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 behave as low priority traffic, i.e., completely yield to delay sensitive flows such as web traffic. This behavior can cause starvation and hence the accumulation of large numbers of flows, leading to flow level instability. In this paper we look at the fundamental problem of designing congestion control protocols for background traffic with minimum impact on delay-sensitive 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 for the negative impact of TCP-based background transfer protocols compared to 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, e.g., in file servers. Simulations of simple networks suggest that this type of access control performs better than protocols emulating low priority.**

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 few 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 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 popular 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.

Not all applications value instantaneous bandwidth equally. It is valued more by web browsing flows than, say, background data transfers such as large batch software updates. The former serve interactive tasks where low transfer delays are important, while the latter are indifferent to small temporal variations of their bandwidth share, provided the data volume downloaded over a long time period does not change. It is well known from scheduling theory (see [2]) that short jobs or jobs with tighter deadlines should be assigned higher priority. Hence using TCP as the common transport protocol creates unnecessary delays to the web flows that are usually short and delay sensitive.

A possible solution, violating the end-to-end (e2e) principle of the Internet architecture, is for the ISP to intervene and throttle the bandwidth assigned to less delay sensitive flows, leaving more space for the web traffic, or offering some form of prioritization. But this is not in many cases an efficient solution, since the ISP cannot have the necessary information on how much throttling is necessary, and for which flows [3]. Unjustified throttling of traffic can have serious side effects for the ISP business, [4].

Recognizing this, Internet engineers have considered e2e solutions, and specialized congestion control algorithms for background data transfers have emerged, e.g., TCP-LP [5], TCP-nice [6], uTorrent transport protocol [7], LEDBAT [8]. Such protocols are typically designed to behave as low priority traffic, i.e., opting for low impact on the delay of short flows (that include the web flows). But the presence of 'elephant' flows, i.e., long lasting[1] flows that usually carry non-real time traffic, is a serious drawback as we explain next, motivating our approach.

In the presence of long lasting TCP flows sharing a link, any true low priority protocol will starve since the fraction of time the link is idle from TCP traffic will be negligible. Then new background flows will accumulate leading to an unstable system with sluggish performance, in contrast to the original intent of the protocol (e.g., see Section IV below). Even if the elephant traffic leaves some unused capacity and the low priority traffic can detect it, this might be too little to accommodate its load, leading again to instability[2]. There is also a serious problem, due to the shrinkage of the capacity region over paths having multiple links, even if no elephant flows are present, [10]. A low priority flow will transmit only if *all links* in the path are not used by other traffic, and this fraction of time decreases fast as the number of links increases.

The above discussion suggests that we need to engineer protocols that can compete with the elephant flows and *share with them* the excess capacity that is left over by the web traffic. But in doing so they need to become more aggressive, *unavoidably harming the short flows*.

This motivates the design of protocols for background traffic that reduce negative externalities to short flows while guaranteeing some minimum performance to the background flows. 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 same throughput (over slow timescales) as TCP would provide to the same flow, as in [11]. This achieves 'incentive compatibility' with the social planer of the ecosystem: it makes originators of the flows indifferent between adopting a new protocol instead of TCP, while reducing the average delays of web flows compared to the case where all flows use TCP.

To find the structure of the optimal congestion control we analyze the fundamental case of a single bottleneck link of capacity C. The Internet traffic passing through the link (see Fig. 1) is abstracted from all its unnecessary details and is comprised by

C. Courcoubetis is with the Singapore University of Technology and Design.
A. Dimakis and M. Kanakakis are with the Deparment of Informatics, Athens University of Economics and Business.

[1]Due to the heavy tail distribution of the file sizes [9].

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

- *Traffic outside our control*: it consists of i) short TCP flow arrivals, referred to as 'short' or 'web' flows, that transfer files with total load $C\rho$, hence leaving an *excess capacity* of $C(1-\rho)$ to the rest of the flows, and ii) a fixed number k of long TCP flows (the elephant flows from our earlier discussion), referred to as 'long' flows, assumed for simplicity to be always on[3].
- *Traffic that we control*: these are the 'Controlled Background Flows (CBFs)', i.e., flows carrying background data. 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 size of web flows, or an endless sequence of file transfer requests[4] at the point of entry. A natural application of this model is to systems where a level of aggregation is possible, such as BitTorrent peers or CDN servers.

Our goal is to derive fundamental limits for system performance and design controllers for the CBF traffic. Using a Markov decision process formulation, different optimal policies arise according to the information available. The 'first-best' assumes full information on the number of flows passing through the link at each time and serves as a benchmark (lower bound) to compare with other more practical polices. This policy cannot be implemented by a distributed controller using e2e information and needs access to centralized information available at the routers. The 'second-best' is the optimal 'implementable' policy, when only congestion feedback is available from the network. We use these results to obtain tight lower bounds for the performance of simple practical policies based on TCP.

A summary of our key findings follows.

Main results

The optimal full information policy. It minimizes the average short flow delays 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.

Another interpretation of this result is that it provides the optimal tradeoff between the average throughput of the CBFs and the delay inflicted to the short flows; any algorithm that achieves the same delay impact to the short flows, cannot do so by offering a higher throughput to the CBFs.

Moreover, the negative impact to the short flows can be arbitrarily large if $f \to 1$.

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, and can be implemented per CBF flow. It has a simple form: 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 second-best policy performs asymptotically as the first-best for $\rho \to 1$, and numerical analysis shows that it is within few percents of the first-best even for smaller values of ρ.

[3]Essentially we require k to change much slower than the time scales of background flow arrivals and departures, see next.

[4]Of sizes comparable to web flows.

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

Performance of TCP-based congestion controllers. Suppose we use wTCP[5] as our congestion control algorithm, where the value of w is chosen to obtain the required long-run average share f of the excess capacity. Then the relative increase of the delays of the short flows is within 20.7% of the optimal policy. This upper bound holds uniformly over $\rho, f, k \geq 1$. It is achieved for intermediate values of f (60%) and when $k=1$ long flows are in the system. When k increases, it decreases rapidly (for $k=3$ it becomes 11%) and hence the details of the congestion controller become insignificant.

This suggests a simple form of access control for CBFs with an arriving stream of files: each arriving background file, instead of immediately transmitting, 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 as to produce a critically loaded queue, i.e., produce long but not infinite queues.

For larger shares of f near one, wTCP's performance converges to the performance of the optimal policy. Hence either when k or f is large, 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 double the delay of short flows.

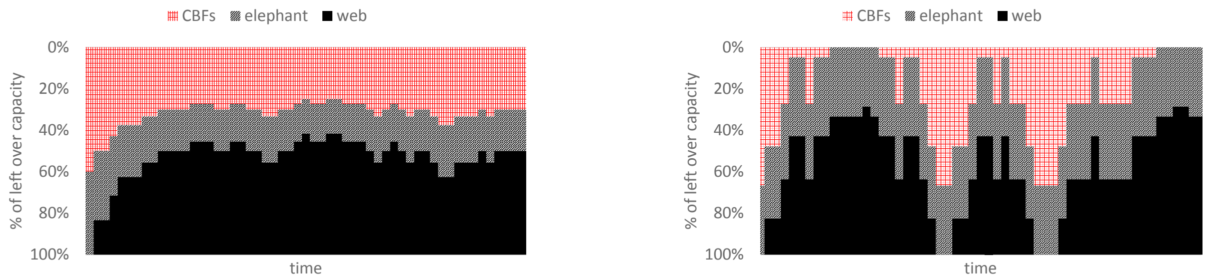
The rest of the paper is organized as follows. In Section II we introduce our system model of short flow arrivals at a single bottleneck link, and establish the optimality of threshold policies. Bounds and formulas for the minimum delay are also given in the case where the offered load of short flows is high. In Section II-C we obtain the optimal policy within the class of policies implementable by congestion feedback, and establish their optimality in heavy traffic. In Section II-D we assess the delay incurred by weighted TCP and compare it with the optimal. 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 congestion control protocols which emulate low priority. In Section V we give further justification of our model assumptions as well as discuss an extension of our methodology. Most proofs are relegated to the appendix.

A. Related work

In [13], [14] the effect of congestion control on the number of ongoing file transfers is studied. We take a similar viewpoint by considering a model where flow-level dynamics are described by a Markovian process, and ignore packet-level effects.

Deb et al. [15] 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 -both background and short- by maximizing a social welfare function which includes the average utility obtained by background traffic and the delay caused to web flows. Since we assume that part of the traffic, namely long and short TCP, is not conforming (i.e., is not optimized), the optimal policies differ considerably from the ones in [15].

[5]A wTCP connection obtains the equivalent of w individual TCP connections.



(a) Bandwidth sharing under a CBF protocol.

(b) A CBF protocol causing less delay to web flows.

Fig. 1. Illustration of bandwidth sharing model: a capacity C link is used by a constant number of CBF and long TCP flows, and a varying number short (web) flows which occupy a fraction ρ of link capacity. As new web flows arrive at the system or old ones complete their transfers, the link capacity is reallocated between all ongoing transfers. Since web and long flows both use TCP, these flows have equal shares at all times. It is possible to decrease the delays of web flows by choosing the CBF protocol in Fig. 1b which obtains a lower share during times where more web flows are in the system, while occupying the same average fraction f of the excess capacity $C(1-\rho)$ as the CBF in Fig. 1a.

A model with nonconforming traffic 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. The starting point of our work is that we turn this into an explicit optimization problem. It is interesting that our second-best policy has the same structure as the farsighted congestion controllers. Also, as mentioned earlier, our results imply these controllers are optimal within the class of implementable policies.

II. Bandwidth sharing for background flows

A. Basic model

Consider a link of capacity C shared by a set of CBFs, k long TCP flows which have always data to send, and a dynamically arriving stream of short TCP flows. The latter concern transfers of files with independent and exponentially distributed file sizes, of mean μ^{-1}, and arrive at the link according to a Poisson process with rate λ arrivals per unit time.

Here and in the next sections we seek to optimize the bandwidth sharing policies used by the CBFs[6] in order to minimize the delay impact on the short TCP flows.

Let x_n denote the download bandwidth of each TCP flow when the number of active short TCP flows is n. This number evolves according to a Markov chain with state space $\{0, 1, \ldots, \}$ 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* brought in the system by the short TCP flows is $C\rho$, where $\rho = \lambda/(\mu C)$ is the *normalized load*. 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 amount of capacity $C(1-\rho)$ left over by short TCP flows, is the *excess capacity* and is consumed in its entirety by the background flows, i.e., the k TCP and the CBFs.

[6]Their precise number does not matter as we will be optimizing the aggregate behavior; we could as well think of optimizing a single CBF.

Now, the choice of $(x_n, n = 0, 1, \ldots)$ determines how (actual, not excess) capacity is shared between (short or background) TCP and CBFs, since at state n the TCP flows use bandwidth $(k{+}n)x_n$ while the CBFs consume the remaining $C - (k+n)x_n$.

B. Optimal sharing

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 TCP flows is minimized, under the constraint that the CBF flows get a fraction f of the excess capacity?* Insofar as we only deal with the delay impact on short flows, we do not deal with how the f fraction is attributed between CBFs; in [12] this problem is considered using a utility maximization framework.

Since by Little's law the delay minimization of short flows is equivalent to minimizing their average number, we arrive at the following optimization problem:

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

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

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

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

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

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

Equalities (3) are the local balance equations corresponding to (1), (5) is the link capacity constraint, and (6) is 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*

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

for some finite nonnegative integer n_*.

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

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

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 (6), i.e.,

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

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*

$$E\left(n_* + k + 1, k, \frac{1-\rho}{\rho}\right) \le f < E\left(n_* + k + 2, k, \frac{1-\rho}{\rho}\right)\,, \tag{9}$$

where

$$E(m, q, r) = \frac{\binom{m-1}{q} r^q}{\sum_{i=0}^{q} \binom{m-1}{i} r^i}\,, m > q\,,$$

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 TCP flows $N_*(k, f, \rho)$ is obtained by invoking the following proposition, which holds for any (not necessarily optimal) threshold policy:

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

$$x_n = \begin{cases} 0\,, & \text{for each } n \le n_0\,, \\ \frac{C}{k+n} & \text{for each } n > n_0 \end{cases} \tag{10}$$

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

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

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

As ρ 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 Erlang B formula for a system with* k *circuits under a load of* a *Erlangs.*

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

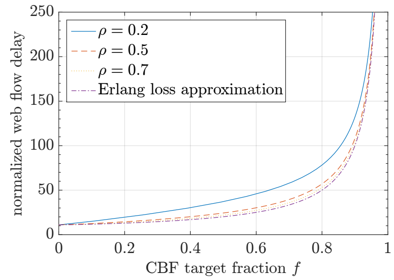


Fig. 2. Minimum delay of web flows (normalized by $1/(\mu - \lambda)$, i.e., the delay in the absence of all background traffic) under the optimal policy for CBFs, as a function of the fraction f of the excess capacity consumed by CBFs. The link is used also by $k = 10$ long TCP flows. The Erlang loss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

In Fig. 2, $N_*(10, f, \rho)$ is plotted against the target fraction f under various load levels ρ, 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 non-CBF background flows for excess capacity, for CBFs to interfere with short flows. Larger portions 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[7]. 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 Φ on the set of target fractions, i.e., the set $[0, 1]$. Let $N(f) = k + 1 + f a_f$ be the limit (as $\rho \to 1$) of the rescaled average number of short flows from Corollary 1. Then the following holds:

Lemma 1. *N is a convex function.*

Proof: The proof is in Appendix E ■

By Jensen's inequality,

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

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 ρ 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 ρ.

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

[7]Sufficient time between switching times should be allowed for the empirical averages to converge, i.e., we are considering a quasi-stationary regime.

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

$$\begin{aligned} \max \quad & (n+k)u(x) + v(y) \\ \text{such that} & (n+k)x + y \leq C \ , \\ \text{over } & x, y \geq 0 \ , \end{aligned} \tag{11}$$

attains its optimum at $x = x_n, y = C-(n+k)x_n$ for every $n > 0$. When such representation of $(x_n, n = 1, 2, \dots,)$ is possible then $x_n = (u')^{-1}(\lambda_n)$, where λ_n is the shadow price of the capacity constraint. A key insight from [16] is that, in a relaxation of (11), λ_n can be interpreted as the rate of congestion indicators fedback by the link to the end users. In [16] 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 (11), i.e., x_n, at equilibrium. For this reason, whenever a policy $(x_n, n = 1, 2, \dots)$ is represented as above, we say it is *implementable (by congestion feedback)*.

Note in particular that (11) implies that λ_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* $(x_n, n = 0, 1, \dots)$ *satisfies*

$$x_n = \begin{cases} x_{n-1} \ , & \text{if } n \leq n_* \ , \\ \frac{C}{k+n} & \text{if } n > n_* \end{cases} , n = 1, 2, \dots \tag{12}$$

for some finite nonnegative integer n_*.

The policy (12) is indeed implementable:

Remark. *The policy in* (12) *is represented by* (11) *by choosing* u *to be any utility function and* $v(y) = u'(x_{n_*})y$ *for all* $y \geq 0$.

Proof: Let $x(n), y(n)$ be the optimum solution of (11) 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 $x(n) = x_{n_*}$. ■

Observe that the slope $p_* = u'(x_{n_*})$ of $v(y) = p_* y$ in the representation depends on the utility function u of the short flows and x_{n_*} which in general is not known by the users. Since the slope is determined by (6), one could start with a utility $v(y) = py$ and adapt p in order for CBF to achieve the target fraction. Such an approach is followed in [11] where (12) 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 $l = 1, \dots, L$, implementability implies a representation similar to (11) is possible where y and $v(y)$ are replaced by $y_1, \dots, y_L$ 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, (12) 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 (12) 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 Figs. 4a-4b the ratio $M_*(1, f, \rho)/N_*(1, f, \rho)$ is plotted for $\rho = 0.5$ and $\rho = 0.9$ respectively. 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 (12) 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 in relative terms, when ρ is close to 1.

Under wTCP 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 wTCP is implementable by taking $v(y) = wu(y/w)$ in (11). Indeed it has been widely considered in the past (see e.g., [3], [17], [18]) and simple implementations exist[8].

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 TCP flows is

$$N_{\text{w}}(k, f, \rho) = \frac{(k + w + 1)\rho}{1 - \rho} = \frac{\rho}{1 - \rho} \left(k + 1 + \frac{kf}{1 - f} \right) . \tag{13}$$

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

Theorem 5 (wTCP performance relative to optimal)**.** *The following hold:*

1)
$$\begin{aligned} \lim_{\rho \to 1} \frac{N_w(1, f, \rho) - N_*(1, f, \rho)}{N_*(1, f, \rho)} &= \frac{f(1 - f)}{1 + (1 - f)^2} \\ &\leq \frac{3 - 2\sqrt{2}}{2\sqrt{2} - 2} \approx 20.7\% \ . \end{aligned}$$

[8]For example, [17] considers a modification of the additive increase and multiplicative decrease parameter of TCP.

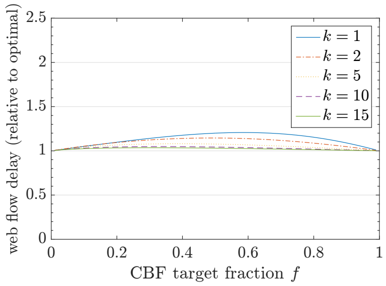


Fig. 3. The delay of web flows caused under the wTCP policy for CBFs, normalized by the delay achieved under the optimal policy for each level of CBF target fraction f. Their maximum difference is 20.7% attained when one long TCP exists in the system. The difference converges to zero as the number k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

The upper bound is tight and is achieved at $f = 2-\sqrt{2} \approx 60\%$.

2) *Let* $k \geq 2$. *Then for every* $0 \leq f < 1$*:*

$$\lim_{\rho\to 1} \frac{N_w(k,f,\rho) - N_*(k,f,\rho)}{N_*(k,f,\rho)} \leq b_k(f)\,, \tag{14}$$

$$\text{where } b_k(f) = \frac{B(k-1,a_f) - f}{1 - [B(k-1,a_f) - f]}\,,$$

with $\sup_{0\leq f<1} b_k(f) < \infty$ *decreasing to zero as* $k \to \infty$.

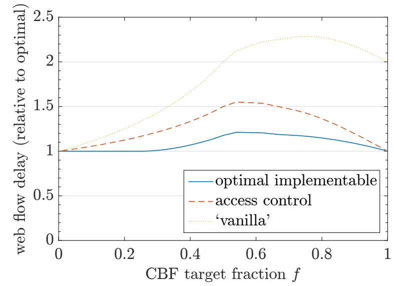
The theorem states that the relative difference between the delay of wTCP and the optimal is bounded. For $k = 1$ the maximum difference is about 20.7%. The second part of Theorem 5 says that the maximum difference goes to zero for large k. We stress that this does not need to be the case for *any* CBF 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 wTCP with $w_{\text{on}} > 0$, and the remaining time it has $w_{\text{off}} = 0$. A similar calculation as above shows that the target ratio f is achieved for $w_{\text{on}} = 2kf/(1-2f)$. 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 $k \geq 2$, wTCP 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, wTCP 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%.

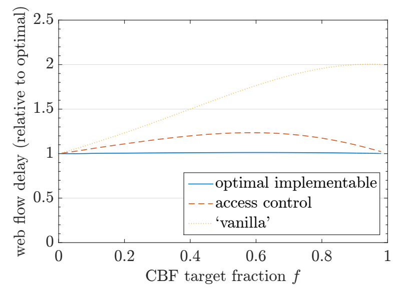
III. Dynamically arriving background flows

In the previous sections we used a system model with a fixed number of long background flows. This was justified by our assumption that 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 λ_b. Each micro-flow is associated with the download of a file with an exponentially distributed size with mean $1/\mu_b$.



(a) $\rho = 0.5$



(b) $\rho = 0.9$

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

Also the file sizes are assumed to be independent across different flows. If the CBF load λ_b/μ_b is less than the excess capacity and no amount of flow is lost, the fraction of the 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 normalized 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 TCP 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 the possibility $m = \infty$ and consider the extended state-space $\mathcal{S} = \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ [9].

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:

Assumption 1. $x_{n,m} \to x_{n,\infty}$ *as* $m \to \infty$ *for every* n*,*

This means when the number of micro-flows is high, each TCP flow will consume 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 $((N_t, M_t), t \geq$

[9] $\mathbb{N} \cup \{\infty\}$ is equipped with the one-point compactification of $\mathbb{N}$ induced by the embedding $m \mapsto \frac{1}{1+m}$ of $\mathbb{N} \cup \{\infty\}$ into $[0,1]$ and the identifications $m + \infty = \infty, 1/\infty = 0$.

0) which evolves according to the transition rates:

$$(n,m) \longrightarrow \begin{cases} (n+1,m) & \text{with rate } \lambda \\ (n-1,m) & \mu n x_{n,m}\ ,\ \text{if } n>0 \\ (n,m+1) & \lambda_b\ ,\ \text{if } 0 \le m < \infty \\ (n,m-1) & \mu_b\left[C-(n+k)nx_{n,m}\right]\ ,\ \text{if } 0 < m < \infty \end{cases} \tag{15}$$

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:

$$N_{**}(k,\rho) = \min \sum_{(n,m)\in\mathcal{S}} n\pi_{n,m} \tag{16}$$

$$\text{such that: } (\pi_{n,m},(n,m)\in\mathcal{S}) \tag{17}$$

is an invariant distribution of (15)

$$x_{n,m} \le \frac{C}{k+n}\ , n\ge 0,\ 1\le m\le\infty \tag{18}$$

$$x_{n,0} = \frac{C}{k+n}\ , n\ge 0 \tag{19}$$

$$\sum_{(n,m)\in\mathcal{S}} \left[C-(n+k)x_{n,m}\right]\pi_{n,m} = \frac{\lambda_b}{\mu_b}\ , \tag{20}$$

over $x_{n,m}, \pi_{n,m} \ge 0, (n,m) \in \mathcal{S}$. Note that the summations in (16),(20) include the points at infinity $m=\infty$. The constraint (20) 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 (16) coincides with (2).

Theorem 6. *$N_{**}(k,\rho) = N_*\left(k, \frac{\rho_b}{1-\rho}, \rho\right)$ and an optimal policy for* (16) *is $x_{n,m} = x_n$ for every $n \ge 0, m > 0$ including $m=\infty$, where $(x_n, n\in\mathbb{N})$ is the optimal policy* (8) *for $f = \frac{\rho_b}{1-\rho}$.*

Proof: See Appendix I. ∎

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 wTCP: the delay of the simple access control policy defined below, coincides with the delay of wTCP in a model without arrivals. In particular this and Theorem 6 imply that the delay induced to the short flows by the of 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[10], 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

$$x_{n,m} = \frac{C}{n+k+\min(m,M)}\ ,\ \text{at any state } (n,m)\in\mathcal{S}\ . \tag{21}$$

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 background flows which also use TCP. Choosing a too low M 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 TCP flows. This is because the micro-flows are kept *outside* of the network until they get to transmit. Choosing a too high M 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[11]. 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.

The above discussion is formalized in the following result:

Theorem 7. *Under the policy* (21)*, where M satisfies*

$$\frac{\rho_b}{1-\rho} = \frac{M}{k+M}\ , \tag{22}$$

the average number of short TCP flows is the same as under a single wTCP 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 (13)*.*

Proof: See Appendix J. ∎

IV. Access control versus Less than Best Effort protocols

Consider a file server which transmits low priority 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 other 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 less-than-best-effort (LBE) protocols such as LEDBAT and TCP-LP, using simulation.

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 (22) 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 by

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

[11]This is true under our simplifying view that we care only for the CBF throughput, not the micro-flow download delay. In practice one would not want these delays to be infinite, and thus pick M strictly above the minimum required for stability.

moving averages with sufficiently long memory such the effect of request arrivals and service completion dynamics 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 our access control policy with the performance of other LBEs 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 high load, the presence of elephant flows, or LBE flows traversing several links. Access control on the other hand, leaves flows outside of the network when congested, and so performs significantly better.

A. Experimental setup

The simulations are performed in ns2 [19] where TCP flows use Reno, TCP-LP flows use the implementation provided in ns2, and LEDBAT flows use the implementation in [20]. 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 random and follow an exponential distribution with mean 3Mbytes.

We consider three linear network topologies described next.

B. Single link

The average load $C\rho_b$ brought to the server by the requests was 3Mbps. In the top row of Fig. 5 the delay of web flows is depicted for various levels of load ρ. The delay is normalized by the delay of the 'vanilla' policy for each level of ρ. In Fig. 5a, where no elephant flows exist, the delay under access control is comparable to LBE protocols. In very heavy loads CBF performs significantly better: at 95% total load 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 a single elephant flow is added as in Fig. 5b: above 90% total load, LBE protocols cause worse delays than the 'vanilla' policy.

To see what happens, consider the normalized average number of active (background) flows when no elephant flows are present, depicted in Fig. 5c. (Again, the normalization is done with respect to the number of active flows under the 'vanilla' policy for the same load ρ.) 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 85% total load-.

Under the presence of a single elephant flow, in Fig. 5d, the number of background 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 already attains its minimum possible share[12]. Thus LBE traffic essentially stops behaving as low priority and competes more equally with TCP. This has a damaging effect on the delay of web flows as the traffic composition now contains more aggressive flows.

On the other hand, the web flow delay caused by access control continues to decrease (relative to 'vanilla') even after the addition of the elephant flow. This is because the active number of 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 web flows.

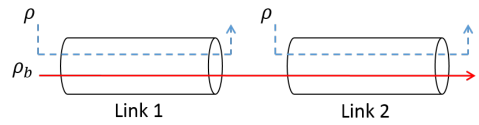


Fig. 6. The simulated two link network topology.

C. Two link network

Here we consider the two link network in Fig. 6, without the presence of elephant flows, in which $C\rho_b$ =3Mbps again but now the background flows traverse both links. In each link there is a separate stream of web flows of normalized load ρ.

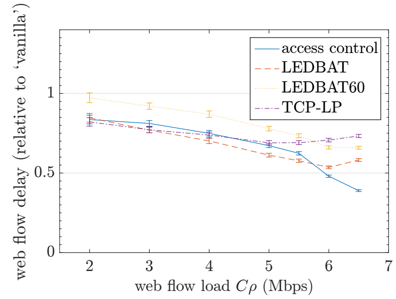
Fig. 7 depicts the web flow delay and number of active background flows for different load levels ρ. In Fig. 7a the delay under access control is similar to LEDBAT for web flow loads less than 5 Mbps. Above this value a similar effect to that when elephant flows are present occurs: in Fig. 7b the number 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., $C\rho > 4.6$Mbps, as truly low priority flows would be able to push data through the two links only at times where no web flows are present - an event which occurs with probability $1-\rho$- (see [10]). This explains the peak in the relative number of LEDBAT flows after $C\rho = 5$Mbps shown in Fig. 7b. Starvation in practice means the absolute number of background flows will increase significantly as in the case with elephant flows. Again since the LEDBAT flow throughput cannot be compressed below a certain limit, LEDBAT stops behaving as low priority traffic and so web delay is damaged. 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 limited number of active micro-flows transmit.

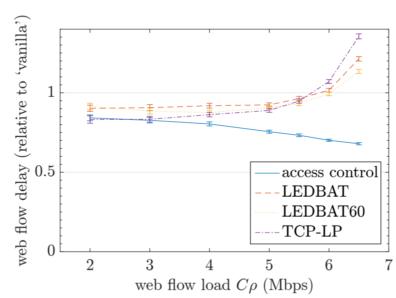
D. Three links

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 web flows with load ρ_1. On each of `route123`,`route2`, and `route23` there is a CBF (carying its own stream of micro-flows) with load $\rho_{123} = 2, \rho_2 = 1, \rho_{23} = 1$, respectively. These three routes also

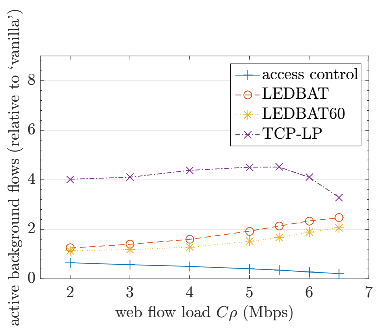
[12] A LBE flow sends at least one packet per round-trip-time, as TCP.



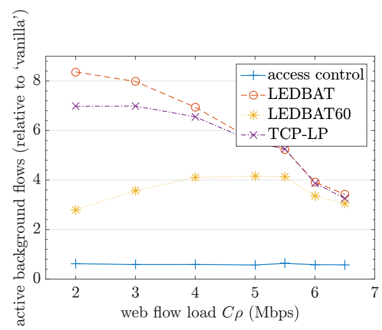
(a) Normalized delay, no elephant flows present ($k = 0$)



(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 web flow delay for each protocol normalized by the delay caused by the 'vanilla' policy for the same level of web flow load. The presence of a single elephant 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.

carry streams of web flows with the same load as the CBFs on the same routes.

Fig. 9 depicts the normalized delay of web 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 web 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

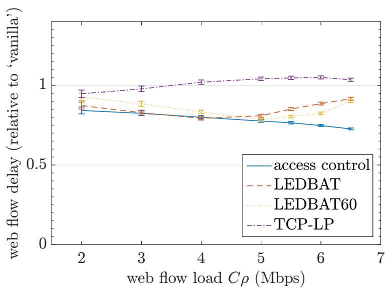
Why low priority CBFs are not optimal? Afterall, in reality background flows are usually transfers of finite-size files, so CBFs could transmit only when short or long TCP flows are not present. There are two reasons why low priority is not desirable as a design goal. Firstly, these are not good emulators of low priority especially in highly loaded links, as seen in simulations of the LBE protocols in Section IV. Secondly, there may be additional system objectives besides the delay of short flows, e.g., objective function terms involving the performance of background flows, such as sums of utilities as in [11], [12], utility rates as in [15], or the delay of CBFs.

In what follows we postulate a system model which dispenses the assumption of a constant number of background flows (long TCP and CBFs) and consider optimization criteria which encompass the performance of these flows.

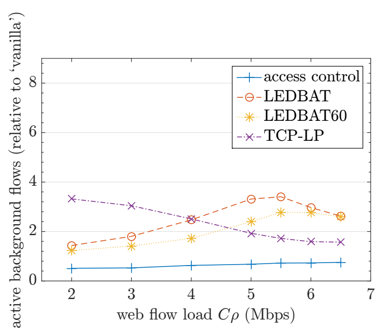
The model should include states of the form (k, l, n) where k, l, n are the numbers of long TCP, CBFs and web flows, respectively, present in the system. Under similar conditions on the arrival and file-size statistics, we are led to consider policies specified by the throughput $x_{k,l,n}$ a TCP flow achieves at every state (k, l, n).

If the sizes of background flows are much larger than those of the short flows, and the arrival rate of the former is much lower than the latter then the flow dynamics evolve on two distinct timescales, since web flows vary much faster now than background flows do. If the 'slow' state (k, l) remains constant for a sufficiently long time, the 'fast' state n evolves according to an ergodic Markov chain with transition rates determined by $(x_{k,l,n}, n \geq 0)$.

Over the time where (k, l) is constant, the average delay of short flows is determined by the quasi-stationary distribution at (k, l). In the same time, the CBFs obtain some fraction $f_{k,l}$ (also determined by the quasi-stationary distribution) of the excess capacity and the rest is consumed by the long TCP flows. Thus $f_{k,l}$ determines the rates at which background flows of each type depart from the system, and so the slow part of the state has a stationary distribution determined by the fractions $f_{k,l}$ achieved by CBFs at each slow state (k, l).



(a) Web flow delay for each protocol normalized by the delay caused by the 'vanilla' policy for the same level of web flow load.



(b) The average number of active micro-flows normalized by the corresponding number under the 'vanilla' policy for the same level of load.

Fig. 7. *Two link network*: comparison of access control, 'vanilla', and LBE protocols for a CBF comprised by a stream of dynamically arriving micro-flows. LBE starvation effects occur at $C\rho \geq 4.6$Mbps even without the presence of elephant flows.

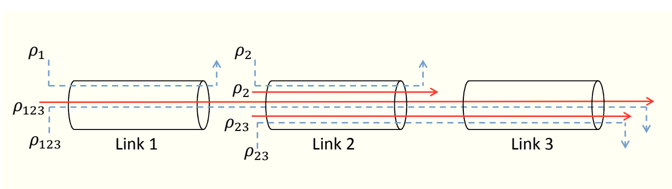


Fig. 8. The three link network simulated in Section IV-D under two different loads ($\rho_1 = 0.3$ and 0.5) for the web flows traversing only link 1.

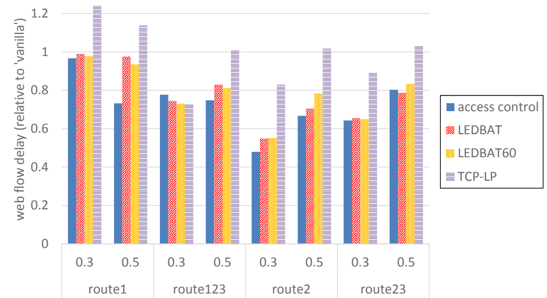


Fig. 9. *Three link network:* A comparison of access control, 'vanilla', and LBE protocols for a CBF comprised by an arriving stream of micro-flows. The web flow delay (normalized by the delay caused by the 'vanilla' policy) for each route and each value for $\rho_1 \in \{0.3, 0.5\}$. The access control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

Thus the assumption of a constant number of background flows made in this paper is not unjustified, and this provides the motivation. In the terminology of the present section, in Section II-B $(x_{k,l,n}, n \geq 0)$ is chosen to minimize the delay of short flows for each (k, l), for any value of $f_{k,l}$. Thus, the policies considered in this paper are relevant whenever the overall system objective includes the delay of short flows along with additional terms involving the performance of background flows. Since both type of performance criteria are determined by the choice of $f_{k,l}$ at every k, l, the system objective is optimized by solving a Markov decision problem involving the slow state process.

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[13], which can be based on the policies of this paper, and ii) *slow timescale feedback control*, that looks at the state of the CBF and adapts, in the slow timescale, the $f_{k,l}$ parameters supplied to the fast timescale congestion controller.

An interesting problem is to assess how well simple CBF controllers perform relative to the optimal, as the latter is likely to depend on non-observable quantities such as the number of background flows.

Appendix
Proofs

A. Proof of Theorem 1

Consider the case $(1-\rho)^k > f$ first. If $x_n, n = 0, 1, \dots$ are defined as in Theorem 1, CBFs do not affect the dynamics of the Markov chain. Hence, the stationary distribution is given by Lemma 5 for $n_0 = 0$. In fact, the average bandwidth constraint (6) is readily shown to hold for this stationary distribution. Since the average number of TCP flows is minimized by setting x_n at their maximum value $\frac{C}{k+n}$ for each $n = 1, \dots$, we conclude that the allocations $x_n, n = 0, 1, \dots$ are optimal.

For the remainder of this section we consider the case $(1-\rho)^k \leq f$.

Lemma 2. *If* $(1-\rho)^k \leq f$ *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 5 for $n_0 = 0$, and

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

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 TCP flows, while the increase of x_0 does not have any effect whatsoever. Thus, the optimal allocation x_0 must be zero. ■

[13]Few round-trip times needed for TCP window adaptation to converge.

We now transform the optimization problem into an equivalent decision problem. For each $n \geq 0$ and x_n, π_n which satisfy (3)-(7), define

$$\bar{\pi}_n = x_n \pi_n \frac{k}{C(1-\rho)(1-f)}\,, y_{n+1} = \frac{\bar{\pi}_n}{\bar{\pi}_{n+1}}\,. \quad (23)$$

The following holds:

Lemma 3. *The numbers $\bar{\pi}_n, y_n, n \geq 0$ as defined in* (23) *satisfy the constraints*

$$\sum_{n=0}^{\infty} \bar{\pi}_n = 1\,, \quad (24)$$

$$y_{n+1} \leq \frac{n+1}{\rho(k+n)}\,, n \geq 0\,, \quad (25)$$

$$\sum_{n=0}^{\infty} n\bar{\pi}_n = \frac{\rho k}{(1-\rho)(1-f)}\,, \quad (26)$$

$$\bar{\pi}_n \geq 0\,, n = 0, 1, \ldots \quad (27)$$

Conversely, for any $\bar{\pi}_n, y_n, n = 0, 1, \ldots$ satisfying (24)-(27) *there exist unique $\pi_n, x_n, n = 0, 1, \ldots$ for which* (3)-(7) *and* (23) *hold.*

Proof: Summing (23) over $n = 0, 1, \ldots$ and using (6) yields (24). (25) follows by mutliplying both sides of (5) by π_n and utilizing (3),(23). Since $\rho < 1$, the average rate of departures must equal λ, that is,

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

which proves (26).

To show the converse part, define

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

$$\text{and } x_n = \begin{cases} \frac{C\rho y_{n+1}}{n+1}\,, & \text{if } \bar{\pi}_{n+1} > 0\,, \\ 0\,, & \text{if } \bar{\pi}_{n+1} = 0\,, \end{cases} \text{ for each } n = 0, 1, \ldots \quad (28)$$

Now, (3) follows by noting that $\frac{n y_n}{n+1} = \frac{\pi_{n-1}}{\pi_n}$ and substitution in the definition of x_n above. Also,

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

by making use of (24). (5) follows by (25) and the definition of x_n above, while (6) follows directly from the definition of π_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 (23) and (3) defines π_n uniquely by the latter equation. Thus, x_n is defined uniquely by (23) for every $n \geq 1$, while x_0 is determined by (6). ■

Now, if we multiply both sides of (3) by $n-1$ and sum over $n = 1, 2, \ldots$ we get

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

when (26) holds. Thus, the minimization of the objective in (2) is equivalent to that of the second moment of the distribution $\bar{\pi}_n, n = 0, 1, \ldots$, when (26) holds. Combining this with Lemma 3 leads to the following equivalent formulation:

$$\text{Minimize } \sum_{n=0}^{\infty} n^2 \bar{\pi}_n \text{ over } \bar{\pi}_n, y_n, n \geq 0$$

such that (24)-(27) and the second equation in (23) hold. (29)

The following Theorem characterizes the optimal solution. It also covers the restriction to the implementable class of policies considered in Theorem 3.

Theorem 8. *The optimal solution of the minimization problem* (29) *satisfies:*

$$y_n = \begin{cases} \frac{n}{\rho(k+n-1)} & n > m \\ 0 & n < m \end{cases}\,, n = 1, 2, \ldots\,, \quad (30)$$

for some $m \geq 1$.

If in addition to (24)-(27),(23) *the constraint*

$$\frac{y_{n+1}}{n+1} \leq \frac{y_n}{n}\,, n = 1, 2, \ldots \quad (31)$$

is imposed, the optimal solution satisfies

$$y_n = \begin{cases} \frac{n}{\rho(k+n-1)} & n \geq m \\ \frac{n y_{n-1}}{n-1} & 1 \leq n < m \end{cases}\,, \quad (32)$$

for some $m \geq 1$.

Theorem 1 follows as a direct corollary of Theorem 8, since then by Lemma 3, there exist unique $x_n, n \geq 0$ given by (28) in terms of the optimal solution $y_n, n \geq 1$ of (29). Thus (8) follows by defining $n_* = m - 2$, where m is as in Theorem 8.

The proof of Theorem 8 itself is based on the following comparison lemma:

Lemma 4. *Let $(y_n), (y'_n)$ be two sequences of transition rates for which the respective stationary distributions $(\bar{\pi}_n)$ and $(\bar{\pi}'_n)$ they induce satisfy* (26)*, and $y_{m+1} < y'_{m+1}, y_n = y'_n$ for all $n \notin \{m, m+1\}$ for some $m \geq 1$.*

Then $\sum_n n^2 \bar{\pi}'_n \leq \sum_n n^2 \bar{\pi}_n$ holds.

Proof: We show that $\bar{\pi}$ dominates $\bar{\pi}'$ 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 $\mathrm{sgn}(\bar{\pi}_n - \bar{\pi}'_n) = \mathrm{sgn}(\bar{\pi}_0 - \bar{\pi}'_0)$ [14] for all $n < m$ since $y_n = y'_n$ in that range. Similarly $\mathrm{sgn}(\bar{\pi}_n - \bar{\pi}'_n) = \mathrm{sgn}(\bar{\pi}_{m+1} - \bar{\pi}'_{m+1})$ for all $n > m$ is true. Now

$$\mathrm{sgn}(\bar{\pi}_m - \bar{\pi}'_m) = \mathrm{sgn}\left(\bar{\pi}_{m-1} y_m^{-1} - \bar{\pi}'_{m-1} {y'_m}^{-1}\right) \leq \mathrm{sgn}\left(\bar{\pi}_{m-1} - \bar{\pi}'_{m-1}\right)\,,$$

since $y'_m \leq y_m$ which follows from the assumption that $\bar{\pi}$ and $\bar{\pi}'$ have the same mean and $y'_{m+1} \geq y_{m+1}$. In turn this implies $\mathrm{sgn}(\bar{\pi}_m - \bar{\pi}'_m) \leq \mathrm{sgn}(\bar{\pi}_{m+1} - \bar{\pi}'_{m+1})$. Thus, $\bar{\pi}_m - \bar{\pi}'_m$ can change sign at most twice as n goes from 1 to ∞. It is easy to see that the distributions $\bar{\pi}$ and $\bar{\pi}'$ are not stochastically ordered so Theorem 1.A.12 in [21] implies that $\bar{\pi}_m - \bar{\pi}'_m$ cannot change sign only

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

once. Thus there are exactly two sign changes and so by Theorem 3.A.57 $\bar{\pi}$ dominates $\bar{\pi}'$ in the convex stochastic order. ■

Now we are ready to prove Theorem 8.

Proof: Let y be the optimal solution of (29) 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' 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 $\bar{\pi}'$ still satisfies (26). 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 (30).

If (31) 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 (32). ■

B. Proof of Proposition 1

Lemma 5. *The stationary distribution of the number of short TCP flows is*

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

for the threshold policy with threshold n_0. 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 $1-\rho$. Then,

$$P(X = n) = \binom{n+k}{k}(1-\rho)^{k+1}\rho^n\,,$$

for every $n = 0, 1, \ldots$. This distribution is readily shown to satisfy the detailed balance equations

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

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.,

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

and 0 for $n < n_0$, where we have used the identity

$$\sum_{n=n_0}^{\infty}\binom{n+k}{k}(1-\rho)^{k+1}\rho^n = \sum_{i=0}^{k}\binom{n_0+k}{i}(1-\rho)^i\rho^{n_0+k-i} \tag{33}$$

which follows by noting 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. ■

The proof of the Proposition follows by the observation that the threshold policy with the threshold set at n_0 with $x_{n_0} = 0$ obtains a fraction $f = \dfrac{\pi_{n_0}}{1-\rho} = E\left(n_0+k+1, k, \dfrac{1-\rho}{\rho}\right)$ of the excess capacity, by Lemma 5. See [12] for more details.

C. Proof of Theorem 2

The average number of short TCP flows at stationarity is, by Lemma 5,

$$\begin{aligned} E(X|X \geq n_0) &= n_0 - 1 + \sum_{n=n_0}^{\infty} P(X \geq n|X \geq n_0) \\ &= n_0 - 1 + \frac{\sum_{i=0}^{k} G_i}{P(X \geq n_0)}\,, \end{aligned} \tag{34}$$

$$\begin{aligned} \text{where } G_i &= \sum_{n=n_0}^{\infty}\binom{n+k}{i}(1-\rho)^i\rho^{n+k-i} \\ &= \frac{1}{1-\rho}\sum_{j=0}^{i}\binom{n_0+k}{j}(1-\rho)^j\rho^{n_0+k-j}\,, \end{aligned}$$

by identifying $k \equiv i$ and $n_0 \equiv n_0+k-i$ in (33), and so,

$$\sum_{i=0}^{k} G_i = \sum_{i=0}^{k}\frac{k+1-i}{1-\rho}\binom{n_0+k}{i}(1-\rho)^i\rho^{n_0+k-i}\,.$$

Plugging this back to (34) gives the formula for N_{n_0} in the statement after some algebra.

D. Proof of Corollary 1

Consider a sequence of threshold policies indexed by ρ for which the threshold level n_ρ 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 ρ 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., $E\left(n_\rho+k, k, \dfrac{1-\rho}{\rho}\right) \to B(k, a)$, 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 $B(k, a_f)$.

E. Proof of Lemma 1

The second derivative of N is $N''(f) = \dfrac{1}{(B')^2}\left[2(B')^2 - BB''\right]$, where B, B', B'' are the values of $B(k, a)$, its first, and second derivatives respectively, with respect to the second argument evaluated at $a = a_f$. (We have also used the fact that $B(k, a_f) = f$ for all f, and so $B'a'_f = 1, B''{a'_f}^2 + B'a''_f = 0$.)

But, $B' = B\,(k/a_f - 1 + B)$ and a further differentiation gives

$$N''(f) = \left(\frac{B}{B'}\right)^2\left[\left(\rho^{-1}-1\right)^2 + \frac{\rho^{-2}}{k} + B\left(\rho^{-1}-1\right)\right]\,, \tag{35}$$

where $\rho = a/k$. For $\rho \leq 1$, (35) is nonnegative.

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

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

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

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

shown by Harel [22].

F. Proof of Theorem 3

Lemma 3 allows one to consider the equivalent problem (29). The constraint $x_n \geq x_{n+1}, n \geq 0$ implied by implementability is equivalent to (31) in the light of (28). Now Theorem 8 characterizes the optimal policy which is just (12) by (28) and the identification $n_* = m - 1$.

G. 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^ρ be a random variable distributed according to the stationary distribution of the number of short flows under threshold policy (10) with $n_0 = m_\rho$. Also let M^ρ be the stationary number of short flows under policy (12) 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 subsequence. By Lemma 6 below, any such limit 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[15] $P((1-\rho)M^\rho \geq m_0) \to 1$ 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,

$$\begin{aligned}\lim_{\rho\uparrow 1}(1-\rho)E(M^\rho) &= \lim_{\rho\uparrow 1}(1-\rho)E(M^\rho | M^\rho \geq m_\rho)\\ &= \lim_{\rho\uparrow 1}(1-\rho)E(N^\rho)\ ,\end{aligned}$$

which establishes the claim.

We will also show that the two policies obtain the same CBF target fractions, or equivalently, the same throughputs. 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_ρ. On the other hand under (12) 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 $\rho \uparrow 1$.

First note that

$$E\left(M^\rho 1\{M^\rho \leq m_\rho\}\right) = \sum_{n=0}^{m_\rho - 1}(n+1)P(M^\rho = n+1)$$

$$= \sum_{n=0}^{m_\rho - 1}\rho(m_\rho + k)P(M^\rho = n) = \rho(m_\rho + k)P(M^\rho < m_\rho)\ ,$$

and thus $b_{\text{imp}}(\rho) = C\left[\frac{m_\rho}{m_\rho + k}P(M^\rho \leq m_\rho) - \rho P(M^\rho < m_\rho)\right] =$ $CP(M^\rho \leq m_\rho)\left[\frac{m_\rho}{m\rho + k} - \rho + \rho P(M^\rho = m^\rho | M^\rho \leq m^\rho)\right]$.

Hence, $b_{\text{imp}}(\rho)/b_{\text{opt}}(\rho)$ equals

$$\frac{P(M^\rho \leq m_\rho)\left[\frac{m_\rho}{m\rho + k} - \rho + \rho P(M^\rho = m^\rho | M^\rho \leq m^\rho)\right]}{P(M^\rho = m_\rho | M \geq m_\rho)}$$
$$= \frac{P(M^\rho \geq m_\rho)\left(\frac{m_\rho}{m\rho + k} - \rho\right)}{P(M^\rho = m_\rho | M \leq m_\rho)} + \rho \to 1\ ,$$

where the limit follows from the fact (e.g., see [23]) 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.

Lemma 6. *For every $\epsilon > 0$, $P(M^\rho \geq (1-\epsilon)m_\rho) \to 1$ 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^ρ 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 ρ and $1-\epsilon$ respectively. Now,

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

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

H. Proof of Theorem 5

1) Follows by simple manipulations using the expressions from Corollary 1 and (13).

2) It is easier to consider the delay difference relative to wTCP:

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

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 (14).

To get the upper bound, first notice that $\sup_{0\leq f\leq 1}[B(k-1,a_f) - f] = \sup_{a\geq 0}[B(k-1,a) - B(k,a)]$, 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 [24], $F(k,a)$ 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 $k \uparrow \infty$.

Lastly, notice that for $k \geq 2$ we have $F(k,0) < 1$, and so the uniform bound is non-trivial since $\sup_{0\leq f<1} b_k(f) < \infty$.

I. Proof of Theorem 6

First we show the inequality:

Lemma 7. $N_{**}(k,\rho) \geq N_*\left(k, \frac{\rho_b}{1-\rho}, \rho\right)$.

Proof: Fix any $(x_{n,m}, \pi_{n,m}, (n,m) \in \mathcal{S})$ satisfying (17)-(20) and define

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

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. ■

[15] 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.

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

Proposition 2. *Let $(x_n, n \geq 0)$ be a policy for the chain* (1)*, possessing an invariant distribution $(\pi_n, n \geq 0)$ with*

$$\sum_{n=0}^{\infty} \pi_n \left[C - (n+k)x_n\right] = C\rho_b \,. \tag{36}$$

Then any policy $(x_{n,m}, (n,m) \in \mathcal{S})$ for the chain (15) *with $x_{n,m} = x_n$ for all $n \in \mathbb{N}, m \geq M$ for some $M \geq 1$, possesses a unique invariant 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 (π_n), and*
3) *the average throughput of the CBF satisfies* (20).

Proof: Clearly $\pi_{n,\infty} = \pi_n, \pi_{n,m} = 0$ for all $m < \infty$, is an invariant distribution. To show that it is unique, we will show that there is no invariant 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

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

$t \geq 0$, where N^+, N^- are unit rate Poisson processes independent of all other processes. $(\tilde{n}_t, \tilde{m}_t)$ evolves according to (15) 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, τ_i, the i-th return time to $\tilde{n}_t = \underline{n}$, for $i \geq 1$, has a finite expectation. Also,

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

where the last inequality follows by (36), 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 (15). Thus the latter process is null recurrent (in the component $m < \infty$) and so it does not possess an invariant distribution (with $m < \infty$ occuring with positive probability). ■

The policy $(x_{n,m})$ defined in Theorem 6 fulfills the conditions of Proposition 2 since (36) is equivalent to (6) for $f = \frac{\rho_b}{1-\rho}$. Thus there exists a policy for (15) which has an average number of short flows equal to $N_*\left(k, \frac{\rho_b}{1-\rho}, \rho\right)$. This implies $N_*\left(k, \frac{\rho_b}{1-\rho}, \rho\right) \geq N_{**}(k, \rho)$.

J. Proof of Theorem 7

Define (x_n) to be the wTCP policy with weight M. Then (36) 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 2 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] 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.

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

[4] 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/.

[5] 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. [Online]. Available: http://dx.doi.org/10.1109/TNET.2006.879702

[6] 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.

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

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

[9] M. E. Crovella, M. S. Taqqu, and A. Bestavros, "Heavy-tailed probability distributions in the world wide web," *A practical guide to heavy tails*, vol. 1, pp. 3–26, 1998.

[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. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, 2005, pp. 2383–g–2394.

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

[13] 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.

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

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

[16] 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.

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

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

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

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

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

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

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

[24] W. Karush, "A queuing model for an inventory problem," *Operations Research*, vol. 5, no. 5, pp. pp. 693–703, 1957. [Online]. Available: http://www.jstor.org/stable/167469