Competitive and Considerate Congestion Control for Bulk-Data Transfers[1]

Shao Liu, Milan Vojnović, and Dinan Gunawardena

February 2006

Technical Report
MSR-TR-2006-24

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

[1]Author affiliations: Shao Liu is with UIUC (e-mail: shaoliu@ifp.uiuc.edu); Milan Vojnović is with Microsoft Research, Cambridge (e-mail: milanv@microsoft.com); Dinan Gunawardena is with Microsoft Research, Cambridge (e-mail: dinang@microsoft.com). Work of Shao Liu performed in part while an intern with Microsoft Research.

Abstract—The current Internet features traffic from diverse applications; ranging from delay-sensitive web browsing to delay-insensitive data file transfers. This motivates service differentiation, yet router-centric solutions, e.g. diffserv, have not been widely deployed. The current practice relies on a limited service differentiation at network edges (e.g. through traffic managing middle-boxes or by the end-hosts). End-hosts often implement such emulators of low priority service to differentiate low and normal priority traffic. A low priority service may fit well for some applications, e.g., software updates, but may not be adequate for bulk file transfers that aim at large throughputs. A scenario motivating home users is that of many simultaneous bulk transfers. This is a common feature of peer-to-peer file sharing applications.

We develop a novel end-to-end congestion control that emulates a different service differentiation than the common low-normal priority. We call the new protocol 4CP (Competitive, Considerate Congestion Control). The target service differentiation enables provisioning of per-flow average bandwidth guarantees to "normal" traffic, but not at the expense of potentially starving the "low" priority traffic (4CP). It thus features incentive compatibility to file-transfer applications that are throughput-greedy but want to be considerate to other traffic.

4CP is implemented and configured as a sender-only adaptation of standard TCP, and requires no special network feedback. Configuration of the bandwidth guarantee is either, statically configured or automatically adjusted by 4CP. The automatic mode aims to be TCP-friendly over appropriately large timescale. We provide analytical results for configuration of the controller parameters. Further, we establish properties of equilibria rates for 4CP automatic and demonstrate feasibility of the design objective through extensive simulations and some Internet experiments.

Keywords—Congestion Control, Service Differentiation, Bulk-Data Transfer, Peer-to-Peer File Sharing, TCP, Low-Priority Emulation, Less Than Best Effort.

1 Introduction

Current Internet provides "best-effort" service that offers no preferential service per application. There have been proposals to upgrade the Internet with service differentiation by the two major proposals discussed at IETF, namely, integrated and differentiated services. The former based on the per-flow reservations at network nodes and latter on defining per-hop behaviors for some service classes. There have been proposals for both premium ("better than best effort") service and "lower than best effort" [5] (e.g. QBone Scavanger [1]). None of these network-centric solutions has achieved a wide-scale usage. Service differentiation of "normal priority" and "low priority" traffic is emulated in practice at the end-points by the protocols such as Microsoft's BITS, used widely for download of software updates with the aim to be non intrusive to user' experience. Similar proposals for low priority emulation were made such as TCP Nice [24], TCP-LP [17], BATS [16], which we discuss later. The common goal of these transport control protocols is to emulate the reference system of two priority classes, high and low, implemented at network nodes by strict priority schedulers that would, essentially, serve low priority traffic only in absence of high priority. This is also the reference system of the lower effort service specified in [5].

The fact is that many file-transfer applications are transfers of large files that are human unattended and last for tens of minutes, hours or even days. The designers of the file transfer applications do want their transfers to achieve good throughputs, and may not have *incentive* to use the transport control protocols that emulate lower than best effort service, as by their very design they may often *starve* for periods of time in presence of any activity along the network path. A common consequence is the preference to use standard TCP for bulk data transfers. For a file transfer using a single TCP connection, then the bandwidth-sharing objective is that of TCP fairness. In the case of a single bottleneck with n TCP connections that all have some common mean round-trip time, TCP fairness mandates allocating a fraction $1/n$ of the link bottleneck to each connection. This presumes this is the only bottleneck for these connections. The problem is that it is now the norm rather than the exception for end users to have several concurrent file transfers (e.g. peer-to-peer file sharing applications or, in general, parallel ftp transfers of large data volumes), resulting in throttling down any other connections to a minuscule TCP fair share of the bottleneck. For concreteness, consider a home user that has several computers at home interconnected with a high-speed LAN and connected to the Internet by a broadband connection. Suppose our user uses a peer-to-peer file sharing application that results in both upload and download file transfers and these may be typically long lasting. Our home user would like her other, (sporadically run) interactive or on-line streaming applications not effected by the presence of long-run bulk data transfers. The user aim would be differentiation of bulk-data transfers such that they achieve appreciable throughput whilst not hurting other traffic.

This paper proposes an end-to-end congestion control that aims to emulate a different reference system than commonly presumed by "strict low priority" protocols [5, 1, 24, 17, 16]. We call the new protocol $4\mathrm{CP}$ (Four 'C' protocol) to signify "Competitive and Considerate Congestion Control". The objective is to provide a specific average rate guarantee to a normal priority connection whenever the bottleneck link can accommodate this. Furthermore, in this case let low priority use the residual bottleneck capacity. If, in contrary, the number of the normal priority connections on the link is larger than can be accommodated by the link, for the specified average bandwidth guarantee, the outcome is to suppress low priority traffic. Note that the strict low priority is a special case of our reference model. In this case, the bandwidth guarantee is set equal to the link capacity, thus starving low priority traffic in presence of any normal priority traffic. The reference system can be seen as a weighted-round robin scheduler

that assigns a link of capacity c as described in Figure 1.

4CP supports fixed or automatically tuned modes for setting the per-connection average bandwidth guarantee for normal priority (TCP). In the fixed mode, the bandwidth guarantee is a configuration parameter set by either user or policy. In the automatic mode, 4CP achieves TCP-fairness over a large timescale. 4CP Automatic is in fact an instance of the family of farsighted congestion controllers introduced in [15]. Hence automatic enjoys all the optimality properties of bandwidth sharing for long-run throughput optimizers as established in [15] as discussed later in the paper.[1] Our findings are summarized as follows:

- We propose 4CP a window based congestion control that emulates the new reference system; it is implemented by a sender-only modification of standard TCP (New Reno); it requires no special network feedback.
- The controller design combines congestion control with detection whether network congestion is high or low. We provide guidelines on setting the control parameters suggested by our analysis results. These include configuring the detector so that false positives are low.
- We provide equilibrium analysis in order to demonstrate benefits in using the new controller with respect to its achieved throughput and induced response times to short-run transfers. These results apply more generally to farsighted controllers, but for simplicity we phrase them for 4CP. We show what best throughput gain can be achieved by a 4CP Automatic compared with the throughput of a long-run TCP connection that both compete for a bottleneck along with short-run transfers. This result is new and adds to the properties found in [15]. The result tells us that the throughput-gain of 4CP in exploiting the fluctuations of network congestion state can be significant.
- We then provide equilibria analysis for single bottleneck with a mixture of 4CP Automatic and TCP like long-run connections that compete with short-run transfers arriving according to a special arrival pattern that we take as a baseline. These results identify cases for which 4CP Automatic induces significantly shorter response times for short-run transfers than if it were TCP. The remarkable property is that 4CP Automatic automatically learns whether competing traffic is short-run or long-run and in extreme cases of low load of short-run transfers treats them as high priority but in the other extreme when short run transfers arrive with large rate does not starve and treat them as long-run.
- We validate the claims suggested by our analysis through extensive simulations in ns2 and complement them with a limited set of experimental results over the Internet obtained by our kernel-implementation of 4CP.

[1] The farsighted controllers aim to optimize their throughputs achieved over long-run, which is in contrast to standard congestion controllers ("myopic") that optimize their short-run throughputs with short-memory about the observed past network congestion state.

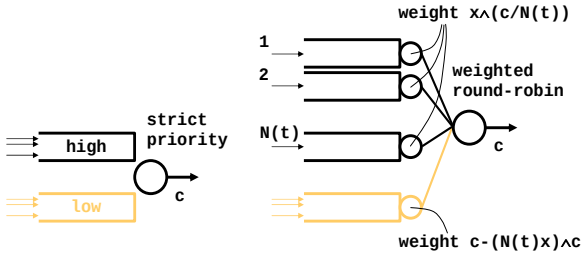


Figure 1: The reference service differentiation systems: (Left) "strict low priority" and (Right) 4CP objective. x is a specified per-connection bandwidth guarantee for normal traffic. Whenever the number of normal priority connections (TCP) is smaller or equal to c/x, each is guaranteed the rate x and the low-priority consumes the residual bandwidth; otherwise, if the number of TCP connections is larger than c/x, they share the link exclusively as they usually do.

1.1 Structure of the Paper

In Section 2, we discuss the principles that underly our proposal and the requirements on the controller design. The protocol is described in Section 3 and followed by the guidelines for configuration parameters setting. Section 4 presents our main analysis results: the maximum throughput gain achievable by 4CP Automatic in Section 4.2; the equilibria for send rates and response times for our baseline arrival of short-run transfers. Simulation and experimental results are shown in Section 5. All our proofs and discussions of some particulars are deferred to appendices.

2 Basic Principles

In this section, we describe design goals of 4CP, for two of its modes: fixed target window and automatic mode. We first provide a description for a single-bottleneck with long-run connections having some common mean round-trip time r, which simplifies exposition of basic principles. Under this homogeneity assumption, we can either consider time-average rate x of a long-run connection that runs a window-based controller or average window $\bar{w}$, by appealing to the mean-value formula $\bar{x} = \bar{w}/r$. We later account for the round-trip time heterogeneity.

2.1 Requirements for 4CP

2.1.1 Fixed Target Window Mode

The goal is to emulate the weighted round-robin reference system in Figure 1–right, for some given reference rate x. The objective of the reference system is further explained in Figure 2. We assume the choice of reference rate is either by a user or by a policy. The latter is preferred in public

environments as letting a user tune this configuration parameter would provide easy means to throttle down downstream traffic by setting the parameter to a small value.

In order to fit better with notation in the rest of the paper, we phrase the design goal in terms of the target window tarw, related to the reference rate x, by $x = \mathrm{tarw}/r$. The reference system assigns rate tarw/r to any normal priority connection, whenever the number of normal priority connections, n, satisfies $n(\mathrm{tarw}/r) < c$, else, it assigns the fair-share rate c/n. The goal is thus to assign the average window

$$w_l(n) = \frac{1}{m} \max\{cr - n \cdot \mathrm{tarw}, 0\}$$

to any low priority connection, whenever there are m of them and there are n normal priority connections. The respective average window for a normal priority connection is:

$$w_h(n) = \min\left\{\mathrm{tarw}, \frac{cr}{n}\right\}.$$

Suppose normal priority connections are adaptive and obey a relation between the average window $\bar{w}$ and loss event rate p, for some positive-valued, decreasing function $f(p)$ on $[0, 1]$:

$$\bar{w} = f(p)$$

For TCP, such relation is well studied. Examples are (i) SQRT formula $f(p) = \sqrt{3/2/b}/\sqrt{p}$, (ii) a simplified version of PFTK [20] formula $f(p) = 1/(\sqrt{2b/3}\sqrt{p} + q3/2\sqrt{3b/2}\sqrt{p^3} + 32\sqrt{p^7})$, where b is the number of packets acknowledged by an acknowledgment (e.g. 2) and q is the ratio of the retransmit timeout value and mean round-trip time.

Enforcing the congestion window tarw to normal priority connections can be seen as enforcing a reference loss event rate tarp. In the prevailing setting, we can interchangeably consider the target window tarw or loss event rate tarp, the two are related by $\mathrm{tarw} = f(\mathrm{tarp})$.

Denoting with p the instantaneous loss event rate observed by a low priority connection, it follows from the above identities that the following holds:

$$\begin{array}{lll} p = \mathrm{tarp} & \text{and} & w_l \geq 0 \\ p > \mathrm{tarp} & \text{and} & w_l = 0. \end{array} \quad (1)$$

The reference loss event rate tarp discriminates network congestion state as either "good" ($p = \mathrm{tarp}$) or "bad" ($p > \mathrm{tarp}$). Note that a low priority connection has a positive window only in good states, and thus the average loss event rate observed by a low priority connection is tarp.

2.1.2 4CP Automatic

We admit the same design objectives as for the fixed target window mode, but instead of arbitrarily fixing the target loss event rate tarp, impose additional constraint:

$$\mathrm{tarw} = \bar{w} \quad (2)$$

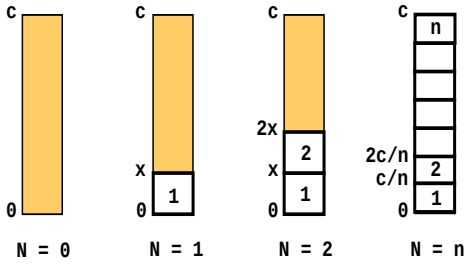


Figure 2: Bandwidth partitioning objective of 4CP reference system. A link of capacity c is shared by normal priority and low priority connections. Each normal priority is given a chunk of bandwidth x, whenever their number N is sufficiently small so that N chunks of bandwidth x can be "packed" into $[0, c]$. The low priority connections are assigned the residual capacity $c - Nx$. Otherwise, when $Nx \geq c$, a normal priority connection is assigned the fair share c/N and low priority connections are assigned no bandwidth.

where $\bar{w}$ is the long-run average window of a low priority connection. The rationale is to make a low priority connection TCP friendly [8, 19] in long-run. Specifically, we admit the definition of a *conservative* [25] control, that says a source of bits is conservative if its throughput, $\bar{x}$ and loss event rate $\bar{p}$ verify $\bar{x} \leq g(\bar{p})$, for a given loss-throughput function g. In view of (2) and noting from (1) that $\bar{p} = \mathrm{tarp}$, the goal is to design a conservative controller that achieves the conservativeness condition, $\bar{x} \leq (1/r)f(\bar{p})$ with equality.

Optimality. The underlying bandwidth sharing objective of 4CP Automatic is optimal for a microeconomics problem that combines both short-run and long-run users, where the latter are assumed to optimize their long-run achieved throughputs. The controller can be seen as a controller that implements the farsighted strategy introduced in [15]. The design of the new controller is a contribution of this paper, but we also establish new equilibria properties in Section 4. We discuss further the connection to microeconomics optimality in Section 6.

3 Protocol

3.1 The 4CP Sender

4CP is a window based controller. Its unique control features are part of congestion avoidance mode. Other modes, such as slow-start, fast recovery, flow control, remain the same as for standard TCP. We now describe the congestion avoidance part of the protocol. The control state comprises: (virtual) window wnd and congestion window cwnd. The parameters use to update the control state are: target loss rate tarp, minimum congestion window mincwnd, maximum congestion window maxcwnd, and minimum wnd value $-$wndbnd. The state $(\mathrm{wnd}, \mathrm{cwnd})$ is updated as follows. Whenever the controller switches to congestion avoidance, $\mathrm{cwnd} = \mathrm{wnd}$.

In congestion avoidance, $(\mathrm{wnd}, \mathrm{cwnd})$ are updated upon following events:

If ack:

$$\mathrm{wnd} \quad \leftarrow \quad \min(\mathrm{wnd} + 1/\mathrm{cwnd}, \mathrm{maxcwnd}) \qquad (3)$$

If triple-dupack:

$$\mathrm{wnd} \quad \leftarrow \quad \max(\mathrm{wnd} - 1/(\mathrm{tarp} \cdot \mathrm{cwnd}), -\mathrm{wndbnd}) \qquad (4)$$

In either case:

$$\mathrm{cwnd} \quad \leftarrow \quad \max(\mathrm{wnd}, \mathrm{mincwnd}) \qquad (5)$$

Rationale. Consider the control by ignoring the reflections at the boundaries $\mathrm{mincwnd}$ and $\mathrm{maxcwnd}$ and assume $\mathrm{cwnd} \geq 0$, i.e. consider $\mathrm{cwnd} \leftarrow \mathrm{cwnd} + 1/\mathrm{cwnd}$ per received acknowledgment and $\mathrm{cwnd} \leftarrow \mathrm{cwnd} - 1/(\mathrm{tarp} \cdot \mathrm{cwnd})$ per triple-duplicate acknowledgment. The former amounts to incrementing the congestion window for 1 segment per round-trip round in absence of congestion indication, presuming no delayed acknowledgements (or $1/2$ segment, if acknowledgements are delayed). Suppose the latter occurs with rate $\mathrm{cwnd}p$, for some instantaneous loss event rate p. Then, the drift of the congestion window is $1 - p/\mathrm{tarp}$. Hence, either $p = \mathrm{tarp}$ and congestion window takes some positive value or $p > \mathrm{tarp}$ and $\mathrm{cwnd} = \mathrm{mincwnd}$ as the drift is strictly negative.

wnd, *congestion window and detector*: The wnd has a dual role. First, if $\mathrm{wnd} \geq \mathrm{mincwnd}$, $\mathrm{wnd} = \mathrm{cwnd}$, wnd is in fact used as congestion window. Second, $\mathrm{wnd} < \mathrm{mincwnd}$, wnd has a role of a detector of bad phase. In the latter case, $\mathrm{cwnd} = \mathrm{mincwnd}$, and thus in the wnd updates, we can replace $\mathrm{tarp} \cdot \mathrm{wnd}$ with $\mathrm{tarp} \cdot \mathrm{mincwnd}$. The detector indicates phase is bad whenever $\mathrm{wnd} < 0$. Increments of wnd and cwnd are additive increase over round-trip rounds, same as with standard TCP. The decrement is specific to 4CP.

3.1.1 4CP **Automatic**

Automatic mode adapts the reference loss-event rate tarp per each received acknowledgment as:

$$\mathrm{tarp} \quad \leftarrow \quad \mathrm{tarp} + a(f(\mathrm{tarp}) - \mathrm{cwnd})/\mathrm{cwnd} \qquad (6)$$

where a is a gain parameter; a small constant after a sufficient number of iterates, and otherwise specified in Section 3.3.3.

Rationale. Taking a small constant $a \ll 1$, by time-averaging argument, (6) aims to the balance:

$$\bar{w} = f(\mathrm{tarp})$$

where $\bar{w}$ is average window sampled over round-trip time rounds. This is precisely the condition (2).

3.2 Receiver

4CP implements no special functionalities at a receiver, and thus can use any standard TCP receiver socket.

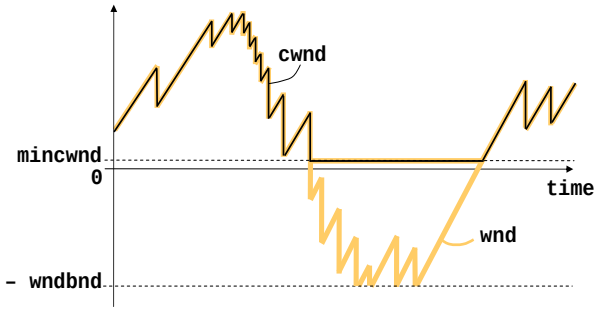


Figure 3: 4CP window control elements: (i) additive-increase (as TCP) and inverse-decrease (4CP specific), and (ii) window wnd extended to negative values. Whenever $\mathrm{wnd} \geq \mathrm{mincwnd}$, $\mathrm{cwnd} = \mathrm{wnd}$ and thus wnd is used as congestion window in a standard way. In the other case, $\mathrm{wnd} < \mathrm{mincwnd}$, wnd turns out to be a detector of bad phase and throughout bad phase congestion window cwnd is fixed to minimum congestion window $\mathrm{mincwnd}$.

3.3 Control Parameter Settings

3.3.1 Minimum Congestion Window $\mathrm{mincwnd}$

Ideally, when congestion state is in bad phase, 4CP should not be sending data at all, and thus $\mathrm{mincwnd}$ should be set to 0. But this in practice is infeasible as the controller needs to continue sensing the congestion state and thus must send some little probe data for inference, and for this reason $\mathrm{mincwnd}$ is set to a positive value. Now, the smaller the $\mathrm{mincwnd}$, the smaller the send rate in a bad phase. In our 4CP implementation by adaptation of a TCP sender, we set $\mathrm{mincwnd}$ to 2 segments in order to prevent timeouts due to Nagle's algorithm. This could be further refined to achieve lower send rates in bad phase than 2 segments per round-trip time round, and is left open for future study; similar techniques such that deployed for [16] could be used.

3.3.2 Bad Phase Detector

We have two conflicting goals to set wndbnd to either a small or a large positive value. The former setting is desirable in order to have a *quick* detector that will take a small number of round-trip time rounds to switch from indicating phase is bad to phase is good, presuming such a transition did happen. The latter, though, is desirable to have small false positives, i.e. the detector indicates incorrectly a transition from bad to good phase. Our problem can be seen as a sequential hypothesis testing problem, known as change point detection, which deals with the trade-off between quick and reliable detection.

In our definition of the bad phase detector wnd (3)–(4), we imposed a lower bound $-\mathrm{wndbnd}$. We need to impose such a bound as in its absence, if for a long time the phase happened to be bad, then wnd will tend to excessively negative values as its drift is strictly negative. In a hypothetical

case, wnd will converge to $-\infty$. For this reason, we need a finite lower bound on wnd.

Claim 1. *The fraction of time of false positives decreases exponentially with the boundary parameter* wndbnd.

The claim is supported by analysis result in the rest of this section and is further validated by simulations in Section 5. The result of Theorem 1 suggests setting the configuration parameter wndbnd as displayed in (9).

Our detector does a sequential hypothesis test with null hypothesis: phase is bad, i.e. loss rate $> \mathrm{tarp}$. Our goal is to estimate what fraction of time the detector indicates phase is good, under the null hypothesis. These are false positives. We denote with $N(0,t]$ the number of loss events observed on a time interval $(0,t]$. We have to impose assumptions on the process of loss events to carry analysis further. To that end, we assume loss events appear at points of a Poisson process with intensity $\lambda(t) = \mathrm{cwnd}(t)p$, for a fixed loss rate $p > \mathrm{tarp}$. Note that by the design, $\mathrm{cwnd}(t) = \mathrm{mincwnd}$, whenever $w(t) \leq \mathrm{mincwnd}$. This motivates to consider the following dynamics of the window, $\mathrm{wnd}(0) \geq -\mathrm{wndbnd}$ and for $t \geq 0$:

$$\mathrm{wnd}(t) = v(0) \vee \sup_{s \leq t}\{(t-s) - cN(s,t] - \mathrm{wndbnd}\} \quad (7)$$

where $v(0) := \mathrm{wnd}(0) + t - cN(0,t]$, $c := 1/(\mathrm{mincwnd} \cdot \mathrm{tarp})$ and loss events appear as points of a homogeneous Poisson process in time with rate $\mathrm{mincwnd} \cdot p$. The dynamics captures the linear increase of the window over round-trip rounds (in the absence of loss events). They also capture fixed decrements upon loss events whenever the window is less than equal to $\mathrm{mincwnd}$.[2]

Theorem 1 (False Positives). *Suppose* $p/\mathrm{tarp} = r > 1$, *i.e. phase is bad. The long-run fraction of time the detector* $\mathrm{wnd}(t)$ *indicates false positives is:*

$$f \quad = \quad e^{-\lambda a \cdot (\mathrm{mincwnd} + \mathrm{wndbnd})}$$

where $\lambda := \mathrm{mincwnd} \cdot \mathrm{tarp} \cdot r$ *and* a *is the solution of*

$$1 - a = e^{-ar}. \quad (8)$$

Configuration guideline. The result suggests to set the control parameter wndbnd as:

$$\mathrm{wndbnd} = \frac{1}{\mathrm{mincwnd} \cdot \mathrm{tarp}} \frac{\log\left(\frac{1}{f}\right)}{ar} \quad (9)$$

for some fixed $r > 1$, with the aim to bound the fraction of time of false positives to f, whenever the loss rate is larger than tarp for a fixed factor r. Note that the boundary wndbnd adapts over a large timescale through the adaptation of tarp. We suggest to lower bound the value (9) to a sufficiently large value so as to ensure quick detection of the bad to good phase transition.

CUSUM optimality. The detector is closely related to optimum change point detection known as CUSUM. See Appendix B.

3.3.3 Gain of Target Loss Rate Adaptation

We want to set the adaptation gain parameter a of the target loss rate tarp (6) to a small value so that tarp is virtually constant. On the other hand, the adaptation gain a should not be too small as then it would take a long time for tarp to converge to equilibrium. These are two conflicting goals. In our design, the adaptation gain is chosen to be initially large and decreases with the number of the updates of tarp to a small value used for the rest of the transfer. In particular, we used a linearly decreasing function with the number of the iterations. The rationale is to set the initial tarp to the instantaneous loss rate by fast adaptive learning and then eventually let it run as prescribed by (6) with fixed adaptation gain.

4 Performance

In this section, we present our main performance analysis results, which we validate in Section 5.

4.1 Stability

Stability results established in [15] imply global asymptotic stability of 4CP like controller, formulated in a standard dynamical systems form, with convergence to equilibria that is a global optima of an underlying utility-maximization problem, under the assumption of zero feedback delays. We demonstrate stability through extensive ns2 simulations with RED bottleneck. Note that as for most congestion controllers, it is important that the bottleneck provides equal loss event rates per packet over competing connections.

We emphasize the following particular claim for network paths over which over duration of a long-run transfer, the loss rate fluctuates in the neighborhood of an operating point:

Claim 2. *Competing* 4CP *Automatic and TCP long-run connections for a bottleneck that exhibits "one-phase" and equalizes loss rate per packet over competing connections, achieve comparable throughputs.*

The claim is suggested by the analysis in [15] and we validated through extensive simulations. This suggests inter-protocol fairness between 4CP Automatic and TCP with comparable mean round-trip times in the cases when network congestion state fluctuates around a single equilibrium loss rate.

4.2 Maximum Throughput Gain of 4CP Automatic

The equilibria analysis in [15] suggests that the throughput of a long-run 4CP Automatic connection would not be smaller

[2] A more detailed analysis would account for the fact that for $\mathrm{wnd}(t) > \mathrm{mincwnd}$, the decrement of $\mathrm{wnd}(t)$, $1/(\mathrm{cwnd}(t)\mathrm{tarp}) \leq 1/(\mathrm{mincwnd}\mathrm{tarp})$, and the intensity of loss events $\mathrm{cwnd}(t)p \geq \mathrm{mincwnd} \cdot \mathrm{tarp}$. However, our estimate would already provide a good accuracy.

than that of a competing long-run TCP connection, in any case. Thus, 4CP Automatic should be able to gain throughput over a competing long-run TCP connections by utilizing the fluctuations of network congestion state. However, it appears unknown whether this throughput gain can be of any significant value to be of practical relevance.

Claim 3. 4CP *Automatic can yield significant throughput benefit over a competing long-run TCP connection by leveraging fluctuations of network congestion state.*

The claim is suggested by our main analysis result that identifies the best possible throughput gain of 4CP Automatic [3] to the full extent for a single bottleneck, and moreover, identifies the fluctuations of network congestion state under which the maximum throughput gain is achieved. The fluctuation of network congestion state is, in the sequel, for concreteness, phrased in terms of the number of competing short-run transfers.

Consider a single link with a capacity $c > 0$. Suppose two long-run connections compete for the link; one 4CP Automatic and one TCP. There are also short-run connections arriving at the link. Their arrivals are subject only to the condition that there are at most n of them at any time and their number over time has a stationary distribution π. Suppose the idealistic setting where all connections achieve their equilibrium rates and this happens instantly for all.[4]

Theorem 2 (Extreme). *The respective throughputs of* 4CP *Automatic and TCP,* $\bar{x}_{4\mathrm{CP}}(\pi)$ *and* $\bar{x}_{\mathrm{TCP}}(\pi)$*, satisfy:*

(i) Throughput-gain upper-bound:

$$\frac{\bar{x}_{4\mathrm{CP}}(\pi)}{\bar{x}_{\mathrm{TCP}}(\pi)} \leq \frac{1}{2}\left(\sqrt{n} + \frac{1}{\sqrt{n}}\right), \quad \textit{all } \pi.$$

(ii) Achievability: maximum throughput-gain is achieved for the extremal distribution of phases π^* *that concentrates all its mass on the end-point phases* 0 *and* n*:*

$$(\pi^*(0), \pi^*(n)) = \left(\frac{1}{\sqrt{n}}, 1 - \frac{1}{\sqrt{n}}\right).$$

The proof (in Appendix) is based on the equilibria analysis and maximization of the throughput ratio over all distributions of phases π on the finite set 0 to n. To reiterate, it is remarkable that the result holds under quite some generality for any distribution of phases of network congestion state over a set as specified and for a farsighted controller, in general.

Commentaries. The result of the theorem tells us that a farsighted controllers such as 4CP Automatic can yield significant throughput gains from fluctuations of the network congestion state. The result tells more. It says that best possible throughput gain is achieved for long epochs of bad phase and short epochs of good phase. The durations of the respective durations are proportional to $1-1/\sqrt{n}$ and $1/\sqrt{n}$, respectively.

If there is at most 1 short-run connection at any time, i.e. $n = 1$, then $\bar{x}_{4\mathrm{CP}} = \bar{x}_{\mathrm{TCP}}$. On average, a good phase lasts longer than a bad phase for $n = 2, 3$; they have equal mean durations for $n = 4$; and otherwise, for $n > 4$, a good phase lasts less than a bad phase. For large n, the mean duration of a good phase [resp. bad phase] is of order 1 [resp. order $\sqrt{n}$], so the maximum throughput gain is achieved for alternations of long-lasting bad phases and short-lasting good phases.

[3] The result applies more generally to farsighted congestion controllers.

[4] In reality, there would be transient phases, which we account for in our simulations.

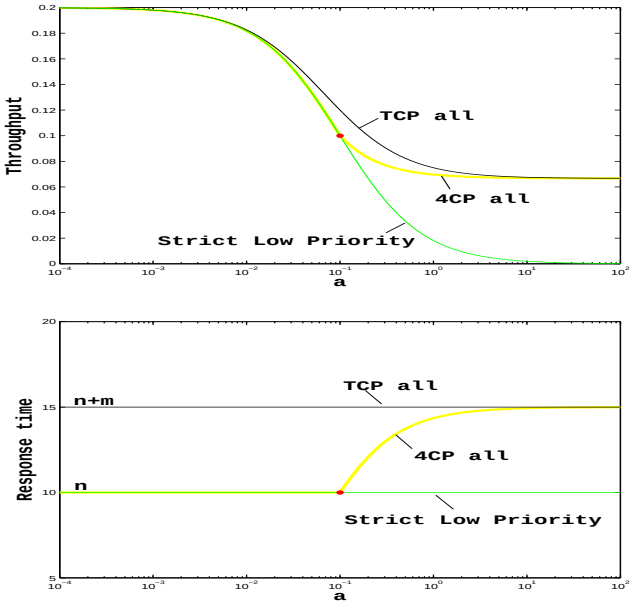


Figure 4: Two-phase baseline: (top) throughputs for the system with m long-run connections using TCP ("all TCP") or all using 4CP ("all 4CP") or served as strict low priority with equal share within low priority class; (bottom) same but showing response times of short-run connections. The plots are for $(n, m) = (10, 5)$. For asymptotically small a, all the systems have the limit rate for long-run connections equal to c/m. "all 4CP" behaves precisely as strict low priority for $a \leq 1/m - 1/n$, otherwise, it achieves larger rate with the same asymptote as for "all TCP", $c/(n+m)$. In the former case, the response time for short-run connections with "all 4CP" is same as that of strict low priority background, n, and in the latter case, it gradually increases with a to that of "all TCP", $n + m$. 4CP is not "starved" as a gets large, in contrast to strict low priority.

4.3 Two-Phase Baseline

The goal in this section is to pose a baseline case for arrival of short-run transfers to evaluate the equilibria send rates for long-run transfers and response times for short-run transfers. Our baseline case is defined as arrival of short-run transfers in batches of n file transfers, with file sizes over batches and

idle times between successive batches being a stationary random sequence. This choice for the arrival of short-run transfers is made for tractability reasons and is motivated by the extremal property found in the earlier section. The baseline case allows us to prove existence of cases for which 4CP Automatic reduces significantly, the response times of short-run transfers, than if it were TCP. We first highlight a claim suggested by the results in the sequel of this section, which we validate by simulations.

Claim 4. *4CP Automatic can induce the mean response time for short-run transfers that is smaller than if 4CP were TCP, and this can be as large as a factor 2.*

4.3.1 Send Rate Equilibria

We consider a single bottleneck of capacity $c > 0$ for which m long-run connections compete, out of which k are 4CP Automatic and $m - k$ TCP. The short-run connections arrive in batches so that there are n file transfers in a batch i, each of size F_i. The time between departure of a batch i and arrival of batch $i + 1$ is called idle, and denoted with τ_i. The sequence (F_i, τ_i) is assumed to be stationary and ergodic. The assumption accommodates a broad set of alternating process, and, in particular, note that stochastic dependences of the transfer and idle epochs are allowed. We denote with f and τ, the respective means of file sizes and idle times and assume both are finite. For a system with k 4CP connections, we denote with $\bar{x}_{\rm TCP}(k)$ and $\bar{x}_{\rm 4CP}(k)$ the respective per-connection throughputs of TCP and 4CP Automatic, and with $r(k)$, the mean file transfer time ("response time") of the short-run transfers. The load of the short-run transfers is captured by the parameter $a := f/(\tau c)$, which can be interpreted as the ratio of the mean transfer time and mean idle time, attained if all the link capacity was assigned to a short-run transfer. We first specify the equilibria for the send rates of long-run transfers.

Theorem 3 (Two-phase). *Let $u^*(a,k) := k/(1-ak)$. If $n \geq u^*(a,k)$, phase n is bad, else good.*

1. *For $k = 0$, i.e. all TCP case:*

$$\frac{\bar{x}_{\rm TCP}(0)}{c} = \frac{1}{m}\left(1 - \frac{an}{1+a(n+m)}\right).$$

2. *If $k \geq 1$ and phase n is bad:*

$$\frac{\bar{x}_{\rm 4CP}(k)}{c} = \frac{1}{m+ank}$$

and

$$\frac{\bar{x}_{\rm TCP}(k)}{c} = \frac{1}{1+an}\left(\frac{1}{m+ank} + a\frac{n}{m-k+n}\right).$$

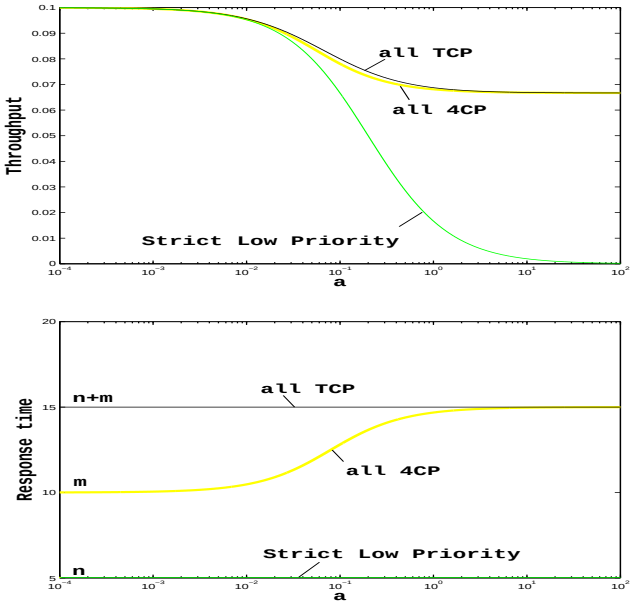


Figure 5: Same as in Figure 4, but $(n, m) = (5, 10)$. Phase n is now always good.

3. *Else if $k \geq 1$ and phase n is good:*

$$\frac{\bar{x}_{\rm 4CP}(k)}{c} = \frac{\bar{x}_{\rm TCP}(k)}{c} = \\ = \frac{1}{2m}\left(1 - a(n+m) + \right. \\ \left. + \sqrt{(1-a(m+n))^2 + 4am}\right).$$

Interpretations. Condition $n \geq u^*(a,k)$ that phase n is bad is equivalent to: $a \leq 1/k - 1/n$. Thus phase n is bad only if $k < n$. In the latter case, for fixed k and n, phase n is bad if the "load" of short-run transfers, a, is sufficiently small. For a fixed a and n, phase n is good if k is sufficiently large.

In the "all TCP" case, for the two limit cases, a small and a large, $\bar{x}_{\rm TCP}(0) \approx c/m$ and $\bar{x}_{\rm TCP}(0) \approx c/(n+m)$, respectively. The former is as short-run transfers do not exist, while the latter as they were long-run.

We now use the equilibria rates established in this section to gain insight on the response times of short-run transfers imposed by TCP and 4CP Automatic.

4.3.2 Response Times

We use as a benchmark, long-run connections that perfectly emulate strict low priority. In this case, whenever phase is n, a short-run transfer is allocated the rate c/n. The per-connection throughput of long-run transfers, denoted with $\bar{x}_{\rm LP}$, is equal to $\pi(0)c/m$, with $\pi(0) = 1/(1 + a/(1/n))$. It follows:

$$\frac{\bar{x}_{\rm LP}}{c} = \frac{1}{m}\frac{1}{1+an}.$$

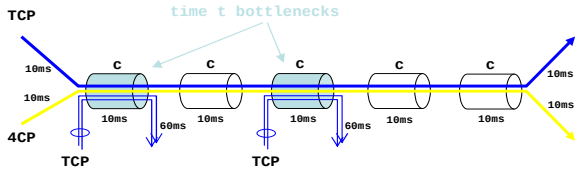


Figure 6: Multi-hop scenario.

Indeed, $\bar{x}_{\rm LP} \to 0$, as $a \to +\infty$, i.e. as the load of short-run transfers tends to be large, the low priority connections get "starved".

We now state the mean response time for systems with the long-run connections served according to a policy σ (either "all TCP" or "k-4CP" or "LP"). These are claimed for a file of unit length, so that the response time is $r_s = 1/x_s$, where x_σ is the per-connection rate of short-run transfers under a policy σ.

Properties. The response times for individual bandwidth sharing policies are: all TCP: $r_{\rm TCP}(n,m,a) = n+m$; strict low priority: $r_{\rm LP}(n) = n$; all 4CP: $r_{\rm 4CP}(n,m,k,a) = n$, for $a \le 1/k - 1/n$, $= 1/\bar{x}_{\rm 4CP}$, for $a > 1/k - 1/n$,

The following proposition supports Claim 4, which is further validated by simulations. The result is of interest as it identifies cases where significant reduction of response times for short-run transfers is provided by 4CP.

Proposition 1 (Best Response Time). *For the prevailing two-phase baseline:*

1. *The response time with k 4CP long-run connections is at least, for all $a > 0$,*

$$r_{\rm 4CP}(n,m,k,a) \ge \begin{cases} n & k < n \\ m & k \ge n. \end{cases}$$

The equality is achieved asymptotically as $a \to 0$.

2. *For all 4CP case, i.e. $k = m$, $r_{\rm 4CP}(n,m,m,a) \ge n \vee m$, and the following best possible reduction of the response time compared to the all TCP case holds:*

$$r_{\rm 4CP}(n,m,m,a) \ge \frac{1}{2} r_{\rm TCP}(n,m)$$

for all $a > 0$ and $n, m \ge 1$.

Achievability: take $n = m$. For any $\epsilon > 0$, there exists $a_0 > 0$ such that for all $a \le a_0$:

$$r_{\rm 4CP}(n,m,m,a) \le \left(\frac{1}{2} + \epsilon\right) r_{\rm TCP}(n,m).$$

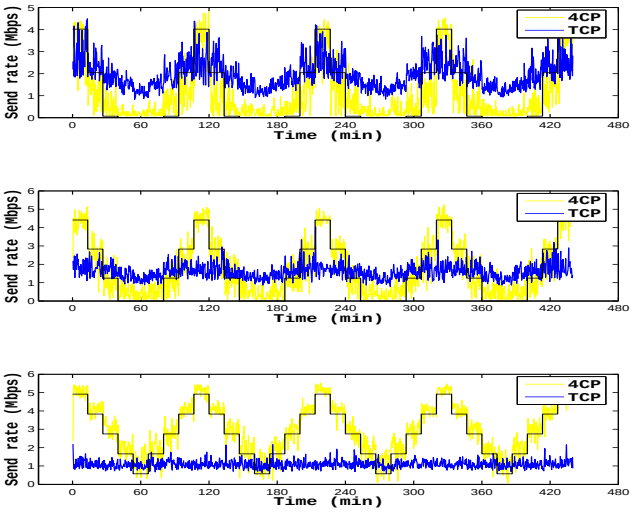


Figure 7: Bandwidth partitioning with 4CP.

5 Experimental Evaluation

We performed an extensive set of ns2 simulations to validate our proposal and the claims that we made. These are presented first and then followed by a description of our kernel implementation along with a sample of results from our experiments over the Internet with the specific goal to demonstrate the benefits to competing short-run transfers.

5.1 Simulation Results

Network configurations We consider both single and multi hop topologies:

- Single-hop: scenario is a standard dumb-bell topology, with the bottleneck queue using RED [10]. RED parameters are set as follows: the queue limit is 150 packets, thresh=20 packets, maxthresh =60 packets, and linterm= 40. The ECN option is turned off. For the 4CP parameter setting, $winbnd = 180$ packets in all the simulations. We do both homogeneous and heterogeneous RTTs simulations. For homogeneous scenario, all users have propagation RTTs being 100 ms; and for heterogeneous scenario, the RTTs vary from 30 ms to 120 ms.

- Multi-hop: we consider a standard "linear-network" of n links with connections being either multi-hop, traversing all the links 1 to n or single-hop traversing a single link; see Figure 6.

"Pyramid" short-run transfers. In several simulations we use a "pyramid" of short-run transfers specified as follows. The number of short-run connections is taken as a periodic function over time. The number of short-run trans-

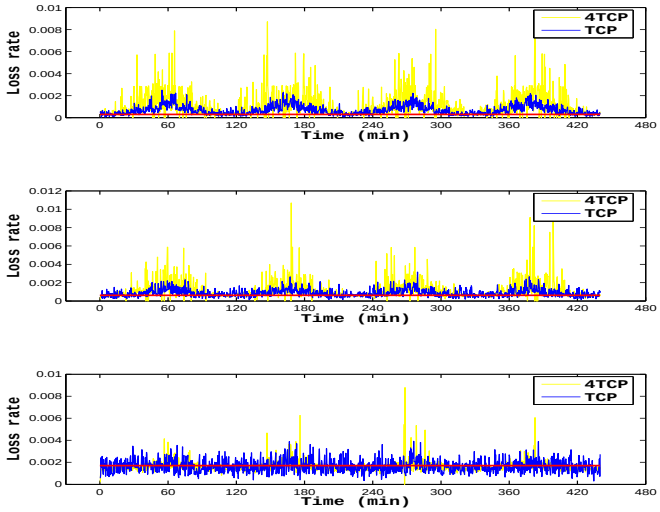


Figure 8: Same as in Figure 7 but showing loss rates.

fers over a period of duration T, given there are at most h short-run transfers over a period, is constructed by initiating a kth short-run transfer within a period $[0, T]$ at time $kT/(2h)$ and terminating this short-run transfer at time $T/(2h)(2h+1-k)$, for $k = 1, 2, \ldots, h$. We use such short-run transfers as it let us control and separate the individual $h + 1$ phases. For this set of results, durations of short-run transfers are fixed.

Bandwidth partitioning objective. We demonstrate the underlying service differentiation model of 4CP, which we illustrated in Figure 2. The setup is single-hop with four pyramid short-run transfers and two long-run transfers, one 4CP and one TCP. Figure 7 shows the send rates over time of the long-run 4CP and TCP for three distinct choices of the target loss rates of $0.0003, 0.000618, 0.0017$, which correspond to the target TCP send rates of $2.3805, 1.6586$, and 1 Mb/s. The long-run TCP achieves the configured target rate whenever the number of TCP flows is not larger than can be accommodated over the link, otherwise, 4CP sends with a small rate and almost the entire link is shared by TCP. Figure 8 shows the corresponding loss rates. It demonstrates equalization of the loss rate to target loss rate over good phases and that otherwise the loss rate exceeds the target loss rate. The analogous results, but for 4CP Automatic, are showed in Figure 9. Initial tarp values are set to be larger than the equilibrium value in the top sub-figures and smaller in the bottom sub-figures. No matter how we set the initial tarp value, tarp converges to its equilibrium value after some periods and the send rate of the TCP and $4CP$ users approximate their corresponding equilibrium values, as can be shown in the first sub-figure in Figure 10, which contains

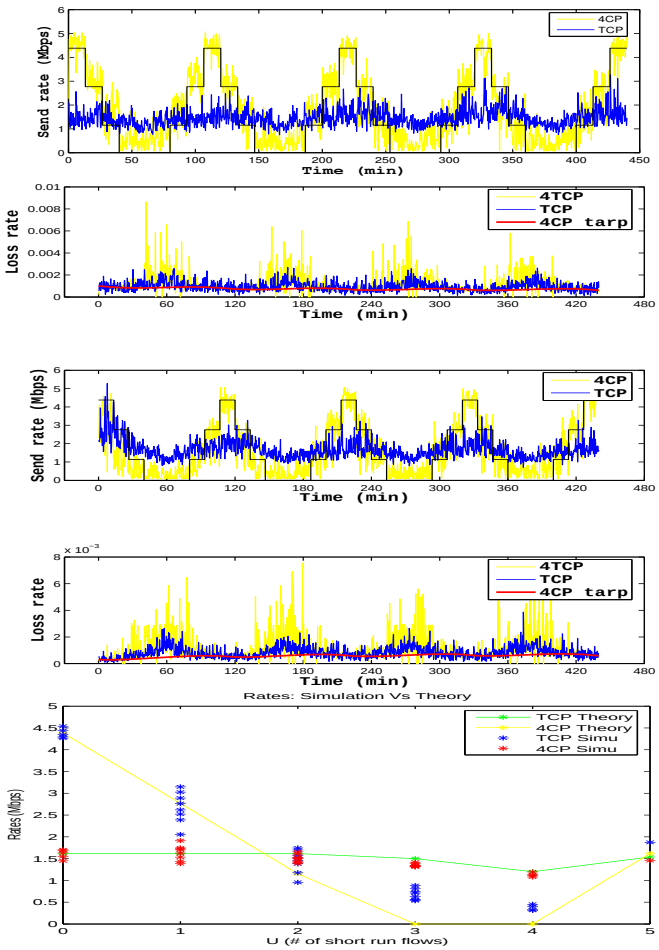


Figure 9: Same as in Figure 7 and Figure 8 but for 4CP automatic. Initial tarp for plots in first two rows is equal to 0.001 (resp. 0.0003 for the third and forth row plots). The fifth row plot: average send rate vs. analytical equilibrium value for each phase (1 to 4) and over entire duration.

the comparison of the measured send rates and their analytical equilibrium values for the two long-run and six short-run setup above and three other setups with four long-run and eight short-run transfers. The window size, send rate and loss rate plots for the three other setups are similar to Figure 9 and are omitted here. If the RTTs are different, in the good phases, the window sizes of the TCP users achieve the target window size value, which is a function of the target loss rate. The rates are inversely proportion to RTTs. The result for the heterogeneous scenario is shown in Figure 13. For all the simulations above, the phase length is 800 seconds. We also performed the pyramid simulations with phase length varied from several seconds to ten minutes and compare the measured average phase length with their analytical equilibrium values for different phases. The result is shown in Figure 14 and we can see that the equilibrium send rates are achieved

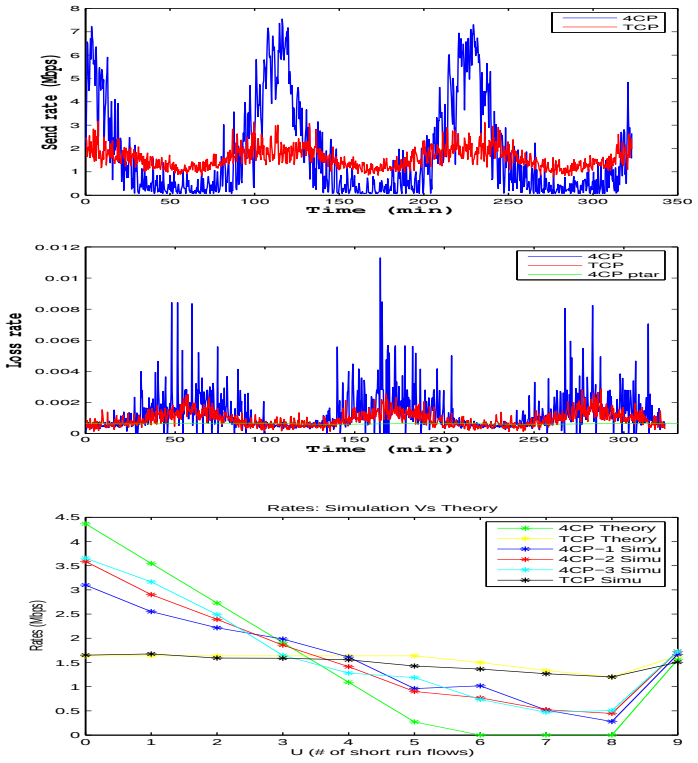


Figure 10: The send rate, loss rate, and average send rate vs. analytical equilibrium value. 1 4CP-Automatic, 3 TCP long-run and 8 short-run connections. (Bottom) average over each phase for $U = 1-8$, and average over the entire duration for $U = 9$.

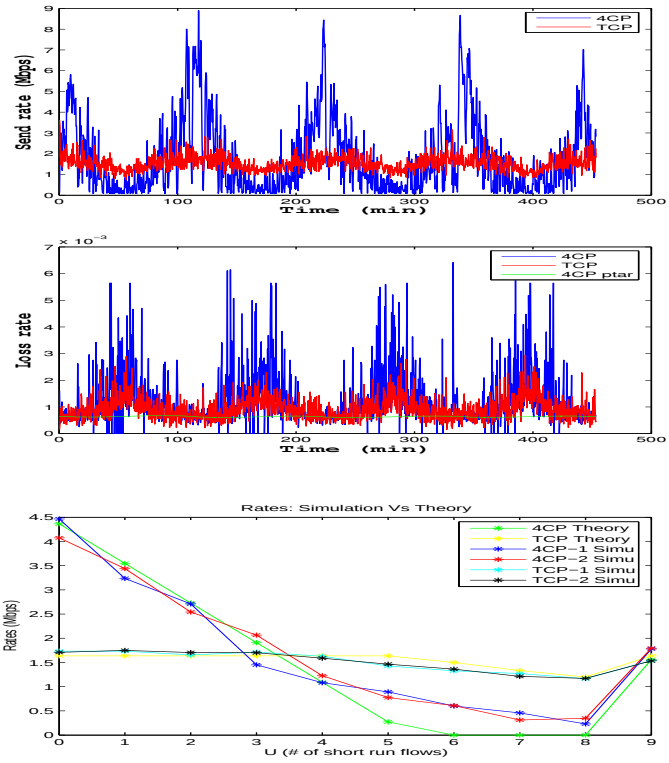


Figure 11: Same as in Figure 10, but 2 4CP-Automatic and 2 TCP long-run connections.

over a wide timescale of phase fluctuations, from sub minute to ten minutes.

One phase claim. We now validate our Claim 2 that says 4CP Automatic and TCP should receive approximately the same throughputs over a bottleneck with loss rate fluctuating around some equilibrium in one phase. We run our single-hop scenario for the number of long-run connections ranging from 2 to 200, out of which a fixed fraction are 4CP Automatic as specified in Figure 15. In the results (i) throughputs of TCP are not affected in any significant manner by the presence of 4CP Automatic connections, (ii) the same holds for the mean round-trip times, and (iii) all connections experience approximately the same loss rate with some notable difference under heavy loss.

Detector false positives claim. The setup is single-hop with one 4CP and the number of TCP connections as specified in Figure 16. We designed experiments so that the loss rate is larger than the fixed target loss rate 4CP, so that phase is bad. We then estimate the fraction of round-trip time rounds wnd is larger than mincwnd, which is an estimate of f in Theorem 1. We observe in Figure 16 that the fraction of time of false positives decreases exponentially with the configuration parameter wndbnd, which validates the assertion of Claim 1.

4CP **over a sequence of bottlenecks**. We designed simulation experiments to demonstrate that 4CP distinguishes good (non congestion) from bad (congestion) phases over a network path, not necessarily over a single bottleneck. To that end, we consider the multi-hop scenario with two long-run multi-hop connections traversing all the links; one 4CP Automatic and the other TCP. On each there are either 0 or n single-hop TCP connections at any time. A link with n single-hop connections at a time is a bottleneck, otherwise not. A phase m corresponds to a case with m bottlenecks. The number of bottlenecks is varied from 0 to the total number of the links as in our single-hop "pyramid" example. The results in Figure 17 show that again 4CP distinguishes good phases from bad ones and perform as predicted by the equilibrium analysis.

Maximum throughput-gain claim. The setup is single-hop with one TCP and one 4CP Automatic long-run connection. The short-run transfers arrive as alternating sequence of instantaneous batches of n file transfers followed by idle time. The period duration is $T = 200$ seconds. The short-run transfer time lasts for $(1 - 1/\sqrt{n})T$ and idle time for $T/\sqrt{n}$

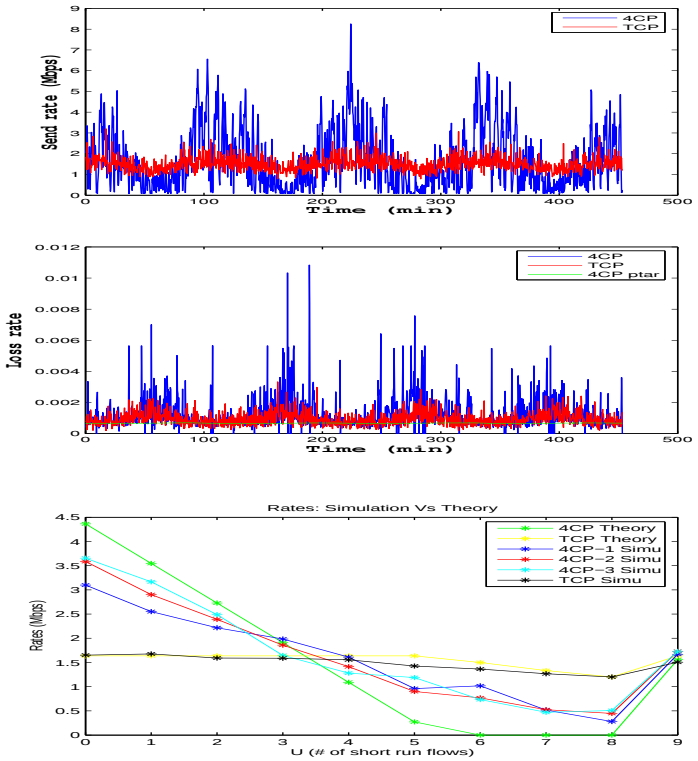


Figure 12: Same as in Figure 10, but 3 4CP-Automatic and 1 TCP long-run connection.

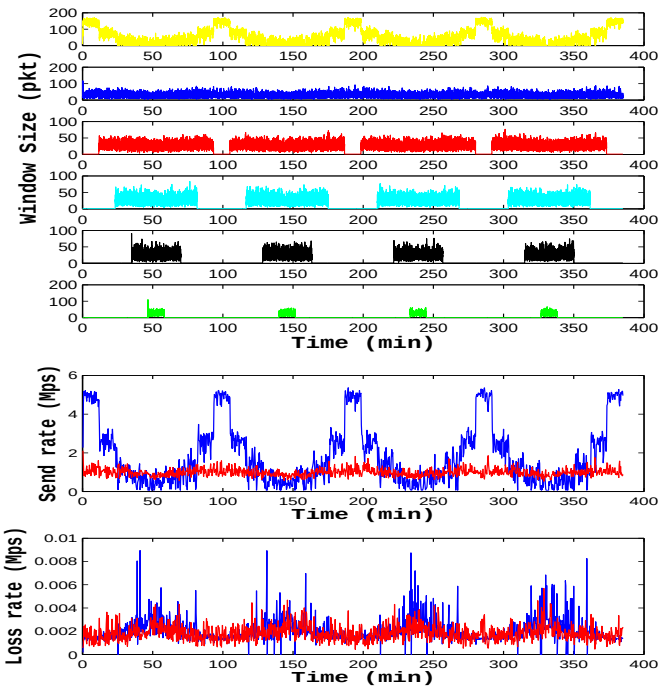
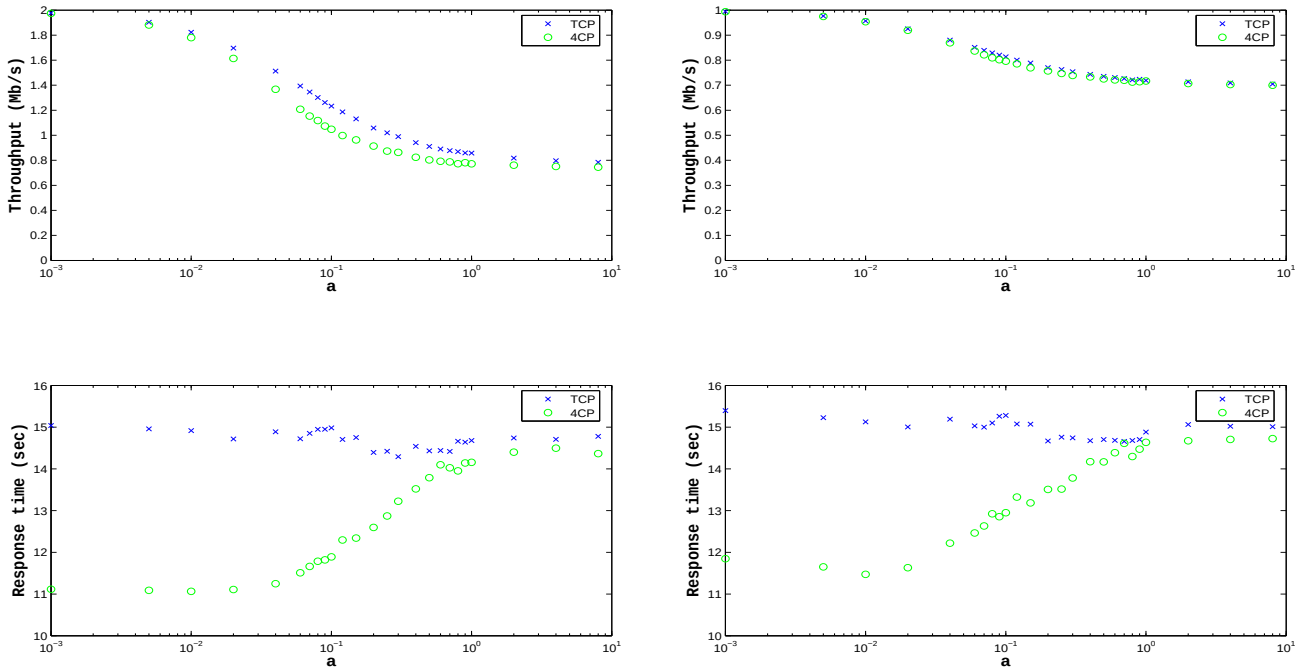


Figure 13: Same as in Figure 7 and Figure 8 but for heterogeneous RTT that ranges from 30 ms to 120 ms. The send rate and loss plots are for the long-run connections. For window plots, from top to bottom, the six plots are for 4CP, TCP, short-run 30 ms, short-run 60 ms, short-run 90 ms, and short-run 120 ms.

seconds. Figure 18 shows the 4CP to TCP throughput ratios for $n = 2, 3, \ldots, 7$. The results demonstrate achievability of the bound in Theorem 2–item 2. We also performed simulations for short-run TCP transfers that arrive and leave at random instants. The throughput ratios obtained in these cases are indeed bounded by the bound of Theorem 2, which supports item 1 of the theorem.

Response times claim for baseline case. We designed a single-hop simulation scenario mimicking the equilibrium rates and response times as predicted by our analysis for the baseline case, depicted in Figure 4 and Figure 5. The results are showed in Figure 19 and they do exhibit very good conformance to the predictions of our analysis. On one end, for sufficiently small load of short-run connections, their response times are almost as if 4CP Automatic were strict low-priority. On the other end, for sufficiently large load of short-run connections, it is as they were long-run. We also performed simulations to validate Claim 4 and Proposition 1, and the result is shown in Figure 20. The setup is single-hop scenario with baseline short-run connections. The sum of the number of transfers per batch for short-run connections, n, and the number of long-run connections, m, is fixed to 20. The long-run connections are either all TCP or all 4CP Automatic. Our analysis predicts that the mean response time for short-run transfers cannot be smaller than $n \vee m$ and this is achieved for $n = m$ and for asymptotically small load of the short transfers. These assertions are validated in Figure 20. For both Figure 19 and Figure 20, the response time for the short transfers with 4CP is larger than analytical results. The following reasons explain this discrepancy. First, 4CP's send rate is zero in bad phases in theory, but it is non-zero in the packet level implementation. In the bad phases, a 4CP user has to keep sending packets to detect the end of the bad phase, and the minimum window size is set to be 2 in the simulations, which causes the non-zero send rate in bad phases. If the false positive probability is considered, the average send rate is even larger. The second reason is the transient phase. In theory analysis, we assume that the convergence is immediate and 4CP stops competing with short-run traffic immediately after the bad phase begins. However, it takes some time for the 4CP user to reduce the window size and converge to a small send rate phase. These two factors combined make the short traffic average response time larger than analytical values.

Impact on web traffic. The web traffic simulation is: we choose the same set up as in [17][Figure 14-15] and [16][Fig-



$(n, m) = (10, 5)$ $(n, m) = (5, 10)$

Figure 19: Simulation companions to Figure 4 (left) and Figure 5 (right). n is the number of short-run connections in a batch and m is the number of long-run connections. The long-run connections are either all TCP or all 4CP. Short-run file sizes are i.i.d. with exponential distribution with mean 1.25 MB. The link capacity is 10 Mb/s. In this case, $a = 1/\tau$, where τ is the mean time between departure and arrival of short-run file transfer batches; the x-axis on the plots can thus be interpreted as $1/\tau$. The response time of short-run connections, in absence of long-run connections, is 10 sec. We see that 4CP yields the response time to short-run transfers as it were strict low-priority, whenever $\tau \geq 10$ sec. At about $\tau = 1$ sec, 4CP treats short-run transfers as they were long-run.

ure 8]. In this model, clients initiate sessions from randomly chosen web servers with several web pages downloaded in each session. Each page contains several objects, and these objects are delivered sequentially by different TCP connections (HTTP 1.0). The inter-page and inter-object time are exponentially distributed and the object size is Pareto distributed with shape parameter 1.2. The means of the distributions are identical to those chosen in [17] and [16]. We run simulations for TCP with web, 4CP with web and web only and for each case, we perform 30 simulations with different random seeds and each simulation has 200 sessions in 2000 second. We measure the size and response time of each web object and the long term average send rate achieved by TCP and 4CP. The response times of all the objects whose sizes are in some given range are averaged to derive the average response time for that size range. The web response times and long-run flow send rates are shown in Figure 21. The first and second rows show the difference and the ratio of the average web response between 4CP case (or web only case) and TCP case for given object size ranges. If the difference is smaller than 0, then the ratio is smaller than 1, and the web users get benefit when the competing long-run traffic chooses 4CP (or when there is no competing long-run traffic). We see that for 4CP Automatic, the 4CP user achieves similar send rate to the TCP user, and the large size web objects benefit from lower response times for 4CP. The small size objects response time have negligible difference, and its download time is small anyway. For 4CP fixed tarp option, we can choose tarp to balance the benefit to web users and the send rate achieved by the 4CP user. When tarp is chosen to be small, the web user behaves like low priority traffic, as the web response time is almost the same as the web only case, and meanwhile, the 4CP user achieves a decent send rate. Even when tarp is chosen to be very large and 4CP user takes a very large send rate, the large size web objects also benefit.

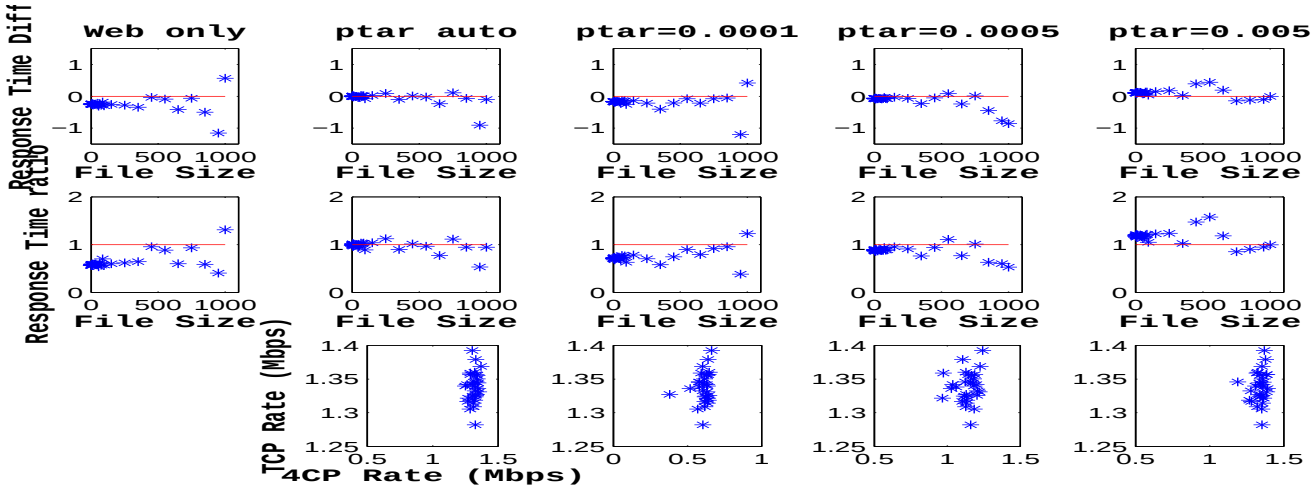


Figure 21: The response time for web-like traffic and the send rate for long-run traffic in a scenario from [17, 16].

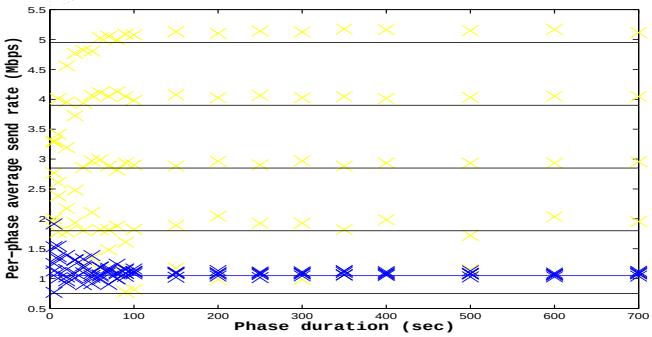


Figure 14: Send rate equilibrium points over phases versus a phase duration for the "pyramid" example of arrival and departure of short-run connections. Equilibrium points are achieved over a wide timescale of phase fluctuations, from sub minute to ten minutes.

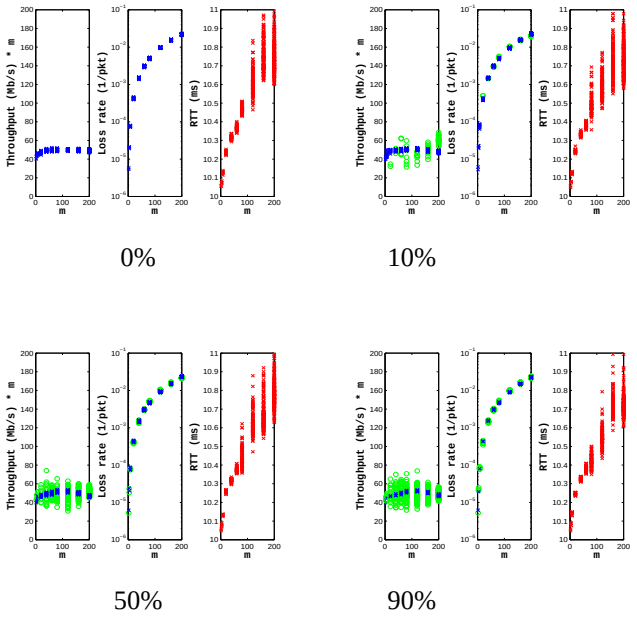


Figure 15: Validation of Claim 2; a bundle of n long-run connections competes for the bottleneck with n ranging from 2 to 200, with percentage $x\%$ 4CP Automatic as indicated in the figure and rest are TCP.

Rate based implementation. Although our standard implementation of 4CP is in transport layer, we can also implement it in the application layer. To support this claim, we implemented a rate based 4CP, and performed simulations to demonstrate its performance. We choose the same pyramid short-run scenario as in Figure 7 and we perform simulations for both fixed $\mathrm{tarp} = 0.0017$ (corresponding to target TCP rate of 1 Mb/s) option and the automatic option. The results are shown in Figure 22 and Figure 23. We see that the rate based implementation achieves similar results as the window based one and this gives us flexibility in implementing 4CP.

5.2 Internet Measurements

Congestion Control Module Implementation. We implemented 4CP in the kernel of a next-generation operating system. The implementation uses a congestion control module that provides an interface to redefine congestion control state of the underlying TCP at specific events such as timeouts, duplicate acknowledgments, etc.

Setup of Experiments. We run a limited set of experiments from a site in Europe to a site on the US West coast.

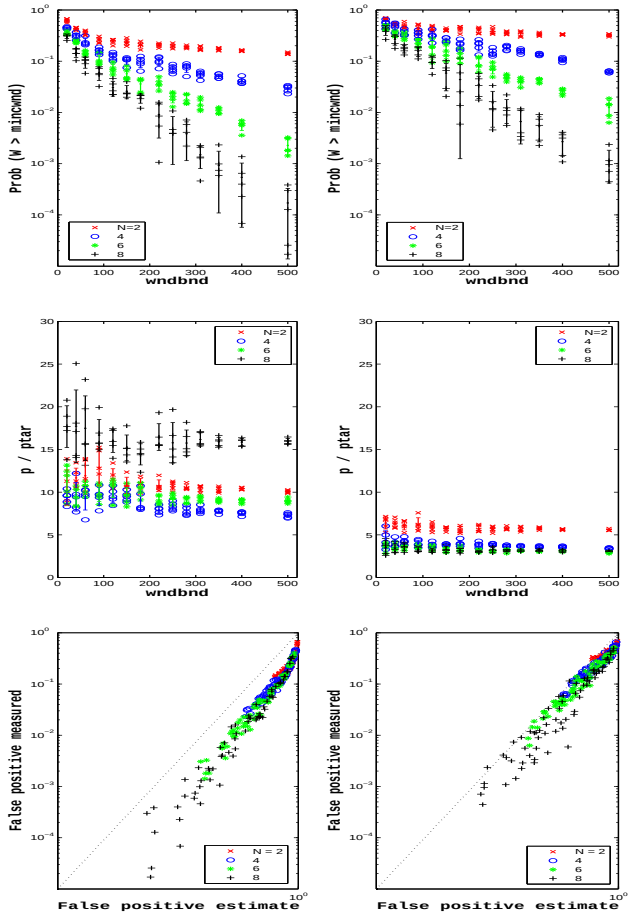


Figure 16: False positives of the bad phase detector: (Left) fraction of time false positives are reported versus the boundary wndbnd; (Middle) the loss rate to target loss ratio; (Right) the estimate of probability of false positives (Theorem 1) versus the estimates obtained from simulations by evaluating the result of Theorem 1 with the estimated loss rate. The fraction of time the false positives are reported decreases exponentially with the boundary wndbnd.

This network path is of capacity about 20 Mb/s with round-trip delay of about 170 ms.

Measurement results. We present two sets of experimental results.

First, the goal is to demonstrate non-intrusiveness of 4CP to short-run transfers with sizes covering that of web like traffic. To that end, we run a bundle of 12 TCP and a bundle of 12 4CP connections in two distinct experiments over a period of over 1 hour. In each experiment, we also initiate short-run transfers with file sizes exponentially increasing over the range of 100 to 125,000 bytes and this is repeated for 10 rounds. Each short-run transfer follows an idle time of 15 seconds. We then measure the response times of short-run transfers and compare the samples observed in all TCP and all 4CP case. Figure 24 shows the difference of the short-run transfer response times and suggests that 4CP has

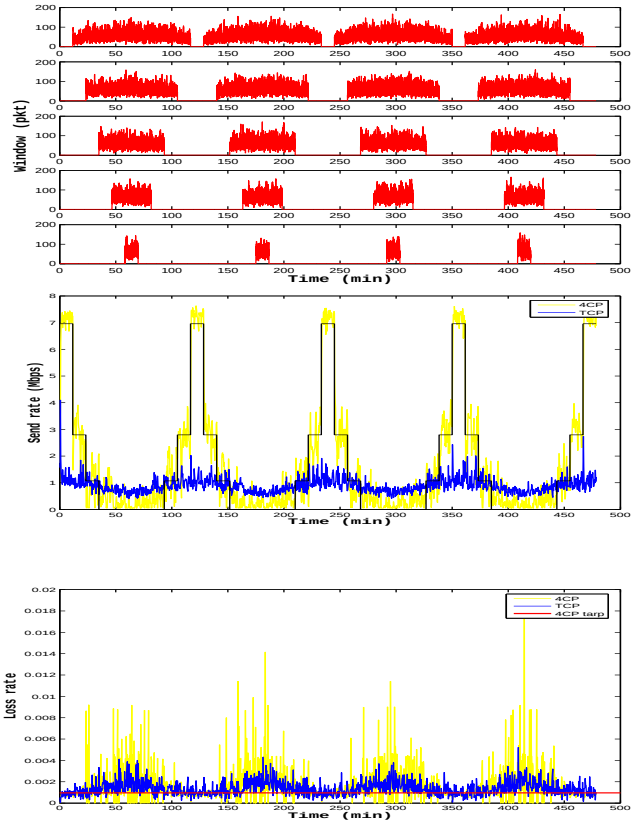


Figure 17: Multi-hop scenario: (Top) Congestion window of single-hop connections; (Middle) send rates; (Bottom) loss rates. The latter two are for multi-hop, long-run 4CP and TCP connections.

no adverse effect on short-run transfers compared with TCP. The sample paths of windows in Figure 24 demonstrate that both TCP and 4CP are in equilibria.

Second, we fix the target loss rate tarp to a value smaller than a priori observed loss rate over the same network path in a mix of 1 4CP and 20 TCP long-run connections. Note that phase is now bad. Figure 25 shows the window of a 4CP and the congestion window of a TCP long-run connection. The 4CP window is almost always negative as it should be with only sporadic excursions to positive window.

6 Discussion and Related Work

We discuss briefly the microeconomics optimality of 4CP (Automatic) controller; its connection to TCP-friendly sources, and go over some related work on service differentiation.

Microeconomics Optimality: 4CP Automatic inherits optimality properties as established in [15]. The controller can be casted to the microeconomics framework ([14]) by associating to it a utility function that is related to a given loss-throughput function f as:

$$U(x) = \int f^{-1}(x)dx$$

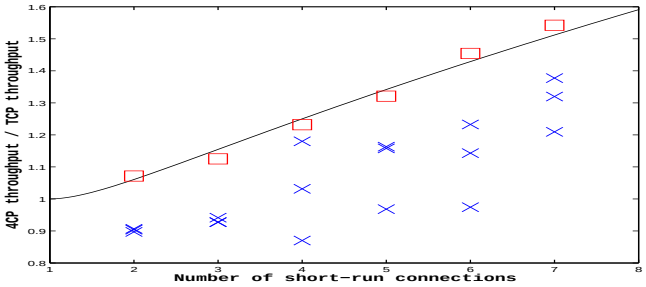


Figure 18: Validation of Claim 3 by simulations: scenario is one bottleneck, two long-run connections (one 4CP and one TCP), and short-run TCP file transfers. The plot shows 4CP to TCP throughput ratio of the long-run connections. Results validate achievability of the bound in Theorem 2 and demonstrate that it is a bound. Throughput gain is indeed significant, ranging from 10 to 50 %.

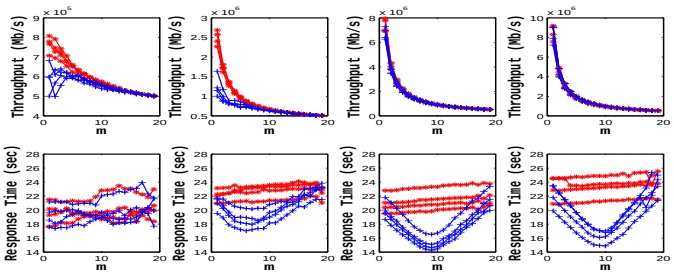


Figure 20: Response time simulation to validate Claim 4 and Proposition 1. $m + n = 20$ is fixed and m varies from 1 to 19. The $n \vee m$ curve for the response times as $a \to 0$ is validated

where f^{-1} is simply the inverse of f. The function f is in microeconomics parlance rephrased as a demand function and a loss rate p rephrased to a price per unit flow. The framework in [15] assumes a dichotomy of users types: (i) standard "myopic" users evaluate their utility functions at the instantaneous send rate; (ii) in contrast, "farsighted" users evaluate their utility functions at the long-run average send rate. The problem is to maximize aggregate sum of user utilities subject to network capacity constraints that accommodates for the user dichotomy and fluctuations of network congestion state. The results in [15] tell that the optimum strategy for the users of type ii is so called farsighted strategy. 4CP Automatic implements the farsighted strategy. The farsighted strategy appears not uncommon in economics literature [23, 22], but to the best of our knowledge appears novel in the area of network congestion control.

Our proposal is somewhat related to the "smart-market" scheme by MacKie-Mason and Varian [18]. In their scheme, each packet carries a price that the user is willing to pay—a bid. At each instant, only packets with bids higher than a cutoff price are serviced. The relation with our proposal is imminent, interpreting the target loss rate tarp as the cutoff price.

TCP-friendliness: The slowly-responsive congestion controllers gained quite some attention for media streaming applications; e.g. Floyd et al [9], Bansal et al [3]. The 4CP Automatic can be seen as a conservative source [25] that obeys a prescribed loss throughput relation, with the objective to maximize the long-run throughput.

Explicit Rate Allocations Clark and Fang [6] propose a framework to explicitly allocate rates to Internet users in periods of congestion. Specifically, the framework consists (i) policing the per-user traffic at network edge by tagging the packets that violate user profiles; (ii) preferential dropping of the tagged packets by network routers. Specific tagging algorithms are proposed to target specified sending rate for bulk-data transfers. The framework provides service with predictable expectation.

Strict Low Priority: There have been several proposal for end-point emulation of strict low priority service, which we outline now. TCP Nice [24] is a delay-based (TCP Vegas style) congestion controller at sender side. TCP-LP [17] is a sender-side controller based on one-way packet delays, implemented by modifying a TCP sender. BATS [16] is an alternative emulation performed at layer-7 by controlling the receiver window of a standard TCP receiver; it thus can be contrasted from TCP Nice and TCP-LP that are sender-side controllers.

Size-based Differentiations: Yang and de Veciana [26], Deb, Ganesh and Key [7] propose modified versions of TCP congestion control with the aim to provide service differentiation with respect file transfer sizes. Both propose redefining increments and decrements of TCP congestion window to some functions of residual file size so that as the file transfer progresses, the controller becomes more aggressive. The size-based differentiation is motivated by known optimality of Shortest Remaining Processing Time scheduling in minimizing mean response time (e.g. [4]). Preferential treatment of short-run connections at network routers is studied by Guo and Matta [12].

Application-based Differentiations: Gibbens and Kelly [11] consider a rate adaptation strategy for file-transfer applications. Given is target number of loss events or marks to undergo and file size. The send rate is adapted over a long timescale to balance loss rate per unit time to a target value. This target value, in turn, is set at short

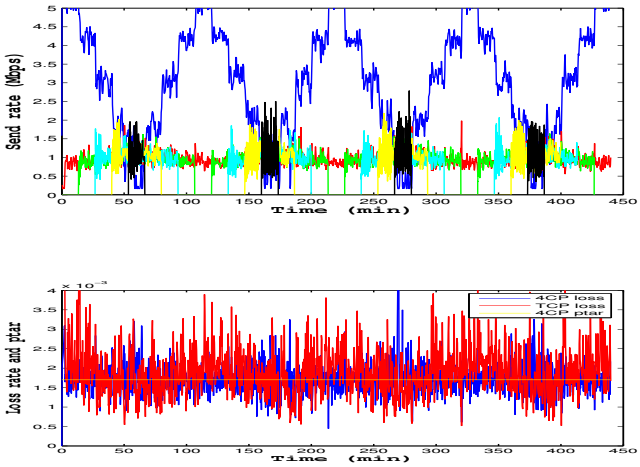


Figure 22: Rate based implementation for 4CP, fixed $\mathrm{tarp} = 0.0017 \rightarrow x_{\mathrm{TCP}} = 1$ Mb/s. (Top plot) send rates. (Bottom) loss rates.

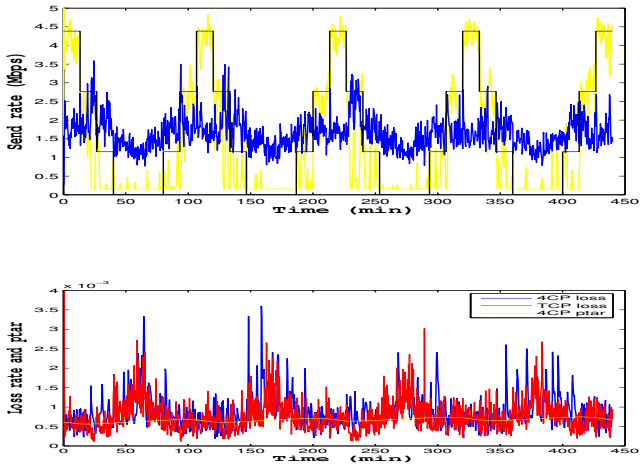


Figure 23: Rate based implementation for 4CP-Automatic. (Top) send rates. (Bottom) loss rates.

timescale with objective to target loss rate equal to the ratio of residual number of loss events to undergo and residual file size. The strategy is different than ours, however, similar is that the algorithm runs on two timescales and the observed bistable behavior in that the source either sends or waits; the latter if the average loss rate per unit time is larger than the target.

7 Conclusion

The growing use of intensive file transfer applications, e.g. peer-to-peer file sharing, with the transfer times spanning tens of minutes, hours, or even days and their potential adverse effects on other traffic motivated us to rethink the way such long-run file transfers are controlled. On one end, the standard way is to use TCP as many applications do, e.g. peer-to-peer. This positions such file transfer application at the same priority level as "normal traffic", e.g. interactive web browsing or media streaming. The problem is that typical file transfers involve several concurrent connections at a time; while this still gives normal traffic a TCP fair share, it may be rather small. The other end is to assign the file transfer applications a lower priority as is implemented by emulators of strict low priority. The latter, though, may lack incentive for application designers to use owing to its tendency for starvation in presence of any traffic on a link.

Motivated by these observations, we propose 4CP congestion control that emulates a different service differentiation than commonly assumed by low-priority service emulators. 4CP offers two modes. First, it offers a tuning knob to adjust per-flow average bandwidth guarantee, for any competing normal priority connection. This is done at the sender-side only in an entirely decentralized fashion and thus alleviates the need for a centralized traffic management controller. Second, 4CP (Automatic) can self-tune the tuning knob to adjust the average bandwidth guarantee for normal traffic so that it verifies TCP loss-throughput relation over a timescale that covers fluctuations in the network state. The 4CP Automatic mode inherits optimality properties of the farsighted congestion controllers, as it does implement the farsighted strategy. We believe our proposal is novel with features such as combining the congestion window control with detection of the network state; its design rationale draws from optimality properties of farsighted control and optimal detection.

We use a combination of analytical results, extensive simulations, and some Internet experiments to demonstrate benefits and co-existence of 4CP with TCP, both long-run and short-run connections (e.g. web like). Our analysis reveals that 4CP Automatic can achieve significant throughput gains over TCP when used in presence of network congestion fluctuations. We also demonstrate examples that show feasibility of significant reductions of response times for competing short-run transfers. All our claims are validated by simulations. We expect the control to work over bottlenecks that provide the same loss rate per packet over competing connections. This would be achieved, for example, with bottleneck using schemes such as RED, but may fail with bottlenecks such as DropTail with highly synchronized losses. The uniformity of loss rates over connections is a standard requirement of many congestion control protocols and is not 4CP intrinsic. Ongoing testing on the real Internet may explore these issues further.

From a systems perspective, it is an advantage that the control can be realized without making any receiver changes. By supporting interconnection with standard TCP receivers, we do not break conformance with the end-to-end model that is of benefit in real world Internet concerns such as firewall / NAT traversal and encryption (e.g. IPSEC). Furthermore, we call out the ability to operate end to end with no changes to

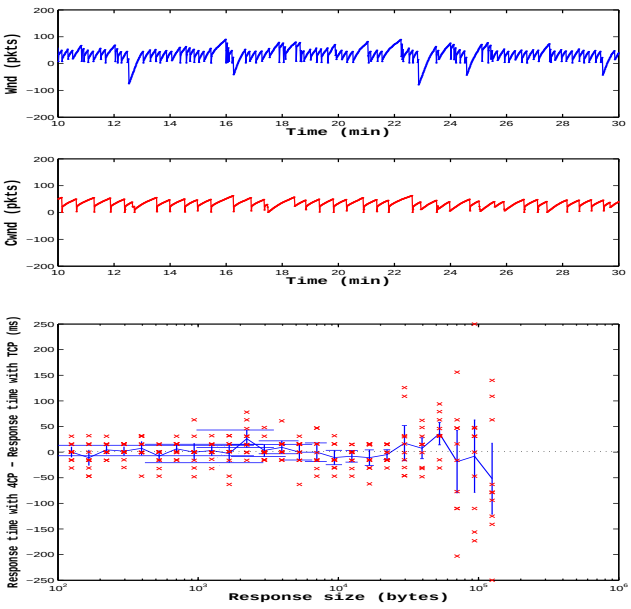


Figure 24: The impact of long-run connections on short lasting transfers over a network path from a site in Europe to US West coast. One set of experiments is for all TCP and other for all 4CP long-run connections with $\mathrm{tarp} = 0.0003$. The short-run transfer sizes range from 100 to 125K bytes inter-spaced with 15 sec idle intervals. (Top) The difference of response times for short-run transfers with 4CP and TCP versus the file size of short-run transfers. (Bottom) Sample paths of the windows of long-run connections.

existing Internet interconnection hardware as a benefit. The ability to implement the control at the end system and to posed its benefits in simple terms, provide many incentives for home users to adopt the control. These arguably include non-starvation of long run flows with low impact on short run flows (e.g. web like) latencies, with control parameter automatic tuning (no magic parameters) and with intra and inter protocol coexistence. These tangible benefits may help drive deployment of the 4CP controller.

Acknowledgments

We are grateful to Laurent Massoulié for discussions at various stages of this work. We appreciate a discussion with Frank Kelly. Hal R. Varian sent us his comments and pointed us to pertinent economics literature. Bing Wang provided us with explanations of simulations in [16]. Richard Black helped us with Microsoft Vista installation. Murari Sridharan assisted us with the CCM code. We thank them all.

References

[1] Qbone scavanger service (qbss), 2005. http://qbone.internet2.edu/qbss.

[2] S. Asmussen. *Applied Probability and Queues*. Springer, 2 edition, 2003.

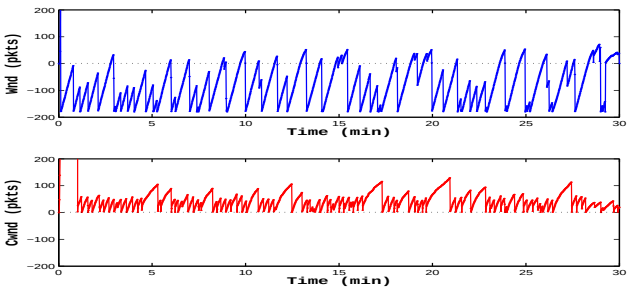


Figure 25: Same setup as in Figure 24 but for a mix of 1 4CP and 20 TCP connections with tarp chosen so that phase is bad, with no short-run transfers initiated. (Top) 4CP window; (Bottom) TCP congestion window.

[3] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker. Dynamic behavior of slowly-responsive congestion control algorithms. In *Proc. of ACM Sigcomm'01*, San Diego, California, USA, August 2001.

[4] N. Bansal and M. Harchol-Balter. Analysis of SRPT Scheduling: Investigating Unfairness. In *Proceedings of ACM Sigmetrics 2001*, 2001.

[5] R. Bless, B. Carpenter, K. Nichols, and K. Wehrle. A lower effort per-domain behavior for differentiated services. *Internet Draft, draft-bcnw-diffserv-pdb-le-00*, June 2002.

[6] D. D. Clark and W. Fang. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Trans. on Networking*, 6(4):362–373, 1998.

[7] S. Deb, A. Ganesh, and P. Key. Resource allocation between persistent and transient flows. *IEEE/ACM Trans. on Networking*, 13(2), 2005.

[8] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Trans. on Networking*, 7(4):458–472, August 1999.

[9] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In *Proc. of the Sigcomm'00*, pages 43–56, 2000.

[10] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397–413, August 1993.

[11] R. J. Gibbens and F. P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35:1969–1985, 1999.

[12] L. Guo and I. Matta. The War Between Mice and Elephants. In *Proceedings of IEEE ICNP'01*, 2001.

[13] T. Kailath and H. V. Poor. Detection of stochastic processes. *IEEE Trans. on Information Theory*, 44(6):2230–2259, October 1998.

[14] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998. (also available from http://www.statslab.cam.uc.uk/ frank).

[15] P. Key, L. Massoulié, and M. Vojnović. Farsighted Users Harness Network Time-Diversity. In *Proceedings of IEEE Infocom 2005*, Miami, FL, 2005.

[16] P. Key, L. Massoulié, and B. Wang. Emulating low-priority transport at the application layer: a background transfer service. In *Proceedings of ACM SIGMETRICS 2004*, pages 118–129, 2004.

[17] A. Kuzmanović and E. Knightly. TCP-LP: a distributed algorithm for low priority data transfer. In *Proceedings of IEEE Infocom 2003*, San Francisco, CA, USA, March 2003.

[18] J. K. MacKie-Mason and H. R. Varian. *Pricing the Internet, in B. Kahin and J. Keller, editors, Public Access to the Internet*. Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[19] J. Mahdavi and S. Floyd. TCP-Friendly Unicast Rate-Based Flow Control. Technical note sent to end-2-end interest mailing list, http://www.psc.edu/-networking/papers/tcp_friendly.html, January 1997.

[20] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Reno Performance: A Simple Model and its Empirical Validation. *IEEE/ACM Trans. on Networking*, 8(2):133–145, 2000.

[21] W. L. Smith. Regenerative stochastic processes. *Proc. Roy. Soc. A*, 232(6):6–31, 1955.

[22] H. R. Varian. A model of sales. *The American Economic Review*, 70:651–659, 1980.

[23] H. R. Varian. private communication, 2006.

[24] A. Venkataramani, R. Kokku, and M. Dahlin. TCP Nice: a mechanism for background transfer. In *Proceedings of OSDI'02*, pages 329–343, 2002.

[25] M. Vojnović and J.-Y. L. Boudec. On the long-run behavior of equation-based rate control. *IEEE/ACM Transactions on Networking*, 13(3):568–581, June 2005.

[26] S. Yang and G. de Veciana. Size-based Adaptive Bandwidth Allocation: Optimizing the Average QoS for Elastic Flows. In *Proceedings of IEEE Infocom 2002*, pages 657–666, New York, NY, 2002.

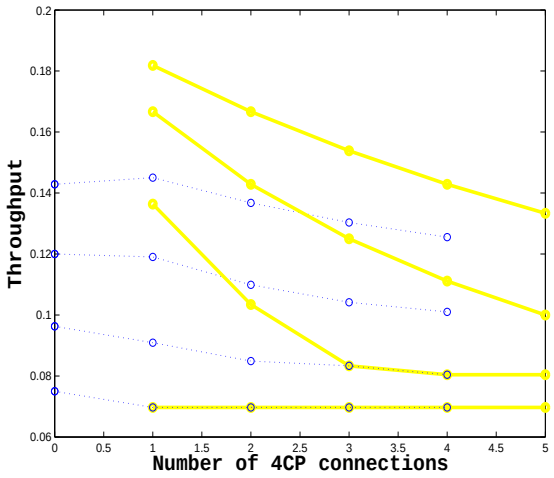


Figure 26: Throughput benefit of switching strategies for the baseline case. There are in total 5 long-run connections; either TCP or 4CP Automatic. The dotted and solid lines are respectively TCP and 4CP Automatic equilibrium throughputs versus the number of TCP connections. In many cases it is throughput beneficial for a connection to switch from TCP to 4CP Automatic, but not always.

A Benefits of Switching Strategies

In this section, we present several properties that characterise the competition of 4CP Automatic and TCP over a bottleneck for our baseline case. These results add to understanding the properties of farsighted congestion controls (recall, 4CP Automatic is a farsighted controller) to those established in [15].

Properties. The following properties characterize the competition of 4CP Automatic and TCP connections for the baseline arrival of short-run transfers:

1. Phase n is a good phase, for all $k = 1, 2, \ldots, m$, if and only if $a > 1 - 1/n$.

2. Diminishing returns. Suppose n, m, and a are fixed and define $k^* = \lfloor \frac{n}{1+an} \rfloor$. Both $\bar{x}_{4\mathrm{CP}}(k)$ and $\bar{x}_{TCP}(k)$ decrease with k, for $k = 1, \ldots, k^*$, and remain fixed to $\bar{x}_{4\mathrm{CP}}(m)$, for $k = k^* + 1, \ldots, m$.

3. Initial strategy switch. For a system with all m long-run TCP connections, it is throughput-beneficial for a long-run connection to switch to 4CP Automatic, i.e. $\bar{x}_{TCP}(0) < \bar{x}_{4\mathrm{CP}}(1)$, if and only if $a < 1 - 1/m$.

4. Subsequent strategy switch. Suppose $k^* > 2$. There exists a $k' < k^* - 1$, such that whenever $k \leq k'$, it is throughput-beneficial for a long-run TCP connection to switch to 4CP Automatic, i.e. $\bar{x}_{TCP}(k) < \bar{x}_{4\mathrm{CP}}(k + 1)$, for $k = 1, \ldots, k'$.

The new information items are [15]: (i) switching from TCP to 4CP Automatic can have diminishing throughput returns for TCP; (ii) switching from TCP to 4CP can be throughput beneficial for a connection, but not always.

B Relation to CUSUM Optimal Detection

Our bad phase detector has the same dynamics as the optimum change point detector known as CUSUM (e.g. [13]). In this section, we elucidate this connection. We believe it is worth noting that the choice of our detector is not arbitrary, but closely related to an optimum detector.

The underlying problem is known as standard Poisson disorder. The detector observes instants in time of some events that are assumed to be according to a homogeneous Poisson process with intensity 1, starting from $t = 0$, which then switches to a homogeneous Poisson process with intensity λ_1, at some unknown time $\theta > 0$. The standard problem assumes that λ_1 is known, which is not the case in our setting, but we know that $\lambda_1 < \text{mincwnd} \cdot \text{tarp}$. The goal of the detector is to detect the change point based on the observed instants of events. The likelihood ratio that $N(0, s]$ points are observed on an interval $(0, s]$ with points according to inhomogeneous Poisson with intensity $\lambda(s)$, $0 \leq s \leq t$, versus a Poisson of intensity 1, is equal to:

$$\text{lik}(t) = \exp\left(\int_0^t \log(\lambda(s))N(ds) - \int_0^t (\lambda(s) - 1)ds\right).$$

The CUSUM statistic X is defined as [13, Section V.B]:

$$X(t) = \sup_{\theta \leq t} \frac{\text{lik}(t)}{\text{lik}(\theta)}.$$

For standard Poisson disorder problem, $\lambda(t) = \lambda_0 + (\lambda_1 - \lambda_0)1_{t \geq \theta}$. Now $X(t) = \exp(\sup_{\theta \leq t} U(t, \theta))$ with $U(t, \theta) := \log(\text{lik}(t)) - \log(\text{lik}(\theta))$. For $\lambda_1 < 1$, we consider the transformed process $W(t) = \sup_{\theta \leq t} U(t, \theta)/(1 - \lambda_1)$, which can be re-written as

$$W(t) = \sup_{\theta \leq t} \{(t - \theta) - c' N(\theta, t]\}. \tag{10}$$

where $c' := \log\left(\frac{1}{\lambda_1}\right)/(1 - \lambda_1)$. Equation (10) is precisely the same dynamics as we have with our detector wnd for $\text{wnd} < 0$. Now, in our case λ_1 is unknown and thus we take the maximum in (10) over $\lambda_1 < \text{mincwnd} \cdot p$, which amounts to $c' = -\log(\text{mincwnd} \cdot \text{tarp})/(1 - \text{mincwnd} \cdot \text{tarp})$. The 4CP detector can be seen as a conservative version in view of $c' \leq 1/(\text{mincwnd} \cdot \text{tarp})$.

C Proof of Theorem 1

Consider the dynamics of wnd specified by (7). Recall that $c = 1/(\text{mincwnd} \cdot \text{tarp})$ and let $\lambda = \text{mincwnd} \cdot p$. Denote with T_n the time of the n-th loss event observed by the detector on $t \geq 0$, $n = 1, 2, \ldots$ and let $T_0 := 0$. Denote with $S_n = T_{n+1} - T_n$, the time between the nth and $n + 1$th loss event and denote with W_n the value of the detector just after the nth loss event occurence. We have, for $n = 0, 1, \ldots$,

$$W_{n+1} = \max(W_n + S_n - c, \underline{w}). \tag{11}$$

where $\underline{w} := -\text{wndbnd}$.

The limit distribution of W_n as n goes to infinity is known in closed-form and specified by:

Lemma 1. *Assume $1/\lambda < c$. We have, for $w \geq \underline{w}$:*

$$\lim_{n \to +\infty} \mathbb{P}(W_n > w) = \left(1 - \frac{\gamma}{\lambda}\right) e^{-\gamma(w - \underline{w})} \tag{12}$$

where a is the solution of

$$1 - \frac{\gamma}{\lambda} = e^{-\gamma c}. \tag{13}$$

Note that $\gamma/\lambda = \lim_{n \to +\infty} \mathbb{P}(W_n = \underline{w})$.

Proof of Lemma Denote with V_n, the shifted process $V_n = W_n - \underline{w}$, so that V_n takes values on $\mathbb{R}^+$. It follows from (11), that V_n obeys a special case of well-known Lindley's recursion; given V_0,

$$V_{n+1} = \max(V_n + X_n, 0),\ n = 0, 1, \ldots$$

with X_n a sequence of independent and identically distributed random variables with $X_0 \sim F$ and $F(x) = \mathbb{P}(X_n \leq x) = 1 - e^{-\lambda(x+c)}$, $x \geq c$ [X_n is a sum of c and a random variable $\sim Exp(\lambda)$].

From Corollary 6.5 [2], we have that for $\lambda < 1/c$,

$$\lim_{n \to +\infty} \mathbb{P}(V_n \leq x) = G(x)$$

where $G(\cdot)$ is a solution of Lindley's integral equation:

$$G(x) = \int_{-\infty}^{x} G(x - y)F(dy),\ x \geq 0. \tag{14}$$

We claim that

$$G(x) = 1 - \left(1 - \frac{\gamma}{\lambda}\right) e^{-\gamma x},\ x \geq 0, \tag{15}$$

with γ given by

$$\int_{-\infty}^{+\infty} e^{\gamma y} F(dy) = 1.$$

This is verified directly. Let $\bar{F}(x) := 1 - F(x)$ and $\bar{G}(x) := 1 - G(x)$. From (14), it follows

$$\bar{G}(x) = \bar{F}(x) + \int_{-\infty}^{x} \bar{G}(x - y)F(dy).$$

Plugging the asserted solution (15), it easily follows

$$1 = \int_{-\infty}^{+\infty} e^{\gamma y} F(dy) + \frac{\lambda}{\lambda - \gamma} \bar{F}(x) e^{\gamma x} - \int_{x}^{+\infty} e^{\gamma y} F(dy).$$

It suffices to show that

$$\frac{\lambda}{\lambda-\gamma}\bar{F}(x)e^{\gamma x}=\int_x^{+\infty}e^{\gamma y}F(dy).$$

But this is direct by plugging the function F.

We thus have obtained the limit distribution of V_n as n goes to infinity. It only remains to relate this to the limit distribution of W_n. Indeed, $W_n > w$ is equivalent to $V_n > w - \underline{w}$, $w \geq \underline{w}$, and thus

$$\lim_{n\to+\infty}\mathbb{P}(W_n > w) = G(w-\underline{w}),\ w \geq \underline{w}.$$

The result follows. **q.e.d.**

Having specified the limit distribution of W sampled just after occurrences of loss events, we proceed with obtaining a closed form expression of the time-limit distribution of W sampled at an arbitrary time. This can be interpreted as the long-run fraction of time, $W(t)$ is larger or equal a given value.

Lemma 2. *Under $1/\lambda < c$, for $w \geq \underline{w}$*

$$\lim_{t\to+\infty}\mathbb{P}(W(t) > w) = e^{-\gamma(w-\underline{w})}$$

where γ is the unique solution of (13).

Proof of Lemma The stochastic process $W(t)$ is regenerative with respect to the set $\{\underline{w}\}$. Under $1/\lambda < c$, the following limit holds from [21][Theorem 2]:

$$\lim_{t\to+\infty}\mathbb{P}(W(t) > w) = \frac{1}{\mathbb{E}(T)}\mathbb{E}\left(\int_0^T 1_{\tilde{W}+s>w}ds\right),$$

where $\tilde{W}$ is a random variable with distribution (12) and $T \sim Exp(\lambda)$. We have, for $w \geq \underline{w}$,

$$\begin{aligned}
&\lambda\mathbb{E}\left(\int_0^T 1_{\tilde{W}+s>w}ds\right) \qquad (16)\\
=\ &\lambda\mathbb{E}(T1_{\tilde{W}>w}) + \lambda\mathbb{E}((T_1-(w-\tilde{W}))^+1_{\{\tilde{W}\leq w\}})\\
=\ &\mathbb{P}(\tilde{W}>w) + e^{-\lambda w}\int_{(\underline{w},w]}e^{\lambda y}d\mathbb{P}(\tilde{W}\leq y)\\
&+\frac{\gamma}{\lambda}e^{-\lambda(w-\underline{w})}.
\end{aligned}$$

In the last equality, we used the fact $\mathbb{E}((X-a)^+) = e^{-\lambda a}/\lambda$, for $X \sim \text{Exp}(\lambda)$ and $a \geq 0$. Now, the integral in the above display is computed directly by plugging the density of (12):

$$\begin{aligned}
&e^{-\lambda w}\int_{(\underline{w},w]}e^{\lambda y}d\mathbb{P}^0(\tilde{W}\leq y)\\
=\ &\gamma e^{-\lambda w}\int_{(\underline{w},w}\left(1-\frac{\gamma}{\lambda}\right)e^{\lambda y}e^{-\gamma(y-\underline{w})}dy\\
=\ &\gamma\left(1-\frac{\gamma}{\lambda}\right)e^{-\lambda w+\gamma\underline{w}}\int_{(\underline{w},w]}e^{(\lambda-\gamma)y}dy\\
=\ &\frac{\gamma}{\lambda}\left(e^{-\gamma(w-\underline{w})}-e^{-\lambda(w-\underline{w})}\right).
\end{aligned}$$

The result follows by substituting the last expression and (12) in (16). **q.e.d.**

Proof of Theorem: Follows directly from Lemma 2 by substitution $a = \gamma/\lambda$ and instantiating the result for $w = \text{mincwnd}$. Condition $1/\lambda < c$, reads $p > \text{tarp}$, i.e. $r > 1$ ("phase is bad"). **q.e.d.**

D Proof of Theorem 2

It suffices to consider a link of unit capacity. Congestion state takes values on a finite set of phases that are enumerated as $u = 0, 1, \ldots, n$. Phase u is intepreted as the number of short-run transfers. In a good phase, u', the equilibrium rate of TCP is such that

$$x_{\text{TCP}}(u') = \bar{x}_{\text{4CP}}$$

which follows from equalization of loss rate over good phases [15] to tarp and $\bar{x}_{\text{4CP}} = f(\text{tarp})/r$, where f is TCP loss to average window function and r is the round-trip time.

It follows, $x_{\text{4CP}}(u) = \max\{1-(u+1)\bar{x}_{\text{4CP}}, 0\}$ and thus

$$\bar{x}_{\text{4CP}}(\pi) = \frac{\sum_{j=0}^{u^*}\pi(j)}{1+\sum_{j=0}^{u^*}(1+j)\pi(j)},$$

where u^* is a positive integer given by

$$\begin{aligned}
(u^*+1)\sum_{j=0}^{u^*}\pi(j) &< 1+\sum_{j=0}^{u^*}(1+j)\pi(j) \qquad (17)\\
(u^*+2)\sum_{j=0}^{u^*}\pi(j) &\geq 1+\sum_{j=0}^{u^*}(1+j)\pi(j). \qquad (18)
\end{aligned}$$

Furthermore,

$$\bar{x}_{\text{TCP}}(\pi) = \bar{x}_{\text{4CP}}(\pi)\sum_{u=0}^{u^*}\pi(j) + \sum_{j=u^*+1}^{n}\frac{1}{1+j}\pi(j).$$

The minimization of $\bar{x}_{\text{TCP}}(\pi)/\bar{x}_{\text{4CP}}(\pi)$ over π can be phrased as:

$$\min\quad x+\frac{1}{x}\left(1+\sum_{j=0}^{u^*}(1+j)\pi(j)\right)\sum_{j=u^*+1}^{n}\frac{1}{1+j}\pi(j) \qquad (19)$$

over $\pi(j) \geq 0$ subject to $\sum_{j=0}^{n}\pi(j) = 1$, with $x := \sum_{j=0}^{u^*}\pi(j)$. We perform the minimisation by first minimising for fixed x and then minimising over $x \in (0,1)$. We can separate the minimisations in (19) over $(\pi(1), \ldots, \pi(u^*))$ and $(\pi(u^*+1), \ldots, \pi(n))$, and thus first consider:

$$\min\quad \sum_{j=u^*+1}^{n}\frac{1}{1+j}\pi(j)$$

over $\pi(j) \geq 0$, $j = u^*+1, \ldots, n$, subject to $\sum_{j=u^*+1}^{n}\pi(j) = 1-x$. The minimum is indeed $(1-x)/(1+$

n) achieved by putting all the mass $1-x$ to the phase n, i.e. $\pi^*(j)=0$, $j=u^*+1,\ldots,n-1$ and $\pi^*(n)=1-x$. It follows that the problem (19) can be rewritten as:

$$\min \quad x+\frac{1}{1+n}\frac{1-x}{x}\left(1+\sum_{j=0}^{u^*}(1+j)\pi(j)\right)$$

over $\pi(j)\geq 0$, $j=0,\ldots,u^*$, subject to $\sum_{j=0}^{u^*}\pi(j)=x$.

The minimum is indeed achieved by putting all the mass x to the phase 0, i.e. $\pi^*(0)=x$ and $\pi^*(j)=0$, $j=1,\ldots,u^*$. We have thus showed that (19) can be rephrased as:

$$\min_{x\in(0,1)} \quad \left(1-\frac{1}{1+n}\right)x+\frac{1}{1+n}\frac{1}{x}.$$

The minimum is achieved for $x^*=1/\sqrt{n}$. Constraints (17)–(18) are verified for $u^*=\lfloor\sqrt{n}\rfloor-1$, where $\lfloor x\rfloor$ is the largest integer smaller or equal to x. Proof is completed.

E Proof of Theorem 3

For our setting, the distribution of phases π is given by:

$$\pi(0)=\frac{1}{1+a/\tilde{x}}, \quad \pi(n)=1-\pi(0) \tag{20}$$

where $a:=f/(c\tau)$ and $\tilde{x}$ is fraction of the bottleneck capacity allocated to a short-run transfer in phase n.

First, consider $k=0$, i.e. all m long-run connections are TCP. The equilibrium rates are:

$$(x_{\rm TCP}(0),x_{\rm TCP}(n)) \quad = \quad \left(\frac{c}{m},\frac{c}{m+n}\right).$$

The distribution of phases π is given by (20) with $\tilde{x}=1/(m+n)$. It follows:

$$\frac{\bar{x}_{\rm TCP}}{c}=\frac{1}{m}\left(1-\frac{an}{1+a(n+m)}\right).$$

Second, consider $k\geq 1$. We distinguish two cases: (i) phase n is bad and (ii) phase n is good.

Case i (phase n is bad). Assume the condition holds. The equilibrium points are:

$$\begin{aligned}(x_{\rm TCP}(0),x_{\rm TCP}(n)) &= \left(\bar{x}_{\rm 4CP},\frac{c}{m-k+n}\right)\\ (x_{\rm 4CP}(0),x_{\rm 4CP}(n)) &= ([c-(m-k)\bar{x}_{\rm 4CP}]/k,0)\,.\end{aligned}$$

We have

$$k\bar{x}_{\rm 4CP}(k)=\pi(0)[c-(m-k)\bar{x}_{\rm 4CP}(k)]. \tag{21}$$

The phase distribution π is given by (20) with $\tilde{x}=1/n$. It follows

$$\pi(0)=\frac{1}{1+an}.$$

and

$$\frac{\bar{x}_{\rm 4CP}(k)}{c}=\frac{1}{m}\frac{1}{1+an(k/m)} \tag{22}$$

Condition "phase n is bad" means $1-(m-k+n)\bar{x}_{\rm 4CP}(k)/c\leq 0$. Plugging (22) we rephrase the condition as

$$a\leq\frac{1}{k}-\frac{1}{n}. \tag{23}$$

Note that phase n cannot be bad for $k\geq n$.

Throughput of a long-run TCP connection is given by:

$$\bar{x}_{\rm TCP}(k)=\pi(0)\bar{x}_{\rm 4CP}(k)+(1-\pi(0))\frac{c}{m-k+n}.$$

Plugging (22), we obtain:

$$\frac{\bar{x}_{\rm TCP}(k)}{c}=\frac{1}{1+an}\left(\frac{1}{m+ank}+a\frac{n}{m-k+n}\right).$$

Case ii (phase n is good). Condition is $a>1/k-1/n$ (Equation 23). The equilibrium points are:

$$\begin{aligned}(x_{\rm TCP}(0),x_{\rm TCP}(n)) &= (\bar{x}_{\rm 4CP},\bar{x}_{\rm 4CP})\\ (x_{\rm 4CP}(0),x_{\rm 4CP}(n)) &= ([c-(m-k)\bar{x}_{\rm 4CP}]/k,\\ &\quad\ [c-(m-k+n)\bar{x}_{\rm 4CP}]/k).\end{aligned}$$

Indeed, $\bar{x}_{\rm TCP}(k)=\bar{x}_{\rm 4CP}(k)$, $k=1,2,\ldots,m$. We have:

$$\begin{aligned}k\frac{\bar{x}_{\rm 4CP}}{c} &= \pi(0)\left(1-(m-k)\frac{\bar{x}_{\rm 4CP}}{c}\right)+\\ &\quad +(1-\pi(0))\left(1-(m-k+n)\frac{\bar{x}_{\rm 4CP}}{c}\right)\end{aligned}$$

with $\pi(0)$ given by (20) with $\tilde{x}=\bar{x}_{\rm 4CP}/c$. It follows:

$$m\left(\frac{\bar{x}_{\rm 4CP}}{c}\right)^2+(a(m+n)-1)\frac{\bar{x}_{\rm 4CP}}{c}-a=0.$$

The positive solution is unique and given by: $\bar{x}_{\rm 4CP}(k)/c=$

$$=\frac{1}{2m}\left(1-a(m+n)+\sqrt{(1-a(m+n))^2+4am}\right)$$

Note that the right-hand does not depend k.

F Proof of Proposition 1

Rewrite (24) as: $r_{\rm 4CP}(n,m,k,a)=$

$$=\begin{cases} n1_{a\leq 1/k-1/n}+\frac{1}{\bar{x}_{\rm 4CP}}1_{a>1/k-1/n} & k<n\\ \frac{1}{\bar{x}_{\rm 4CP}} & k\geq n.\end{cases} \tag{24}$$

Now, $n\leq 1/\bar{x}_{\rm 4CP}$ and the right-hand side is decreasing with a. Thus, for $k<n$, $\inf_{a>0} r_{\rm 4CP}(n,m,k,a)=n$, else for $k\geq n$, $\inf_{a>0} r_{\rm 4CP}(n,m,k,a)=m$. Thus the asserted identity under item 1. The tightness follows as both the last infima are achieved for $a=0$.

We now show item 2. From item 1, we have

$$\frac{r_{\rm 4CP}(n,m,m,a)}{r_{\rm TCP}(n,m)}\geq\frac{n\vee m}{n+m}, \quad a>0.$$

Denoting $x=m/(n+m)$, we have $(n\vee m)/(n+m)=x\vee(1-x)$. Indeed, $\inf_{x\in[0,1]} x\vee(1-x)=1/2$. The result follows.