x(n) = (u')^{-1}(\lambda_n) < x_{n_*}$. Thus, $\lambda_n = u'(x_{n_*})$ and so $x(n) = x_{n_*}$.

### H. Proof of Theorem 4

For every $m_0 \geq 0$ and $\rho \in (0,1)$ define $m_\rho = m_0/(1-\rho)$ and let $N^\rho$ be a random variable distributed according to the stationary distribution of the number of short flows under threshold policy (12) with $n_0 = m_\rho$. Also let $M^\rho$ be the stationary number of short flows under policy (15) with $n_* = m_\rho$. We will first show that $\lim_{\rho\uparrow 1} EM^\rho/EN^\rho = 1$.

Note that $((1-\rho)M^\rho, \rho \in (0,1))$ is a tight sequence of random variables since $\limsup_{\rho\uparrow 1}(1-\rho)EM^\rho \leq \lim_{\rho\uparrow 1}(1-\rho)EN^\rho = k+1+m_0B(k,m_0)$, where the last limit follows by Corollary 1. Thus $(1-\rho)M^\rho \xrightarrow{d} \hat{M}$ for some $\hat{M}$, over a converging[7] subsequence.

**Lemma 5.** *For every $\epsilon > 0$, $P((1-\rho)M^\rho \geq (1-\epsilon)m_\rho) \to 1$ as $\rho \uparrow 1$.*

*Proof:* First note that for the birth-death chain describing the number of short flows, in states $n \leq (1-\epsilon)m_\rho$ the birth to death transition rate ratio is at least $\rho/(1-\epsilon)$. This means that $M^\rho$ stochastically dominates $L$ where the latter has the stationary distribution of the birth-death process over states $\{0,\ldots,(1-\epsilon)m_\rho\}$ with birth and death rate $\rho$ and $1-\epsilon$ respectively. Now,

$$P\left(L(1-\rho) < m_0(1-2\epsilon)\right) \leq \left(\frac{1-\epsilon}{\rho}\right)^{m_\rho\epsilon} \tag{48}$$

for every $\rho$, and so $\lim_{\rho\uparrow 1} P(L(1-\rho) \leq m_0(1-2\epsilon)) = 0$. But then $\liminf_{\rho\uparrow 1} P(M^\rho(1-\rho) \geq m_0(1-2\epsilon)) \geq 1 - \lim_{\rho\uparrow 1} P(L(1-\rho) < m_0(1-2\epsilon)) = 1$. ∎

By Lemma 5, $\hat{M}$ must satisfy $P(\hat{M} > (1-\epsilon)m_0) = 1$ for every $\epsilon > 0$, and so $P(\hat{M} \geq m_0) = 1$. But then we must have[8] $P((1-\rho)M^\rho \geq m_0) \to 1$ over the convergent subsequence. Since this holds over any such subsequence we have $\lim_{\rho\uparrow 1} P(M^\rho \geq m_\rho) = 1$. In turn this implies, $\lim_{\rho\uparrow 1}(1-\rho)E(M^\rho) = \lim_{\rho\uparrow 1}(1-\rho)E(N^\rho)$ which establishes the claim.

We will also show that the two policies obtain the same CBF target fractions, or equivalently, the same throughput. The background throughput under the threshold policy is $b_{\text{opt}}(\rho) = CP(N^\rho = m_\rho)$ since background flows transmit only at the lowest state, i.e., $m_\rho$. On the other hand under (15) background flows grab whatever bandwidth is left over by TCP flows, i.e., $b_{\text{imp}}(\rho) = E\left(C - \frac{(\min(M^\rho, m^\rho)+k)C}{m_\rho+k}\right)$. We show that $b_{\text{imp}}(\rho)/b_{\text{opt}}(\rho) \to 1$ as $\rho \uparrow 1$. After some algebra,

$$\frac{b_{\text{imp}}(\rho)}{b_{\text{opt}}(\rho)} = \frac{P(M^\rho \geq m_\rho)\left(\frac{m_\rho}{m\rho+k} - \rho\right)}{P(M^\rho = m_\rho | M \leq m_\rho)} + \rho \to 1\,, \tag{49}$$

where the limit follows from the fact (e.g., see [26]) that $P(M^\rho = m_\rho | M^\rho \leq m_\rho) = B(m_\rho, m_\rho + \rho k - m_0) = O(\sqrt{1-\rho})$ as $\rho \uparrow 1$. This and the previous claim establish the theorem.

### I. Proof of Theorem 5

1) Follows by simple manipulations using the expressions from Corollary 1 and (16).

2) It is easier to consider the delay difference relative to $w$TCP:

$$\lim_{\rho\to 1} \frac{N_{\text{w}}(k,f,\rho) - N_*(k,f,\rho)}{N_{\text{w}}(k,f,\rho)} \leq \frac{\frac{kf}{1-f} - a_f f}{k + \frac{kf}{1-f}}$$
$$= \frac{1-f}{k} a_f \left(B(k-1,a_f) - f\right) \leq B(k-1,a_f) - f\,. \tag{50}$$

by using the identity $kB(k,a_f)/(1-B(k,a_f)) = a_f B(k-1,a_f)$ and the definition $B(k,a_f) = f$. The last step follows by noting that the average number of ongoing calls $a_f(1-B(k,a_f))$, in the associated loss system, is less than the number of circuits $k$. Reexpressing the delay difference relative to the optimal yields the bound in (18).

To get the upper bound, first notice that $\sup_{0\leq f<1}\left[B(k-1,a_f) - f\right] = \sup_{a\geq 0}\left[B(k-1,a) - B(k,a)\right]$, and for every $a \geq 0$, $F(k,a) = B(k-1,a) - B(k,a) \to 0$ as $k \uparrow \infty$. Moreover, since $B(k,a)$ concave in $k$ [27], $F(k,a)$ is nonincreasing in $k$. Thus the convergence is uniform over intervals of the form $[0,a_0]$. It is uniform over the entire positive axis if $\sup_{k\geq k_0, a\geq a_0} F(k,a)$ can be made arbitrarily small by some choice of $k_0, a_0$. But this follows since for every sequence $a_k \uparrow +\infty$ we have $F(k,a_k) \leq F(k_0,a_k)$ for any $k \geq k_0$, and $F(k_0,a_k) \to 0$ as $k \uparrow \infty$.

### J. Proof of Theorem 6

**Lemma 6.** $N_{**}(k,\rho) \geq N_*\left(k, \frac{\rho_b}{1-\rho}, \rho\right)$.

*Proof:* Fix any $(x_{n,m}, \pi_{n,m}, (n,m) \in \mathcal{S})$ satisfying (22)-(25) and define

$$\pi_n = \sum_{m\in\mathbb{N}\cup\{\infty\}} \pi_{n,m}\,, \quad x_n = \frac{\sum_{m\in\mathbb{N}\cup\{\infty\}} x_{n,m}\pi_{n,m}}{\sum_{m\in\mathbb{N}\cup\{\infty\}} \pi_{n,m}} \tag{51}$$

for every $n = 1,2,\ldots$ then it is easy to check that (3)-(6) hold with $f = \frac{\rho_b}{1-\rho}$. Moreover the policy $(x_n, n = 0,1,\ldots)$ for the chain in (1) achieves the same average number of short flows. ∎

The reverse inequality holds due to the following result which is also useful in showing Theorem 7.

[7] The symbol $\xrightarrow{d}$ denotes convergence in distribution.

[8] For this to hold we must also ensure that $P(\hat{M} = m_0) = 0$. This can be shown by an easy coupling argument which we omit because it is overly technical.

**Proposition 3.** *Let $(x_n, n \geq 0)$ be a policy for the chain* (1)*, possessing a stationary distribution $(\pi_n, n \geq 0)$ with*

$$\sum_{n=0}^{\infty} \pi_n \left[C - (n+k)x_n\right] = C\rho_b \,. \tag{52}$$

*Then any policy $(x_{n,m}, (n,m) \in \mathcal{S})$ for the chain* (20) *with $x_{n,m} = x_n$ for all $n \in \mathbb{N}, m \geq M$ for some $M \geq 1$, possesses a unique stationary distribution $(\pi_{n,m}, (n,m) \in \mathcal{S})$ which satisfies $\pi_{n,\infty} = \pi_n$ for all $n$.*

*In particular:*

1) *The mass of $(\pi_{n,m})$ is concentrated at $m = \infty$, i.e., the number of micro-flows is unstable.*
2) *The number of short flows is distributed according to $(\pi_n)$, and*
3) *the average throughput of the CBF satisfies* (25)*.*

*Proof:* Clearly $\pi_{n,\infty} = \pi_n, \pi_{n,m} = 0$ for all $m < \infty$, is a stationary distribution. To show that it is unique, we will show that there is no stationary distribution which assigns positive probability in states with $m < \infty$.

Let $(\tilde{n}_t, t \geq 0)$ be the Markov chain in (1), and define

$$\tilde{m}_t = \tilde{m}_0 + N^+\left(\lambda_b t\right) - N^-\left(\int_0^t \mu_b \left[C - (\tilde{n}_{s-} + k)\, x_{\tilde{n}_{s-}}\right] ds\right), \tag{53}$$

$t \geq 0$, where $N^+, N^-$ are unit rate Poisson processes independent of all other processes. $(\tilde{n}_t, \tilde{m}_t)$ evolves according to (20) in all states except those with $m \leq M-1$. Let $\underline{n} = \min\{n > 0 | \pi_n > 0\}$. Since $(\tilde{n}_t)$ is positive recurrent, $\tau_i$, the $i$-th return time to $\tilde{n}_t = \underline{n}$, for $i \geq 1$, has a finite expectation. Also,

$$\begin{aligned} E\left(\tilde{m}_{\tau_1} - \tilde{m}_0 | \tilde{m}_0, \tilde{n}_0 = \underline{n}\right) &= \lambda_b E(\tau_1 | \tilde{n}_0 = \underline{n}) \\ &\quad - E\left[\int_0^{\tau_1} \mu_b \left[C - (\tilde{n}_{s-} + k)\, x_{\tilde{n}_{s-}}\right] ds \,\middle|\, \tilde{m}_0, \tilde{n}_0 = \underline{n}\right] \\ &= \frac{\lambda_b}{\lambda \pi_{\underline{n}}} - \frac{1}{\lambda \pi_{\underline{n}}} \sum_n \mu_b \left[C - (n+k)x_n\right] \pi_n = 0 \,, \end{aligned} \tag{54}$$

where the last inequality follows by (52), and the one before by the cycle-formula. Thus the sequence $\tilde{m}_{\tau_1}, \tilde{m}_{\tau_2}, \ldots$ is a zero drift random walk on $\mathbb{Z}$. Consequently, the chain $(\tilde{n}_t, \tilde{m}_t)$ is null recurrent, and so is its truncation $(\hat{n}_t, \hat{m}_t)$ in $\mathbb{N} \times \{M, M+1, \ldots\} \subset \mathcal{S}$.

Now, the null recurrence of $(\hat{n}_t, \hat{m}_t)$ and the positive recurrence of $(\hat{n}_t)$ imply $P\left((\hat{n}_t, \hat{m}_t) \in \mathbb{N} \times \{M\}\right) \to 0$ as $t \to \infty$. Hence $\hat{m}_t$ visits $M$ very infrequently and so must the second component of the chain (20). Thus the latter process is null recurrent (in the component $m < \infty$) and so it does not possess a stationary distribution (with $m < \infty$ occuring with positive probability). ■

The policy $(x_{n,m})$ defined in Theorem 6 fulfills the conditions of Proposition 3 since (52) is equivalent to (6) for $f = \frac{\rho_b}{1-\rho}$. Thus there exists a policy for (20) which has an average number of short flows equal to $N_*\left(k, \frac{\rho_b}{1-\rho}, \rho\right)$. This implies $N_*\left(k, \frac{\rho_b}{1-\rho}, \rho\right) \geq N_{**}(k, \rho)$.

### K. Proof of Theorem 7

Define $(x_n)$ to be the $w$TCP policy with weight $M$. Then (52) holds because the left-hand side equals $\frac{M}{M+k} C(1-\rho) = C\rho_b$. Thus $(x_n), (x_{n,m})$ fulfill the conditions of Proposition 3 and so $(x_{n,m})$ behaves as $(x_n)$.

## REFERENCES


[1] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *ACM SIGCOMM Computer Communication Review*, vol. 4, no. 30, October 2000, pp. 43–56.

[2] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 362–373.

[3] L. Schrage, "Letter to the editor - a proof of the optimality of the shortest remaining processing time discipline," *Operations Research*, vol. 16, no. 3, pp. 687–690, 1968.

[4] B. Briscoe, "A fairer, faster internet protocol," *IEEE Spectrum*, vol. Dec 2008, pp. 38–43, Dec. 2008.

[5] N. Anderson, "Claim your 16$! comcast P2P settlement now final," in Ars Technica, July 2010, available at http://arstechnica.com/tech-policy/2010/07/claim-your-16-comcast-p2p-settlement-now-final/.

[6] A. Kuzmanovic and E. W. Knightly, "TCP-LP: low-priority service via end-point congestion control," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 739–752, August 2006.

[7] R. Venkataramani, R. Kokku, and M. Dahlin, "TCP Nice: A mechanism for background transfers," in *5th Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, USA, December 2002.

[8] A. Norberg, *uTorrent transport protocol*, 2009, draft.

[9] S. Shalunov, *Low Extra Delay Background Transport (LEDBAT)*, 2009, draft-shalunov-ledbat-congestion-00.

[10] T. Bonald and L. Massoulié, "Impact of fairness on Internet performance," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 29, no. 1. ACM, 2001, pp. 82–91.

[11] P. Key, L. Massoulie, and M. Vojnovic, "Farsighted users harness network time-diversity," in *INFOCOM, 2005 Proceedings IEEE*, vol. 4, 2005, pp. 2383–2394.

[12] C. Courcoubetis and A. Dimakis, "Fair background data transfers of minimal delay impact," *INFOCOM, 2012 Proceedings IEEE*, pp. 1053–1061, 2012.

[13] J.-Y. Le Boudec, "Rate adaptation, congestion control and fairness: A tutorial," *Web page, November*, 2005.

[14] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[15] F. Kelly, A. Maulloo, and Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[16] R. Gibbens and F. Kelly, "Resource pricing and evolution of congestion control," *Automatica*, vol. 35, pp. 1969–1985, 1999.

[17] S. Deb, A. Ganesh, and P. Key, "Resource allocation between persistent and transient flows," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 302–315, 2005.

[18] L. Massoulie and J. W. Roberts, "Bandwidth sharing and admission control for elastic traffic," *Telecommunication systems*, vol. 15, no. 1-2, pp. 185–201, 2000.

[19] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, 2000.

[20] G. De Veciana, T.-J. Lee, and T. Konstantopoulos, "Stability and performance analysis of networks supporting elastic services," *IEEE/ACM Trans. Netw.*, vol. 9, no. 1, pp. 2–14, 2001.

[21] M. Shaked and J. Shanthikumar, *Stochastic Orders*. Springer-Verlag New York, 2007.

[22] Y. R. Yang and S. S. Lam, "General AIMD congestion control," in *International Conference on Network Protocols 2000*. IEEE, 2000, pp. 187–198.

[23] S. McCanne and S. Floyd, "ns Network Simulator," http://www.isi.edu/nsnam/ns/.

[24] "Ledbat ns2 implementation," Online, available at http://perso.telecom-paristech.fr/ drossi/index.php?n=Software.LEDBAT.

[25] A. Harel, "Sharp and simple bounds for the Erlang delay and loss formulae," *Queueing Systems*, vol. 64, no. 2, pp. 119–143, 2010.

[26] D. Jagerman, "Some properties of the Erlang loss function," *Bell Systems Technical Journal*, no. 53, pp. 525–551, 1974.

[27] W. Karush, "A queuing model for an inventory problem," *Operations Research*, vol. 5, no. 5, pp. 693–703, 1957.

**Prof. Costas A. Courcoubetis** was born in Athens, Greece and received his Diploma (1977) from the National Technical University of Athens, Greece, in Electrical and Mechanical Engineering, his MS (1980) and PhD (1982) from the University of California, Berkeley, in Electrical Engineering and Computer Science. He was MTS at the Mathematics Research Center, Bell Laboratories, Professor in the Computer Science Department at the University of Crete, Professor in the Department of Informatics at the Athens University of Economics and Business, and since 2013 Professor in ESD Pillar, SUTD. His current research interests are economics and performance analysis of networks and internet technologies with applications in the development of pricing schemes that reduce congestion and enhance stability and robustness, sharing economy, regulation policy, smart grids and energy systems, resource sharing and auctions. Besides leading a large number of research projects in these areas he has also published over 100 papers in scientific journals such as Operations Research, Mathematics of Operations Research, Journal on Applied Probability, ToN, IEEE Transactions in Communications, IEEE JSAC, SIAM Journal on Computing, etc. and in conferences such as FOCS, STOC, LICS, INFOCOM. GLOBCOM, ITC, ACM SIGMETRICS. His work has over 12,000 citations according to the Google Scholar. He is co-author with Richard Weber of Pricing Communication Networks: Economics, Technology and Modeling (Wiley, 2003).

**Prof. Antonis Dimakis** received the BSc (1996) and MSc (1999) degrees in Computer Science from the University of Crete, Greece, and the PhD degree in Electrical Engineering and Computer Sciences from the University of California at Berkeley, in 2006. Since 2007 he has been with the faculty of Informatics at the Athens University of Economics and Business, where he is an Assistant Professor. His research interests lie in the areas of queueing systems, stability and optimal control of stochastic systems, and congestion control.

**Dr. Michalis Kanakakis** is a senior researcher with the Network Economics and Services Group in the Athens University of Economics and Business (AUEB). He received his BSc in Computer Engineering and Informatics in 2007, from the department of Computer Engineering and Informatics of the University of Patras, his MSc in Computer Science in 2009 and his PhD in Computer Science in 2016 from the Department of Informatics of the AUEB, under the supervision of Prof. Costas Courcoubetis. His research interests involve the application of tools from stochastic process theory in the analysis of transmission control protocols. He has been involved in several EU-funded FP7 and H2020 projects such as TRILOGY, OPTET, WATTALYST, OPTi and WiseGrid.