Delay based congestion control for heterogeneous environments

Łukasz Budzisz, Rade Stanojević, Arieh Schlote, Robert Shorten and Fred Baker

***Abstract*—This paper presents and develops a novel delay-based AIMD congestion control algorithm. The main features of the proposed solution include: (1) low standing queues and delay in homogeneous environments (with delay-based flows only); (2) fair coexistence of delay- and loss-based flows in heterogeneous environments; (3) delay-based flows behave as loss-based flows when loss-based flows are present in the network; otherwise they revert to delay-based operation. It is also shown that these properties can be achieved without any appreciable increase in network loss rate over that which would be present in a comparable network of standard TCP flows (loss-based AIMD). To demonstrate the potential of the presented algorithm both analytical and simulation results are provided in a range of different network scenarios. These include stability and convergence results in general multiple-bottleneck networks, and a number of simulation scenarios to demonstrate the utility of the proposed scheme. In particular, we show that networks employing our algorithm have the features of networks in which RED AQM's are deployed. Furthermore, in a wide range of situations (including high speed scenarios), we show that low delay is achieved irrespective of the queueing algorithm employed in the network, with only sender side modification to the basic AIMD algorithm.**

***Index Terms*—delay-based congestion control, TCP.**

I. Introduction

CONGESTION control in TCP/IP networks is traditionally handled using packet losses to indicate congestion [1]. An alternative approach to respond to congestion involves the use of network delay. This was first proposed by Jain [2] and since then, there has been much work and debate on this topic [3–11]. Delay-based congestion control is conceptually very attractive. Potential benefits include the ability to allocate the network bandwidth between competing sources with: (i) low (zero) packet loss; (ii) very low queueing delay; and (iii) with full utilisation of network links. Networks which exhibit this property are said to operate at the *knee of the throughput-delay* curve [12]. Motivated by these and other potential benefits, delay-based congestion control remains an active area of research and new algorithms continue to be developed. Recent examples include: Fast TCP [3, 13]; Microsoft Compound [14] (partially based on delay); more recent delay-based additive increase multiplicative decrease (AIMD) variants [15–18]; and this present work.

Ł. Budzisz, and R. Shorten, are with Hamilton Institute, NUI Maynooth, Ireland (e-mail: [Lukasz.Budzisz, Robert.Shorten]@nuim.ie). R. Stanojević is with Telefónica I+D, Barcelona, Spain (e-mail: Rade.Stanojevic@nuim.ie). A. Schlote is with Institut für Mathematik, Universität Würzburg, Germany (e-mail: arieh.schlote@stud-mail.uni-wuerzburg.de). F. Baker is with Cisco Systems (e-mail:fred@cisco.com).

Despite this large body of work, several issues concerning the use of queueing delay remain to be resolved before delay based congestion control can be deployed. These include: (i) the difficulty in obtaining delay estimates from network measurements [19]; (ii) network sampling issues [19–22]; (iii) the inability of existing delay-based algorithms to maintain a low standing queue [21, 22]; and (iv) the inability of delay-based flows to coexist fairly with loss-based flows in mixed environments. These items have been the subject of much discussion by us and other researchers [19–21] which we do not repeat here. Rather, we focus the specific issue of coexistence. Note that the issue here is not just simple coexistence; after all, delay-based flows may simply switch to a loss-based mode once a packet loss is detected, thereby solving the fairness problem. The issue that makes coexistence difficult is that delay-based flows must revert back to delay-based operation when loss-based flows are no longer present. One of the recent works in the area of coexistence [23, 24] develops a mathematical framework that helps prove the existence of the network equilibrium in heterogeneous environments. In this work we go one step further, and propose to design the nature of the network equilibria to achieve our goals.

Given these basic observations, our contribution in this paper is to develop and explore a strategy that resolves (at least in part) these issues. In particular, our principal contribution in this work is to propose a strategy that allows delay-based TCP flows to coexist fairly and without any discernible increase in network loss rate with loss-based flows when they are present, and to revert back to a delay-based behaviour whenever loss-based flows are no longer present. While the basic idea underlying this work was presented in [25, 26], this paper extends and complements this work in a number of ways[1]. These include more comprehensive simulation studies involving realistic network scenarios (multiple bottlenecks, reverse traffic, and the presence of on/off network flows), and a detailed mathematical analysis. Finally, we demonstrate the potential benefits of the proposed strategy when applied to high speed links.

Our paper is structured as follows. In Section II we introduce the proposed algorithm, illustrating its efficacy in Section III with a number of simulations in both single and multiple bottlenecks. Then, in Section IV we provide mathematical analysis that describes the ability of the pro-

[1]This current paper is a journal version of the articles [25, 26].

posed solution to switch between loss-based and delay-based operation regimes and reflects on stability issues. Next, in Section V, we compare the proposed solution to other existing strategies, namely RED and one of the most recent delay-based congestion control schemes, as well as provide some insights on its application in more versatile scenarios, including high speed networks. Finally, in Section VI we discuss constraints limiting the application of the proposed algorithm.

II. AQM EMULATION TO ENSURE FAIR COEXISTENCE WITH LOSS-BASED FLOWS

Our basic idea is motivated by recent work concerning Active Queue Management (AQM) emulation from end-hosts using delay measurements, called Probabilistic Early Response TCP (PERT) [17]. The basic idea behind PERT is very simple and involves responding to delay in a probabilistic rather than deterministic manner. By judiciously selecting the manner of the probabilistic response, Reddy et al. [17] are able to emulate, from end-hosts, the behaviour of a range of AQM's. To facilitate such AQM emulation, each PERT flow probes the network for congestion as a normal AIMD flow, but reduces its congestion window in a probabilistic manner that depends on the estimated network delay. We refer to this mechanism as a *back-off policy*. The authors of PERT demonstrate that (in principle) any AQM can be emulated by selecting the back-off policy appropriately. However, it was subsequently shown that their algorithm fails to solve the coexistence problem in a satisfactory manner [26]. In particular, it is shown in this latter paper that PERT can lead to high loss rates when loss-based flows are present, and that the delay-based flows may fail to revert to delay-based operation when the loss based flows leave the network. Our main contribution here is to present a similar idea that can be used to solve the coexistence problem by carefully choosing the probabilistic back-off function, while avoiding many of the side-effects of the PERT strategy. As we shall see our strategy, referred to as Coexistent TCP (C-TCP) further in this article, avoids problems of adjusting AIMD parameters, keeps the network loss rate low when loss-based flows are present, and ensures that the delay-based flows revert to delay-based operation when loss-based flows are no longer present in the network (termed here as *on/off behaviour*), even though these flows do not attempt to sense the presence of such flows directly.

Specifically, our basic idea is to achieve coexistence by carefully selecting the back-off policy to achieve fairness and on/off behaviour. In what follows, we assume that queueing delay (δ), minimum (RTT_{min}), and maximum round-trip time (RTT_{max}) can be estimated reliably by all delay-based flows in the network and do not consider the issue of slow start for delay-based flows; these, and other issues that are not within the scope of this paper, are discussed in our previous work, and in previous work by other authors [15, 19–21].

We select probabilistic back-off strategies of the form depicted in Fig. 1. As can be observed, the per-packet back-off probability function $p = g(\delta)$ has two parts; a part that increases monotonically with the delay δ (Region A), and a part that decreases monotonically with δ (Region B). This form of AQM emulation has the following properties:

(i) assuming that the maximum equilibrium loss rate (p_{max}) is large enough, the network stabilises in Region A when only delay-based flows are present;

(ii) when loss-based flows are present, the network is driven to Region B, and delay-based flows behave as loss-based flows due to the low per-packet back-off probability;

(iii) when loss-based flows switch off, the network cannot stabilise in this region due to a backward pressure exerted by the probability function. Namely, as the flows experience back-offs, the queueing delay reduces, thereby increasing the per-packet back-off probability, thus making further back-offs more likely. This process continues until the network stabilises in Region A.

As can be seen, this type of strategy should achieve co-existence of loss- and delay-based AIMD flows, without a discernible increase in network loss rate. Furthermore, the back-pressure described in (iii) should ensure on/off behaviour. The basic C-TCP algorithm is summarised in Algorithm 1. δ is estimated as the difference between the weighted average of the RTT and its minimum observed value. To ensure similar number of RTT samples for flows with different RTTs, the RTT weight is set proportionally to the ratio of the average RTT and the cwnd value for a given flow. Again, the reader should be reminded that this paper does not discuss the accuracy of such an estimate.

Finally we emphasize, our objective here is to present a simple idea that may be very useful in solving the coexistence

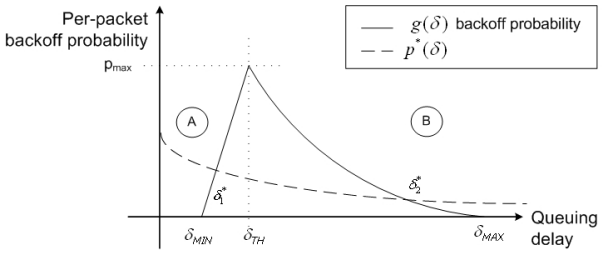


Fig. 1. Per-packet back-off probability as a function of the observed delay.

Algorithm 1 Pseudo-code for C-TCP algorithm

On receipt of each ACK:
Estimate the current queueing delay: δ
Set $p = g(\delta)$: (only non-zero values are shown)

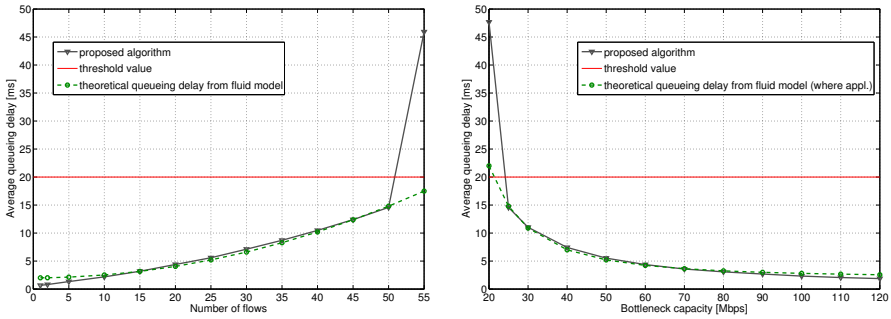
$$g(\delta) = \begin{cases} p_{max}\frac{\delta-\delta_{min}}{\delta_{th}-\delta_{min}} & \text{for} \quad \delta_{min} < \delta \leq \delta_{th} \\ p_{max}\left(\frac{\delta_{max}-\delta}{\delta_{max}-\delta_{th}}\right)^4 & \text{for} \quad \delta_{th} < \delta < \delta_{max} \end{cases}$$

Pick a random number $rand$, uniformly from 0 to 1
if $rand < p$ **then**
  reduce $cwnd$ by $0.5 \cdot cwnd$
else
  increment $cwnd$ by $1/cwnd$
end if

TABLE I
SUMMARY OF THE SCENARIO PARAMETERS.

PARAMETER	LOW DELAYS	REVERSE TRAFFIC	FEATURES IN HOMOGENEOUS ENVIRONMENTS	FAIRNESS	SCALABILITY	LOSS RATE	ON-OFF SWITCHING
Capacity	(1): 25 Mbps (2): 20-120 Mbps	25 Mbps	50 Mbps	25 Mbps	500 Mbps	10-100 Mbps	25 Mbps
RTT	all flows: 100 ms	all flows: 100 ms	30-130 ms, uniformly	30-130 ms, uniformly	30-130 ms, uniformly	30-130 ms, uniformly	30-130 ms, uniformly
Number of flows	(1): 1-50(55) (2): 50	forward path: 1-50 reverse path: 20	(1-2): 30 flows (3): 51 flows (20 in at 200 s, 30 out at 400 s)	30 flows in 2 mixes: (20,10) and (10,20)	600 flows: (400,200) mix	30 flows: (20,10) mix	30 flows: 20 delay-based and 10 inter-mittent‡ (100-300 s) loss based flows
Web traffic	no	no	no	yes: 10 flows	yes: 100 flows	yes: 10 flows	no
Common parameters	For all tests: buffer size on each link: 1BDP; packet size: 1000 Byte; simulation time: 500 s; All loss based flows (web traffic sessions as well) use Reno TCP.						

‡ 10 intermittent loss based flows enter the scenario in 0.5 s intervals (100-105 s) and leave in 5 s intervals (250-300 s).



(a) (b)

Fig. 2. Proposed algorithm in a single bottleneck scenario - average queueing delay as a function of: (a) the number of flows for a bottleneck capacity of 25 Mbps; and (b) capacity for 50 flows.

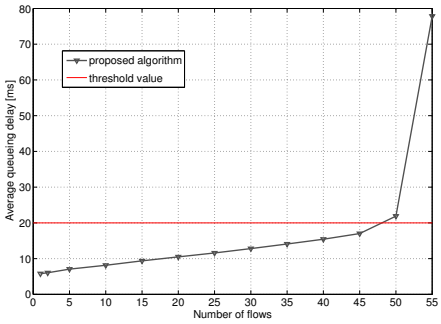


Fig. 3. Performance with 20 flows on the reverse path: average queueing delay as the number of flows on the forward path.

problem. In its current embodiment, the idea works best in heavily multiplexed environments; and this is reflected in the experimental results given in the paper.

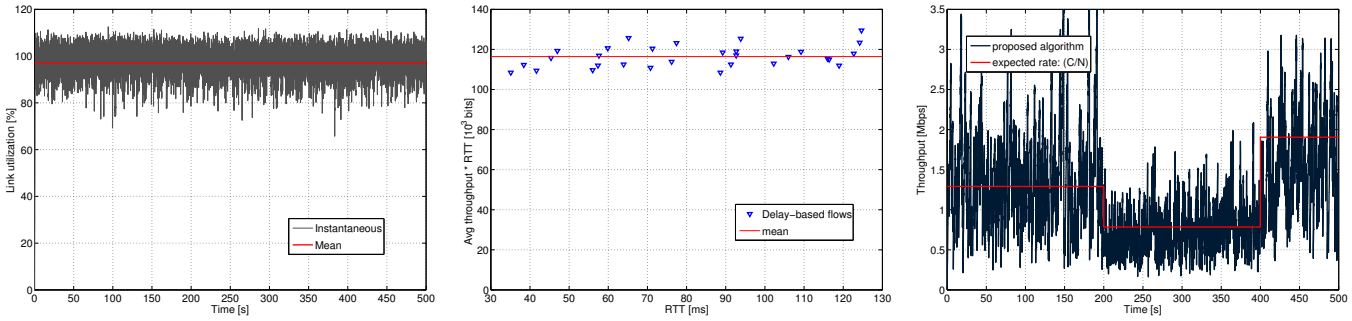
III. BASIC OPERATION OF ALGORITHM

To illustrate the basic operation of C-TCP described in Section II, we now present a number of simple experiments. We begin with a single bottleneck topology and then progress to a more realistic multiple bottleneck scenario. Our objectives are: (i) to demonstrate that low delay networks can be realised in homogeneous environments; (ii) to illustrate that almost perfect coexistence can be achieved; (iii) to examine the fairness properties of the algorithm in mixed environments; (iv) to show that the loss rate is low (not worse than that of just loss based flows) when loss and delay based flows coexist together; and (v) to demonstrate that on/off switching can be achieved. All experiments are constructed with these objectives in mind.

A. Single bottleneck

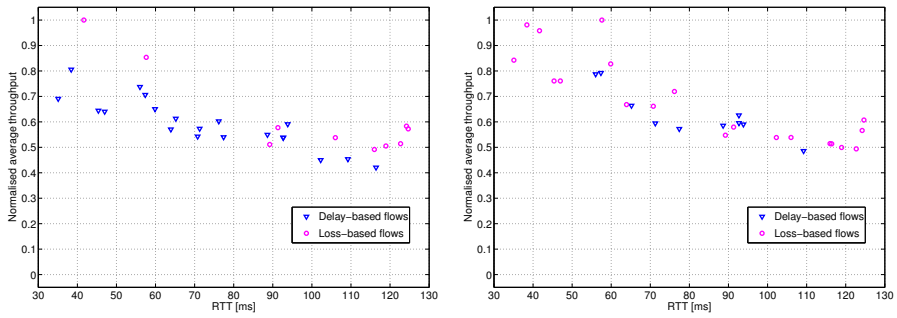
We begin with the evaluation of C-TCP in a single-bottleneck scenario in a classic *dumb-bell* topology. In the following tests, we use ns-2 [27] simulations with the most important settings specified in Table I. As for the C-TCP parameters, the algorithm proposed here emulates RED AQM in the region A (Fig. 1). Consequently, one method to select the parameter values for δ_{min} (5 ms), δ_{th} (20 ms) and p_{max} (5 %) is to use the rules for RED parameter settings [17]. Meanwhile δ_{max} is estimated for each flow separately, with an initial value of 100 ms. Note that parameter settings provided here are suitable for a range of link capacities, and bandwidth. See basic feature and scalability tests for more comments on that matter.

1) **Basic feature - low delays for homogeneous networks:** We first demonstrate that our algorithm does indeed yield low delay networks in case all flows are delay based. As already mentioned in Section II, the maximum equilibrium loss rate is given by p_{max} (see Fig. 1). This means that the network will revert to a loss based network if there is a very large number of network flows (related to the available network capacity); namely, if the required equilibrium loss rate is greater than p_{max}. This property is desirable as it is well known that estimation of queueing delay is difficult in networks with very large multiplexing of flows [21]. As an example, Fig. 2a presents the average δ as a function of the number of delay-based flows sharing a 25 Mbps bottleneck. It can be clearly seen that the maximum number of flows for that given capacity (N_{max}) lies between 50 and 55 flows; readers however should keep in mind that the bottleneck capacity influences this upper limit for any selected parameter settings. N_{max} can be also approximated theoretically using the fluid model, as it will be



(a) (b) (c)

Fig. 4. Illustration of the properties of the proposed algorithm in homogeneous scenarios: (a) link utilization; (b) window fairness; and (c) convergence properties.



(a) (b)

Fig. 5. Coexistence of the delay-based flows and standard TCP flows in terms of the normalised average throughput for the following mixes of flows: (a) the (20,10) mix; and (b) the (10,20) mix.

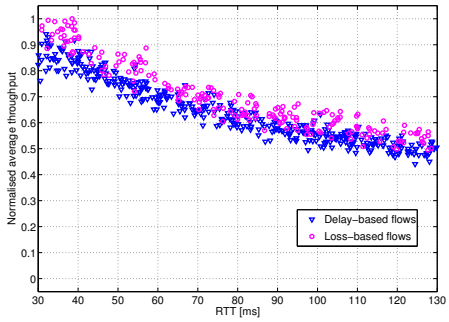


Fig. 6. Scalability of the proposed solution (preserving the C/N ratio): coexistence of the (400, 200) mix of flows in a scenario with a 500Mbps bottleneck bandwidth.

explained later in Section IV.

Next, in Fig. 2b we show the average δ as a function of the bottleneck capacity for 50 delay-based flows competing for the same bottleneck. Again, we will validate the results with the fluid model in Section IV. This figure illustrates also that for a given number of flows the proposed parameter settings permit the application of C-TCP in the entire capacity range tested. For more comments on the scalability issue, see the scalability test.

2) **Reverse path traffic:** It is well-known that reverse-path traffic can increase the ACK losses for the forward path flows. To check the influence of this issue on the performance of our algorithm, we introduce 20 long-lived flows using also C-TCP on the reverse path. Fig. 3 illustrates the results for the average δ as a function of the number of delay-based-forward-path flows sharing the same 25 Mbps bottleneck with 20 delay-based flows on the reverse path. It can be clearly seen, as if compared to Fig. 2a that the limit of the operation of the presented algorithm has been decreased to accommodate the reverse path traffic.

3) **Further features in homogeneous scenarios:** Apart from maintaining low queueing delay when operating in the delay-based mode, C-TCP has several interesting features that are illustrated in Fig. 4.

(i) The proposed algorithm depends on a non-zero average queuing delay. So by the very nature of this hypothesis, we are assuming that the link is close to being fully utilised at all times. This is indeed the case, see Fig. 4a depicting link utilisation for 30 flows competing on a 50 Mbps bottleneck.

(ii) The proposed dropping policy (shown in Fig. 1) emulates RED. As such, it strives to achieve *window fairness*, i.e., in the equilibrium state all flows have, on average, the same cwnd. This is illustrated experimentally in Fig. 4b. Here, 30 flows with RTTs uniformly distributed between 30 and 130 ms all achieve similar average window size.

(iii) Finally, we examine the convergence properties of C-TCP in a setting where N varies instantaneously for a given bottleneck capacity (50 Mbps). This is depicted in Fig. 4c. Initially, 31 delay-based flows run the proposed algorithm, and then, flows either enter (additional 20 flows at time: 200 s), or leave (30 flows at time: 400 s) the discussed scenario. As can be seen, the allocation of bandwidth switches instantaneously to the correct equilibria.[2]

4) **Fairness:** Next, we examine the ability of delay-based flows to coexist (fairly) with standard loss-based flows. Fig. 5 depicts the normalised average throughput (calculated as a ratio of the best average throughput over a 500 second-long experiment) for a network with two different mixes of standard TCP and C-TCP flows (a total of 30 flows). Note that while there is a bias in favour of the loss-based flows (due to the fact that the delay-based flows experience a small number of non-loss induced back-offs in the high-queue regime), there is a reasonably fair coexistence of the loss-based and delay-based

[2]Due to the limited editorial space not all result for this particular experiment can be included here. For more results, please refer on-line at: http:\\www.hamilton.ie\lukasz.

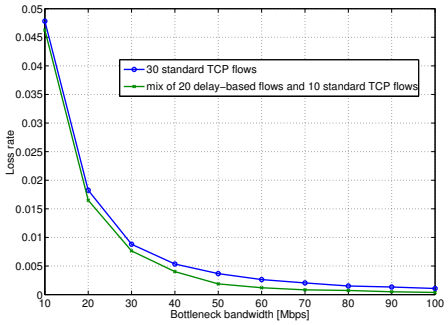
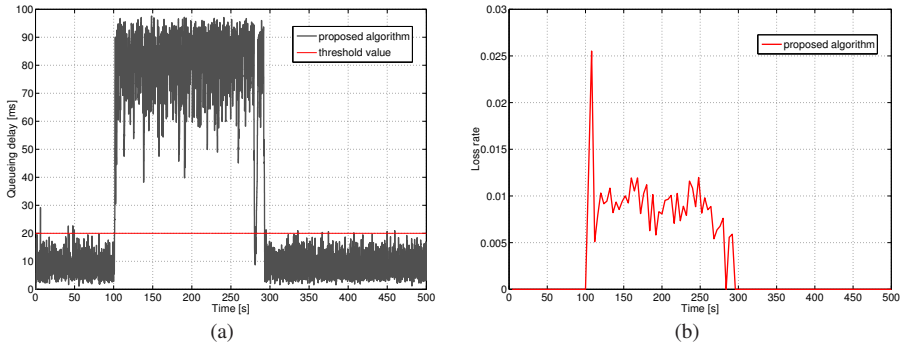
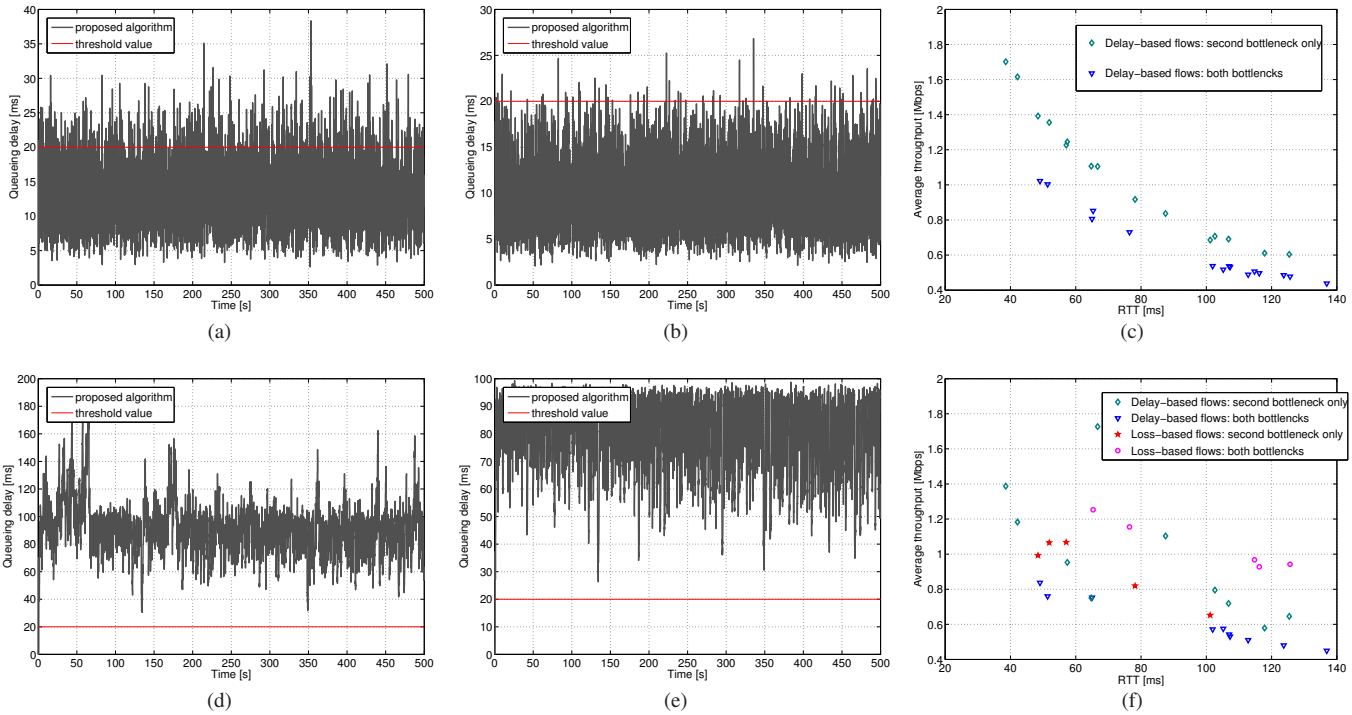


Fig. 7. Comparison in terms of loss rate: 20 delay-based flows coexisting with 10 standard TCP flows, and 30 standard TCP flows.



(a) (b)

Fig. 8. Coexistence of 20 delay-based flows with 10 loss-based flows switching on and off: (a) queueing delay at the bottleneck; and (b) loss rate.



(a) (b) (c) (d) (e) (f)

Fig. 9. Coexistence of 20 delay-based flows and 10 standard TCP flows (lower row) in comparison to 30 delay-based flows (upper row) in a multiple bottleneck scenario: (a) and (d): queueing delay at the first bottleneck; (b) and (e): queueing delay at the second bottleneck; and (c) and (f) average throughput.

AIMD flows. Furthermore, the aforementioned bias in favour of loss-based flows can be controlled by carefully selecting the back-off policy. Notwithstanding this latter observation, the experiments nevertheless demonstrate very good coexistence of the delay-based and loss-based flows as measured by the average throughput.

5) **Scalability:** As already mentioned, the parameter settings of the proposed algorithm influence the range of bandwidth in which C-TCP can operate. To check that, we run a test in a scenario with a 500 Mbps bottleneck. To meet the design assumptions we keep the ratio of C/N from the fairness experiments unchanged. This yields a scenario with a mix of (400, 200) delay and loss-based flows. As can be seen in Fig. 6 (on the previous page), C-TCP is able to keep the fairness property constant, despite of a slight bias in favour of the loss-based flows (see fairness comments).

6) **Loss rate:** Now, we examine the effect of our algorithm on the network loss rate (in the loss-based operation mode the back-off probability is low enough so that the congestion is controlled by packet losses). To do this we compare the behaviour of 20 delay-based C-TCP flows coexisting with 10 standard TCP flows with a scenario in which all 30 flows are standard loss-based TCP. The results are depicted in Fig. 7. Observe that the proposed algorithm does not significantly increase the network loss rate in the presence of loss-based flows. Fair coexistence is achieved without any unnecessary trade-offs.

7) **On/Off switching:** Our primary objective in this work was to develop a delay-based algorithm that behaves as a loss-based TCP when competing with loss-based TCP flows, but otherwise reverts to delay-based operation. This behaviour is captured in Fig. 8. Here, 30 flows (20 delay-based and 10 intermittent loss-based flows) compete for the available band-

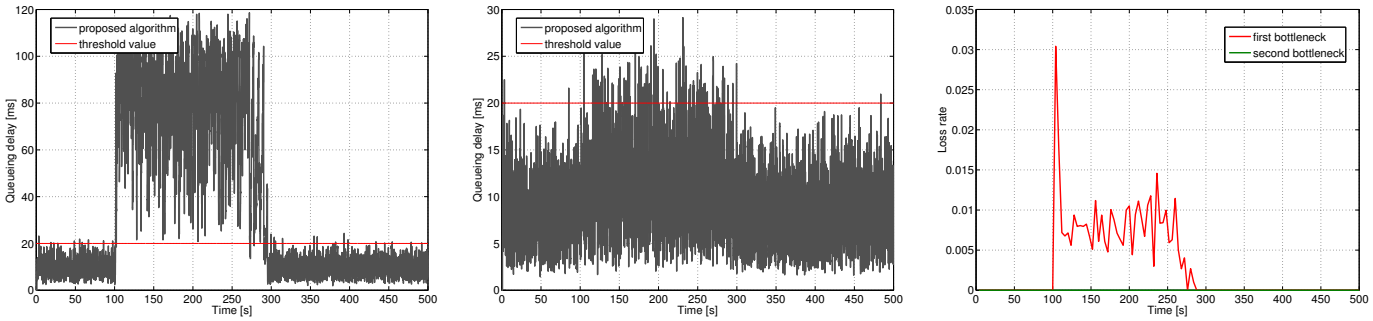


Fig. 10. Coexistence of 20 delay-based flows with 10 loss-based flows switching on and off: queueing delay at (a) the first bottleneck (where the loss-based interference appears), and (b) the second bottleneck; and (c) corresponding loss rates.

width. Between 100. and 300. second, when the loss-based flows appear in the network, the delay-based flows behave as standard TCP flows and compete fairly for bandwidth. Otherwise they strive in a cooperative manner to keep the queueing delay below a certain threshold (δ_{th}). Note also that the mode switching occurs automatically (and swiftly) without any complicated sensing or signal processing to determine whether or not the loss-based flows have left the network. Note also that for this algorithm, the operation mode is decided upon by the average value of δ, and not the instantaneous one. Thus, δ_{th} may be exceeded locally, but it is the average value of δ that decides which operation mode to choose. We believe that this mechanism is novel in delay-based congestion control context, and has uses beyond the present context.

B. Multiple bottleneck scenarios

Next, we examine C-TCP in a multiple-bottleneck situation. Objectives stay the same, as stated in the introduction to this section. The scenario under test is the simplest *parking-lot topology,* and comprises of a cascade of two bottlenecks, with half of all flows traversing both *first* and *second bottleneck* (and the remaining half living in the second bottleneck only). The first bottleneck has a capacity of 12.5 Mbps and a 155 packet queue, whereas the second bottleneck has 25 Mbps capacity, and a 310 packet queue, respectively. Each of the bottlenecks introduces 5 ms propagation delay, and serves a drop tail queue. Apart from the flows under evaluation, in the coexistence experiment 10 mice flows are added on each bottleneck to make the scenario more realistic. All the remaining parameters are the same as for the corresponding tests in Section III-A.

1) **Coexistence with loss-based flows:** First, we compare 30 delay-based flows using C-TCP to a mix of (20,10) delay- and loss-based flows split evenly between two bottlenecks. Fig. 9 (on the previous page) shows the results for each set in separate rows (the scenario with all flows being delay-based is shown in the upper row, whereas the mix of delay- and loss-based flows is shown in the lower row). In case there are only delay-based flows, the algorithm is able to keep queueing delay below the δ_{th} threshold at each of the bottlenecks (again, readers should be reminded that it is the average value of δ that decides the operating area). We shall see that such a homogeneous network has similar properties to a network in which a RED AQM is deployed. Note also that in presence of the standard TCP flows the algorithm switches to a loss-based operation, filling the queues at each bottleneck, but preserving the overall fairness with standard TCP counterparts (Fig. 9f).

2) **On/Off switching:** Next, we check the dynamic behaviour of C-TCP in the multiple bottleneck scenario with 30 flows (20 delay-based and 10 intermittent loss-based flows) competing for the available bandwidth (Fig. 10). When 10 loss-based flows appear at the first bottleneck (100-300 s), the 5 delay-based flows living there switch to the long-queueing-delay mode and compete fairly for bandwidth. Meanwhile, the 15 delay-based flows at the second bottleneck stay in the low queueing delay regime. Note that the high queueing delay (and non-zero loss rate) operation affects only the bottleneck where the loss-based flow appeared. Otherwise, when there are only delay based flows (before 100 s, and after 300 s), the low-queueing delay regime is maintained at both bottlenecks.

IV. Mathematical analysis

Having outlined the basic properties of our algorithm, we now characterise some of these properties in a mathematical framework. Our starting point is the single bottleneck scenario. We show, using a fluid (Kelly) like argument, that the on/off behaviour is a manifestation of the interaction of stable and unstable equilibria in the network. We then extend this result to multiple bottleneck networks.

Basic notation: By $\delta(t)$ and $W_i(t)$, where $i = 1, \ldots, N$, we denote the queueing delay and congestion window size ($cwnd$) of flow i at time t. Those quantities are related to link capacity C, as: $\sum_{i=1}^{N} \frac{W_i(t)}{RTT+\delta(t)} = C$.

Basic model (single bottleneck): In steady state (for RED like AQM's) the average window sizes do not depend on the round trip time; it is just a scaling for the speed of the evolution. Thus $W_1(t) = W_2(t) = \cdots = W_N(t) =: W(t)$ and together we get $W(t) = \frac{C(RTT+\delta(t))}{N}$. Using a standard fluid model the evolution of $cwnd$ is given by:

$$\frac{\Delta W(t)}{\Delta t} = \frac{1}{RTT + \delta(t)} - \frac{q_0}{\Delta t} \cdot \frac{W(t)}{2}, \qquad (1)$$

where q_0 is the probability that during the time interval $(t, t+\Delta t)$ a back-off occurred. We denote by M_0 the number of

packets a source with congestion window size of $W(t)$ sends in the interval $(t, t+\Delta t)$. Then $M_0 = \frac{\Delta t \cdot W(t)}{RTT+\delta(t)}$, and we can approximate q_0 as $q_0 = 1-(1-p)^{M_0} \approx pM_0 = \frac{p\Delta t \cdot W(t)}{RTT+\delta(t)}$. It therefore follows that (1) can be written as

$$\frac{\Delta W(t)}{\Delta t} = \frac{1}{RTT+\delta(t)}\left(1-\frac{p}{2}W^2(t)\right). \quad (2)$$

We are going to use this very general equation (2) to model our system also in the multiple bottleneck case. In the single bottleneck case we can further simplify (2) to

$$\frac{\Delta W(t)}{\Delta t} = \frac{1}{RTT+\delta(t)}\left(1-\frac{p}{2}\left(\frac{C(RTT+\delta(t))}{N}\right)^2\right) \quad (3)$$

As the right hand side of (3) no longer depends on Δt we can now write $\dot{W}(t)$ instead of $\frac{\Delta W(t)}{\Delta t}$. The network equilibria are given by $\frac{p}{2}\left(\frac{C(RTT+\delta(t))}{N}\right)^2 = 1$. We denote by $p^*(\delta)$, the equilibrium probability $p^*(\delta(t)) = \frac{2N^2}{C^2(RTT+\delta(t))^2}$. Recall that the per-packet back-off rate p is a function of the delay δ: $p = g(\delta)$. Therefore, the system (1) is in equilibrium at the points of intersection of curves $p^*(\cdot)$ and $g(\cdot)$. Those two curves have zero or one, or two points of intersection $\delta_1^* < \delta_2^*$ (see Fig. 1).

Single bottleneck result: Our objective is to design the network so that there are two equilibria (the regular regime). Using the Lyapunov function $V(\delta) = \delta^2$, it is easily deduced that the right point of intersection (δ_2^*) is an unstable equilibrium and that the other left equilibrium (δ_1^*) is a stable one. It also follows that if δ becomes smaller than δ_2^* the system will be driven to the stable equilibrium δ_1^*. We formalise these statements in the following lemma.

Lemma IV.1. *The system (1), has 0,1 or 2 equilibrium points:* (i) *If $p^*(\delta_{th}) < p_{max}$ there are two equilibrium points: $\delta_1^* < \delta_2^*$.The right equilibrium point δ_2^* is unstable and the left one, δ_1^* is stable.* (ii) *If $p^*(\delta_{th}) = p_{max}$ there is one, unstable, equilibrium and the cwnd dynamics is mainly driven by the packet drops, when the queue is full.* (iii) *If $p^*(\delta_{th}) > p_{max}$ there is no equilibrium, and the cwnd dynamic is mainly driven by the packet drops, when the queue is full.*

Proof: We prove item *(i)* here. Items *(ii)* and *(iii)* either follow directly or are proved analogously. Define a candidate Lyapunov function for the queue dynamics as $V(\delta(t)) = \delta(t)^2$. The assertions of the Lemma IV.1 follow from the fact that $\dot{V}(\delta(t)) = 2\delta(t)\dot{\delta}(t)$, and from the fact that $\dot{\delta}(t) > 0$ for all $\delta_{min} < \delta(t) < \delta_1^*$; $\dot{\delta}(t) < 0$ for all $\delta_1^* < \delta(t) < \delta_2^*$; and $\dot{\delta}(t) > 0$ for all $\delta(t) > \delta_2^*$. These facts follow directly from (1) and (3). Namely, for queueing delays $\delta < \delta_1^*$, the dynamics of the congestion window "overcomes" the per-packet back-off rate and forces the network toward δ_1^*. Between δ_1^* and the apex of the per-packet back-off rate, the back-off rate is sufficiently large to overcome (1) and forces the network to δ_1^*. Between the apex and δ_2^*, this latter mechanism is reinforced by back-off rate that increases as δ decreases. Thus, using standard Lyapunov theory, one concludes that δ_1^* is a stable equilibrium, whereas δ_2^* is not. ■

Model validation: The fluid like argument can be used to approximate theoretically the maximum number of flows that can be maintained in the low delay regime (N_{max}). We demonstrate that in the experiments for the single bottleneck scenario, shown in Section III-A in Fig. 2a and 2b. Of course, this is a coarse approximation to what is actually happening. The network is certainly not transporting fluid, but rather discrete packets. It is also discrete event in nature, and not continuous. Furthermore, each flow estimates δ and cannot see an exact value of this quantity for a variety of reasons; filtered measurements of δ (difference between weighted average of the RTT and its minimum observed value); different RTT's. Finally, our networks do not support an infinite number of flows, but rather, a small number of flows. Nevertheless, the fluid model yields some insights into the qualitative behaviour of our network and does approximate in some sense the network dynamics. With these qualifying remarks, we do use this model to calculate the N_{max}. As shown in Fig. 2a and 2b (on page 3), within the scope of application of the fluid model the results are acceptably close to the values measured from simulations.

Extended model (multiple bottlenecks): We now want to extend our model to handle networks of a more complicated topology. We emphasise that extending our results to this case is non-trivial; see [28] for examples of networks that are single-bottleneck-stable, but become unstable in the multiple bottleneck case due to queue interactions. In what follows we reuse the approach given in [29]. Although Wang et al. use a different model in [29], following their approach we can prove global asymptotic stability if our algorithm uses the following piecewise linear probability function:

$$p(\delta) = K \cdot \delta, \quad (4)$$

which is an approximation of the original probability function, shown in Fig. 1. We consider the network consisting of a set of resources $\mathcal{J} = \{1, \ldots, J\}$ (links) with capacities $\{C_1, \ldots, C_J\}$. We identify each route $\{1, \ldots, N\}$ with a subset $i \subset \mathcal{J}$ and assign it the window size $W_i(t)$, which is to be understood as the cumulative sum of the window sizes of the flows along route i and is to take values in the positive real numbers for all $t \in \mathbb{R}_+$. Each link is equipped with a queue. The queue sizes $q_1(t), \ldots, q_J(t)$ are also to take values in the positive real numbers for all $t \in \mathbb{R}_+$.

We can model the W_i in the following way

$$\dot{W}_i(t) = \frac{1}{RTT_i+\delta_i(t)}\left(1-\frac{p_i}{2}W_i^2(t)\right). \quad (5)$$

This is the same equation as in the single bottleneck case (2), where we use $\delta_i(t) = \sum_{j\in i}\frac{q_j(t)}{C_j}$ and

$$\dot{q}_j(t) = \begin{cases} \sum_{i:j\in i}\frac{W_i(t)}{RTT_i+\delta_i(t)} - C_j & \text{for} \quad q_j > 0 \\ \{\sum_{i:j\in i}\frac{W_i(t)}{RTT_i+\delta_i(t)} - C_j\}^+ & \text{for} \quad q_j = 0 \end{cases}, \quad (6)$$

for $i = 1, \ldots, N$, $j = 1, \ldots, J$. For any function f that maps into the real numbers, we denote by $\{f\}^+$ the non-negative part of f, that is $\{f\}^+ = \max\{f, 0\}$. The first equation is familiar from the single bottleneck case and the

second equation just states that queueing dynamics simply depend on the difference between total arrival rate and capacity at each link, with the constraint that the queue never takes negative values, but rather stays at zero until the dynamic is positive again. Setting (5) and (6) to zero, and assuming that at equilibria we have $\frac{1}{RTT_i+\delta_i(t)} = \frac{1}{RTT_i+\delta_i}$ for all $i = 1, \dots, N$, then

$$W_i^* = \sqrt{\frac{2}{p_i^*}} \tag{7}$$

$$\sum_{i:j\in i} \frac{W_i^*}{RTT_i + \delta_i} = C_j, \tag{8}$$

$i = 1, \dots, N$, $j = 1, \dots, J$.

To ease exposition we give our mathematical proofs for piecewise linear probability functions $p(\delta)$. All results generalise to the nonlinear case by replacing the piecewise linear function $p(\delta)$ with the appropriate linearisations (Jacobians). This extension is given formally (without proof) as Corollary IV.3. Full details and proofs can be found in [30]. To prove the desired results we partition our system by choosing two special probability functions, showing certain properties and fitting the two systems together again. At first we choose

$$p_i(\delta_i) = K\delta_i, \tag{9}$$

where δ_i is the sum over all queueing delays experienced by flow i and $K \in \mathbb{R}_+$ for all $i = 1, \dots, N$. Then we look at

$$p_i(\delta_i) = p_0 - \hat{K}\delta_i, \tag{10}$$

where again δ_i is the sum over all queueing delays experienced by flow i and $p_0 > 0$, $\hat{K} \in \mathbb{R}_+$ for all $i = 1, \dots, N$, where we make the additional assumption that $\delta_i \leq \frac{p_0}{\hat{K}_i}$, $i = 1, \dots, N$ such that $p_i(\delta_i)$ is always positive.

In the final step, we allow each flow to switch between the probability functions (9) and (10) in a deterministic manner so that we get an overall probability function

$$p_i(\delta_i) = \begin{cases} K\delta_i & \text{for} \quad 0 < \delta_i \leq \delta_{th} \\ p_0 - \hat{K}\delta_i & \text{for} \quad \delta_{th} < \delta_i \leq \delta_{max} \end{cases}. \tag{11}$$

Multiple bottleneck result: We can now state the main result of this section. This is a direct extension of the result for the single bottleneck case; albeit with an extremely involved proof.

We assume that the equilibria of our system lie in the appropriate area. Especially we assume for the stable equilibrium, that is the topic of the following theorem, that $\delta_i^* < \delta_{th}$ for all $i = 1, \dots, N$. To achieve this is rather a designing task as their location depends on the calibration of the algorithm's parameters.

Theorem IV.2. *The system described by* (5), (6), (11) *has only one asymptotically stable fixed point. Each other fixed point is unstable.*

Proof: The proof consists of three parts. First we will consider a system, where every flow uses (9) as its probability function. We will be able to show that there is a unique globally asymptotically stable fixed point to this system.

To ease the exposition of the final step, in the second step we will consider a system, where all flows use (10) as their probability function. We will be able to show, that all equilibria of this system are unstable. In the final step we consider (11), which will again prove unstable as long as some flows are in the high delay area.

(i) Let every flow use (9) as its probability function. We calculate the equilibrium $(W_1^*, \dots, W_N^*, q_1^*, \dots, q_J^*)$ using (7) and (8). Next we consider the following transformation of coordinates $\tilde{W}_i(t) = W_i(t) - W_i^*$ and $\tilde{q}_j(t) = q_j(t) - q_j^*$. We get the following system

$$\dot{\tilde{W}}_i(t) = -\frac{1}{2}\frac{K}{RTT_i+\delta_i}\left(\sum_{j\in i}\frac{\tilde{q}_j(t)}{C_j}\left(\tilde{W}_i(t)+W_i^*\right)^2 + \right.$$
$$\left. + \sum_{j\in i}\frac{q_j^*}{C_j}\left(\tilde{W}_i^2(t) + 2W_i^*\tilde{W}_i(t)\right)\right) \tag{12}$$

$$\dot{\tilde{q}}_j(t) = \sum_{i:j\in i}\frac{\tilde{W}_i(t)}{RTT_i+\delta_i}, \tag{13}$$

as can be seen from

$$\begin{aligned}
\dot{\tilde{W}}_i(t) &= \dot{W}_i(t) - \dot{W}_i{}^* \\
&= \frac{1}{RTT_i+\delta_i}\left(1 - \frac{p_i}{2}W_i^2(t)\right) \\
&= \frac{1}{RTT_i+\delta_i}\left(1 - \frac{1}{2}\sum_{j\in i}K\frac{q_j(t)}{C_j}W_i^2(t)\right) \\
&= \frac{1}{RTT_i+\delta_i}\cdot \\
&\quad \cdot\left(1 - \frac{1}{2}\sum_{j\in i}K\left(\frac{\tilde{q}_j(t)}{C_j} + \frac{q_j^*}{C_j}\right)\left(\tilde{W}_i(t)+W_i^*\right)^2\right) \\
&= \frac{1}{RTT_i+\delta_i}\left(1 - \frac{1}{2}(W_i^*)^2\sum_{j\in i}K\frac{q_j^*}{C_j} - \right. \\
&\quad - \frac{1}{2}\sum_{j\in i}K\frac{\tilde{q}_j(t)}{C_j}\left(\tilde{W}_i(t)+W_i^*\right)^2 - \\
&\quad \left. - \frac{1}{2}\sum_{j\in i}K\frac{q_j^*}{C_j}\left(\tilde{W}_i^2(t) + 2W_i^*\tilde{W}_i(t)\right)\right) \\
&= -\frac{1}{2}\frac{K}{RTT_i+\delta_i}\left(\sum_{j\in i}\frac{\tilde{q}_j(t)}{C_j}\left(\tilde{W}_i(t)+W_i^*\right)^2 + \right. \\
&\quad \left. + \sum_{j\in i}\frac{q_j^*}{C_j}\left(\tilde{W}_i^2(t) + 2W_i^*\tilde{W}_i(t)\right)\right)
\end{aligned} \tag{14}$$

and

$$\begin{aligned}
\dot{\tilde{q}}_j(t) &= \dot{q}_j(t) - \dot{q}_j^* \\
&= \sum_{i:j\in i}\frac{W_i(t)}{RTT_i+\delta_i} - C_j \\
&= \sum_{i:j\in i}\frac{\tilde{W}_i(t)+W_i^*}{RTT_i+\delta_i} - C_j \\
&= \sum_{i:j\in i}\frac{\tilde{W}_i(t)}{RTT_i+\delta_i} + \sum_{i:j\in i}\frac{W_i^*}{RTT_i+\delta_i} - C_j \\
&= \sum_{i:j\in i}\frac{\tilde{W}_i(t)}{RTT_i+\delta_i}
\end{aligned} \tag{15}$$

$$\dot{V}(\tilde{W}_1,\ldots,\tilde{W}_N,\tilde{q}_1,\ldots,\tilde{q}_J) = 2\sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\tilde{W}_i\dot{\tilde{W}}_i + \sum_{j=1}^{J}\frac{K}{C_j}\tilde{q}_j\dot{\tilde{q}}_j$$

$$= -\sum_{i=1}^{N}\frac{2}{(W_i^*)^2}\,\tilde{W}_i\,\frac{1}{2}\,\frac{K}{RTT_i+\delta_i}\left(\sum_{j\in i}\frac{\tilde{q}_j}{C_j}\left(\tilde{W}_i+W_i^*\right)^2+\sum_{j\in i}\frac{q_j^*}{C_j}\left(\tilde{W}_i^2+2W_i^*\tilde{W}_i\right)\right)+\sum_{j=1}^{J}\frac{K}{C_j}\tilde{q}_j\sum_{i:j\in i}\frac{\tilde{W}_i}{RTT_i+\delta_i}$$

$$= -\sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\,\tilde{W}_i\,\frac{K}{RTT_i+\delta_i}\left(\sum_{j\in i}\frac{\tilde{q}_j}{C_j}\left(\tilde{W}_i^2+2W_i^*\tilde{W}_i\right)+\sum_{j\in i}\frac{q_j^*}{C_j}\left(\tilde{W}_i^2+2W_i^*\tilde{W}_i\right)\right) -$$

$$-\sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\,\tilde{W}_i\,\frac{K}{RTT_i+\delta_i}\left(\sum_{j\in i}\frac{\tilde{q}_j}{C_j}(W_i^*)^2\right)+\sum_{j=1}^{J}\frac{K}{C_j}\tilde{q}_j\sum_{i:j\in i}\frac{\tilde{W}_i}{RTT_i+\delta_i}$$

$$= -\sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\,\tilde{W}_i^2\,\frac{K}{RTT_i+\delta_i}\left(\sum_{j\in i}\frac{\tilde{q}_j}{C_j}\left(\tilde{W}_i+2W_i^*\right)+\sum_{j\in i}\frac{q_j^*}{C_j}\left(\tilde{W}_i+2W_i^*\right)\right)$$

$$= -\sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\,\frac{K}{RTT_i+\delta_i}\,\tilde{W}_i^2(\tilde{W}_i+2W_i^*)\left(\sum_{j\in i}\frac{\tilde{q}_j+q_j^*}{C_j}\right)\le 0. \tag{17}$$

A Lyapunov function for our system is given by

$$V(\tilde{W}_1,\ldots,\tilde{W}_N,\tilde{q}_1,\ldots,\tilde{q}_J) = \sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\,\tilde{W}_i^2+\frac{1}{2}\sum_{j=1}^{J}\frac{K}{C_j}\,\tilde{q}_j^2. \tag{16}$$

It is clearly positive definite, and further (17) (shown at the top of the page) demonstrates the stability of the equilibrium. Using LaSalle's invariance principle we can show that it is in fact globally asymptotically stable. To this end we look at the set $\dot{V}^{-1}(0)$. It is

$$\dot{V}^{-1}(0) = \{(\tilde{W}_1,\ldots,\tilde{W}_N,\tilde{q}_1,\ldots,\tilde{q}_J)|\dot{V}(\tilde{W}_1,\ldots,\tilde{W}_N,\tilde{q}_1,\ldots,\tilde{q}_J)=0\}$$
$$= \{(\tilde{W}_1,\ldots,\tilde{W}_N,\tilde{q}_1,\ldots,\tilde{q}_J)|(\tilde{W}_1=\cdots=\tilde{W}_N=0) \text{ or } (\tilde{q}_1=-q_1^*,\ldots,\tilde{q}_J=-q_J^*)\}. \tag{18}$$

A simple deliberation shows that the only invariant subset contained in $\dot{V}^{-1}(0)$ is the origin. If we consider a vector from $\dot{V}^{-1}(0)$, that is not the origin, it can be easily seen from (12) and (13), that we get pushed out of $\dot{V}^{-1}(0)$. And thus the equilibrium is globally asymptotically stable. This concludes this part of the proof.

(ii) In the second step we choose the probability function for every flow to be (10).
We calculate again an equilibrium $(W_1^*,\ldots,W_N^*, q_1^*,\ldots,q_M^*)$ using (7) and (8). Next we consider the following transformation of coordinates $\tilde{W}_i(t) = W_i(t) - W_i^*$ and $\tilde{q}_j(t) = q_j(t) - q_j^*$.
For $i=1,\ldots,N$ and $j=1,\ldots,J$ we get the following normalised dynamics

$$\dot{\tilde{W}}_i(t) = -\frac{1}{2}\,\frac{1}{RTT_i+\delta_i}\cdot \cdot\left(-\left(\sum_{j\in i}\hat{K}\frac{\tilde{q}_j(t)}{C_j}\right)\left(\tilde{W}_i(t)+W_i^*\right)^2+ + \left(p_0-\sum_{j\in i}\hat{K}\frac{q_j^*}{C_j}\right)\left(\tilde{W}_i^2(t)+2W_i^*\tilde{W}_i(t)\right)\right) \tag{19}$$

$$\dot{\tilde{q}}_j(t) = \sum_{i:j\in i}\frac{\tilde{W}_i(t)}{RTT_i+\delta_i}, \tag{20}$$

where we used the same trick as for (12) and (13). A Lyapunov function that will prove instability is given by

$$V(\tilde{W}_1,\ldots,\tilde{W}_N,\tilde{q}_1,\ldots,\tilde{q}_J) = \sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\,\tilde{W}_i^2-\frac{1}{2}\sum_{j=1}^{J}\frac{\hat{K}}{C_j}\tilde{q}_j^2. \tag{21}$$

In every neighbourhood of the equilibrium we can find $(\tilde{W}_1,\ldots,\tilde{W}_N,\tilde{q}_1,\ldots,\tilde{q}_J) \in R_+^{N+J}$ such that $V(\tilde{W}_1,\ldots,\tilde{W}_N,\tilde{q}_1,\ldots,\tilde{q}_J)$ is negative. Further, (22) (placed on the top of the next page) shows the instability of the considered system by the Lyapunov Instability Theorem, see e.g. [31, Theorem 3.2.37], which is applicable because there are no invariant sets in $\dot{V}^{-1}(0)$. The argument here is the same as in (i).

(iii) Now let each flow use (11) as its probability function. Thus each flow in our system uses one of the probability functions (9) and (10), with the above defined switching rule. Let I^s denote the set of flows using initially (9) and by I^u the set of flows using initially (10). For any partition I^s, I^u of $\{1,\ldots,N\}$ (7) and (8) give us an equilibrium of our system. We will show that this equilibrium can only be stable if I^u is the empty set. If I^u is the empty set then the equilibrium is asymptotically stable. Let us assume $I^u \neq \emptyset$. The dynamic becomes

$$\dot{\tilde{W}}_i(t) = -\frac{1}{2}\,\frac{1}{RTT_i+\delta_i}\left(K\sum_{j\in i, j\in I^s}\frac{\tilde{q}_j(t)}{C_j}\left(\tilde{W}_i(t)+W_i^*\right)^2+ + K\sum_{j\in i, j\in I^s}\frac{q_j^*}{C_j}\left(\tilde{W}_i^2(t)+2W_i^*\tilde{W}_i(t)\right)-\right.$$

$$\dot{V}(\tilde{W}_1,\dots,\tilde{W}_N,\tilde{q}_1,\dots,\tilde{q}_J) = 2\sum_{i=1}^{N}\frac{1}{(W^*)^2}\tilde{W}_i\dot{\tilde{W}}_i - \sum_{j=1}^{J}\frac{\hat{K}}{C_j}\tilde{q}_j\dot{\tilde{q}}_j$$

$$= -2\sum_{i=1}^{N}\frac{1}{(W^*)^2}\tilde{W}_i\frac{1}{2}\frac{1}{RTT_i+\delta_i}\left(-\left(\sum_{j\in i}\hat{K}\frac{\tilde{q}_j}{C_j}\right)\left(\tilde{W}_i+W_i^*\right)^2+\left(p_0-\sum_{j\in i}\hat{K}\frac{\tilde{q}_j^*}{C_j}\right)\left(\tilde{W}_i^2+2W_i^*\tilde{W}_i\right)\right) -$$

$$-\sum_{j=1}^{J}\left(\hat{K}\frac{\tilde{q}_j}{C_j}\right)\sum_{i:j\in i}\frac{\tilde{W}_i}{RTT_i+\delta_i}$$

$$= -2\sum_{i=1}^{N}\frac{1}{(W^*)^2}\tilde{W}_i\frac{1}{2}\frac{1}{RTT_i+\delta_i}\left(-\left(\sum_{j\in i}\hat{K}\frac{\tilde{q}_j}{C_j}\right)\left(\tilde{W}_i^2+2W_i^*\tilde{W}_i\right)+\left(p_0-\sum_{j\in i}\hat{K}\frac{\tilde{q}_j^*}{C_j}\right)\left(\tilde{W}_i^2+2W_i^*\tilde{W}_i\right)\right)$$

$$= -\sum_{i=1}^{N}\frac{1}{(W^*)^2}\tilde{W}_i^2\frac{1}{RTT_i+\delta_i}\left(p_0-\sum_{j\in i}\hat{K}\frac{q_j}{C_j}\right)\left(\tilde{W}_i+2W_i^*\right)\le 0. \tag{22}$$

$$-\left(\sum_{j\in i,j\in I^u}\hat{K}\frac{\tilde{q}_j(t)}{C_j}\right)\left(\tilde{W}_i(t)+W_i^*\right)^2+$$

$$+\left(p_0-\sum_{j\in i,j\in I^u}\hat{K}\frac{q_j^*}{C_j}\right)\left(\tilde{W}_i^2(t)+2W_i^*\tilde{W}_i(t)\right)\Big) \tag{23}$$

$$\dot{\tilde{q}}_j(t) = \sum_{i:j\in i}\frac{\tilde{W}_i(t)}{RTT_i+\delta_i}. \tag{24}$$

The function

$$V(\tilde{W}_1,\dots,\tilde{W}_N,\tilde{q}_1,\dots,\tilde{q}_J) = \sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\tilde{W}_i^2+$$

$$+\frac{1}{2}\sum_{j\in I^s}\frac{K}{C_j}\tilde{q}_j^2-\frac{1}{2}\sum_{j\in I^u}\frac{\hat{K}}{C_j}\tilde{q}_j^2 \tag{25}$$

is a Lyapunov function that will prove instability. In every neighbourhood of the equilibrium we can find $(\tilde{W}_1,\dots,\tilde{W}_N,\tilde{q}_1,\dots,\tilde{q}_J)\in R_+^{N+J}$ such that $V(\tilde{W}_1,\dots,\tilde{W}_N,\tilde{q}_1,\dots,\tilde{q}_J)$ is negative. And further, (26) proves instability of the system by the Lyapunov Instability Theorem ([31, Theorem 3.2.37]), which is again applicable because there are no invariant sets in $\dot{V}^{-1}(0)$. The argument here is the same as in (i).

Notice how we used findings from (17) and (22) to develop (26).

■

The stability results for non-linear probability functions follow from [30, Theorem 6.8]. We state the relevant special case in the following corollary.

Corollary IV.3. *Let x^* be an hyperbolic equilibrium of the system described by* (5), (6) *and arbitrary smooth probability functions $p_i(\delta_i), i=1,\dots,N$. Then*

(i) *x^* is locally asymptotically stable for the system if for all $i=1,\dots,N$ the functions $p_i(\delta_i)$ are strictly increasing in a neighbourhood of x^*,*

(ii) *x^* is unstable for the system if for one $i=1,\dots,N$ the function $p_i(\delta_i)$ is strictly decreasing in a neighbourhood of x^*.*

The assumption of hyperbolicity[3] can not be omitted. However we suppose that it is a generic property of our system. From Corollary IV.3 it is now clear that any equilibrium of our system must be unstable if even one flow is in the high delay

[3]An equilibrium of a non-linear system is called hyperbolic, if its Jacobian does not have eigenvalues on the imaginary axis. By the continuous dependence of eigenvalues on matrix entries almost all matrices have this property.

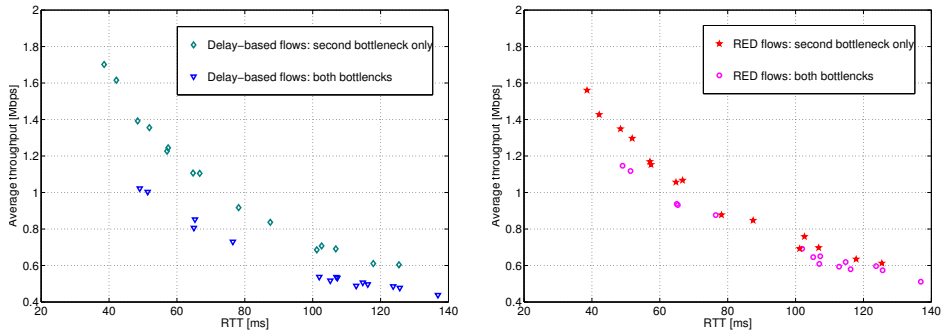
$$\dot{V}(\tilde{W}_1,\dots,\tilde{W}_N,\tilde{q}_1,\dots,\tilde{q}_n) = 2\sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\tilde{W}_i\dot{\tilde{W}}_i+\sum_{j\in I^s}\frac{K}{C_j}\tilde{q}_j^2-\sum_{j\in I^u}\frac{\hat{K}}{C_j}\tilde{q}_j\dot{\tilde{q}}_j$$

$$= -\sum_{i=1}^{N}\frac{2}{(W_i^*)^2}\tilde{W}_i\frac{1}{2}\frac{K}{RTT_i+\delta_i}\left(\sum_{j\in i,j\in I^s}\frac{\tilde{q}_j}{C_j}\left(\tilde{W}_i+W_i^*\right)^2+\sum_{j\in i,j\in I^s}\frac{q_j^*}{C_j}\left(\tilde{W}_i^2+2W_i^*\tilde{W}_i\right)\right)+\sum_{j=1,j\in I^s}K\frac{\tilde{q}_j}{C_j}\sum_{i:j\in i}\frac{\tilde{W}_i}{RTT_i+\delta_i}-$$

$$-\sum_{i=1}^{N}\frac{2}{(W_i^*)^2}\tilde{W}_i\frac{1}{2}\frac{1}{RTT_i+\delta_i}\left(\left(\sum_{j\in i,j\in I^u}\hat{K}\frac{\tilde{q}_j}{C_j}\right)\left(\tilde{W}_i+W_i^*\right)^2-\left(p_0-\sum_{j\in i,j\in I^u}\hat{K}\frac{\tilde{q}_j^*}{C_j}\right)\left(\tilde{W}_i^2+2W_i^*\tilde{W}_i\right)\right)-$$

$$-\left(\sum_{j=1,j\in I^u}\hat{K}\frac{\tilde{q}_j}{C_j}\right)\sum_{i:j\in i}\frac{\tilde{W}_i}{RTT_i+\delta_i}$$

$$= -\sum_{i=1}^{N}\frac{1}{(W_i^*)^2}\frac{1}{RTT_i+\delta_i}\tilde{W}_i^2\left(\tilde{W}_i+2W_i^*\right)\left(\left(\sum_{j\in i,j\in I^s}K\frac{q_j}{C_j}\right)+\left(p_0-\sum_{j\in i,j\in I^u}\hat{K}\frac{q_j}{C_j}\right)\right)\le 0, \tag{26}$$



(a) (b)

Fig. 11. Comparison of 30 flows in a multiple bottleneck scenario in terms of the average throughput: (a) all flows delay-based; and (b) all flows RED.

region, while the unique equilibrium in the low delay region is asymptotically stable.

Comment IV.4. *The practical interpretation of this result is the following. If we consider Fig. 1 all solutions converge to the unique attractive fixed point left of δ_{th} as long as we stay left of any equilibrium on the right side of δ_{th}. Once we get on the right side of such an equilibrium, as may happen if loss based flows are present, we might get driven further into the high delay region. But due to the fluctuations in the queue sizes driven by the multiplicative back-off behaviour of the flows, there is always a positive possibility that we reach a position left of the equilibrium. If this happens the dynamic drives the solutions to the low delay equilibrium.*

V. Case studies

We now present a number of case studies to illustrate some important features of our algorithm.

The first two are *comparative studies*. We first show that our algorithm effectively emulates a network of buffers in which RED AQM's are deployed. We then, briefly, compare C-TCP with a recently proposed delay based algorithm that purports to solve the coexistence problem; namely, the PERT algorithm [17].

Next, we present an *application study*, in which we show, again briefly, that C-TCP may be of use in certain situations where high delays are inevitable (but nevertheless undesirable). Specifically, we examine the High Speed TCP variant.

A. Comparative study #1: Emulation of RED AQM

We first show that networks in which our algorithm is deployed (and in which no loss based flows are present), have similar characteristics to networks where RED AQM schemes are deployed. These results are consistent with the simulation studies presented by Reddy et al. in [17].

We reuse the multiple bottleneck scenario, discussed previously in Section III-B. The parameters for the RED model are adjusted to be the same, as the corresponding parameters of C-TCP, namely δ_{min}, δ_{th} and p_{max}. Fig. 11 illustrates the results for the average throughput for 30 flows, half of which traverse both bottlenecks. As can be observed the performance of the proposed algorithm in terms of average throughput is similar to RED AQM. Consequently, if all flows use C-TCP the queueing delay can be kept at low levels, and the network has similar fairness properties to that of a RED network.

B. Comparative study #2: The mPERT algorithm

Next, we compare C-TCP with the recently proposed PERT strategy [17] in a single bottleneck scenario from Section III-A. PERT is a derivative of the AQM emulation work of Reddy et al., and its modified version (mPERT) [18] purports to be a delay based protocol that solves the coexistence problem. As we shall see, it is not clear that mPERT actually solves the coexistence problem, and the cost of the algorithm can be very high.

PERT, and its various embodiments, are examined experimentally in [17]. A subsequent modification, mPERT, aiming to solve the network coexistence problem, is presented in [18], please refer to these documents for further description of this work. A number of issues of concern arise upon examination of the mPERT algorithm in the coexistence context. Namely:

(i) **Fairness :** mPERT makes no attempt to ensure that mPERT based flows compete fairly with their loss-based counterparts.

(ii) **Loss rate :** The effect of increasing the aggressiveness with which mPERT flows probe for bandwidth increases the network loss rate. This in turn leads to an increase in aggressiveness with which the flows probe for bandwidth and constitutes an unstable positive feedback loop which results in high network loss rate.

(iii) **Detection :** The presence of loss-based flows is inferred through an increase in the average queueing delay beyond some threshold. This is an unreliable indicator of the presence of loss-based flows. In particular, in the presence of many mPERT flows, there is no reason to believe that the average queueing delay will reduce below this threshold once the loss-based flows leave the network.

Our set of experiments demonstrates that mPERT can be made to coexist with standard TCP, but not necessarily fairly. We present the plots obtained in a scenario with different mixes of delay/loss-based flows. Fig. 12 (on the next page) gives normalised average throughput obtained by each of the flows. Next, we examine the effect on the network loss rate when mPERT operates in its loss-based mode. Specifically, we

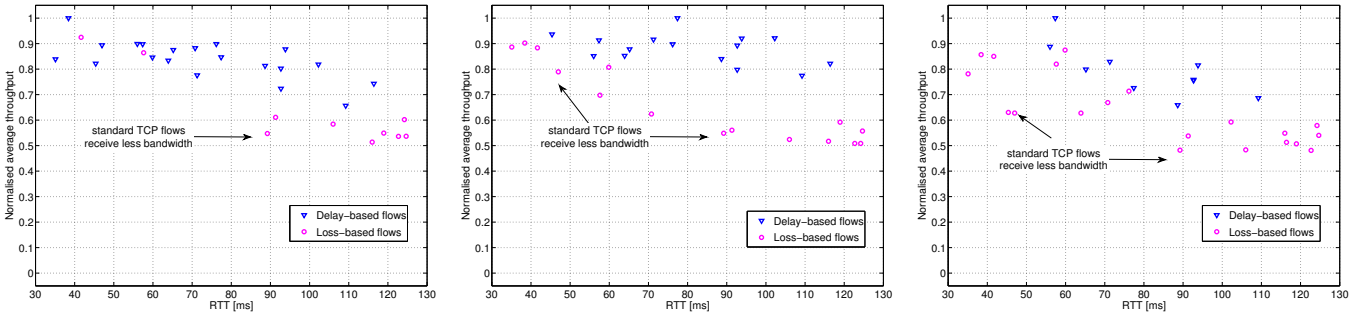


Fig. 12. Coexistence of mPERT flows and standard TCP flows in terms of the normalised average throughput for the following mixes of flows: (a) the (20,10) mix; (b) the (15,15) mix; and (c) the (10,20) mix.

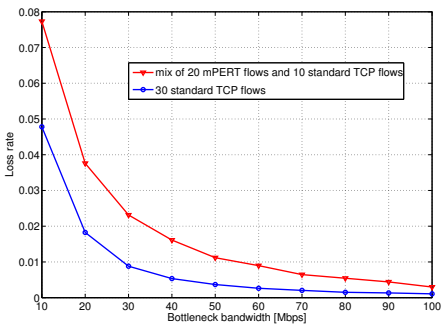


Fig. 13. Comparison in terms of loss rate: 20 mPERT flows coexisting with 10 standard TCP flows, and 30 standard TCP flows.

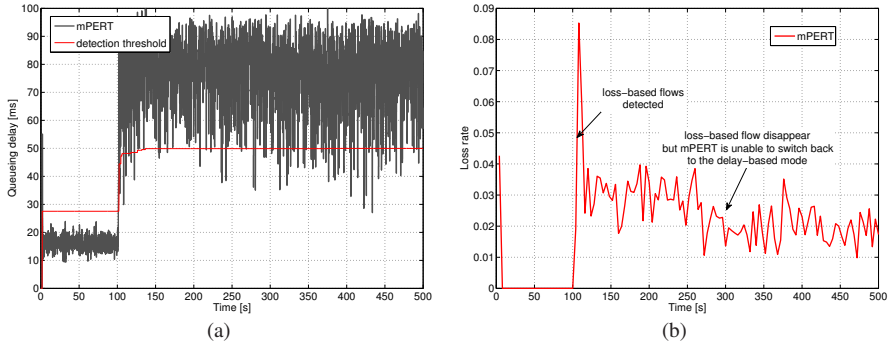


Fig. 14. Coexistence of 20 mPERT flows with 10 loss-based flows switching on and off: (a) queueing delay at the bottleneck; and (b) loss rate.

compare the behaviour of 20 mPERT flows coexisting with 10 standard TCP flows, with a scenario in which all 30 flows are standard loss-based TCP (conventional AIMD). The respective loss rates are shown in Fig. 13. As can be seen, mPERT leads to loss rates that are an order of magnitude higher than would be the case with only loss-based flows.

Finally, we consider a scenario with 20 mPERT flows and 10 intermittent loss-based flow that are active between 100 s and 300 s (we repeat the on/off switching test from Section III-A). Notice that the average queueing delay does not reduce back below this 50% threshold once the loss-based flow leaves the network. Clearly, in this situation mPERT is unable to revert to the low queueing delay regime, and stays in the loss-based mode (Fig. 14a), which is characterised by the increased network loss rate (Fig. 14b) and the high queueing delays.

C. Application Study: HighSpeed TCP

We now show that our basic idea can also be applied to enhance the performance of high speed networks. We begin by modifying Sally Floyd's HighSpeed TCP (HS-TCP, its details can be found in [32]) to incorporate our back-off strategy. In this situation our back-off strategy offers a number of benefits. These include not only a low queueing delay, but also enhance the network fairness properties by reducing the likelihood of flow synchronisation (by means of the probabilistic back off). A series of tests in a single bottleneck scenario (low delay, fairness, loss rate and on/off switching), with the same parameters as specified in Section III-A has been carried out. Fig. 15 (on the next page) illustrates the basic feature of the presented proposal, the ability to keep the queueing delay low in a homogeneous scenario consisting of flows that use HS-TCP extended with our algorithm.

We also examine the ability of delay based flows to coexist fairly with standard HS-TCP flows. Fig. 16 depicts the normalised average throughput for a network with three different mixes of standard and extended HS-TCP flows (a total of 30 flows). Again, as in case of standard TCP, we can observe that the proposed algorithm guarantees fair coexistence in terms of average throughput rates, between loss and delay based flows. Next, we check the network loss rate. We observe in Fig. 17 that HS-TCP flows using the proposed algorithm experience similar loss rates as standard HS-TCP flows. In particular, a comparison of 20 delay-based HS-TCP flows (using the proposed algorithm) coexisting with 10 standard HS-TCP flows with a scenario in which all 30 flows are standard (loss-based) HS-TCP, witness no significant difference in network loss rate.

Finally, we test dynamic properties of the presented algorithm in a high speed network scenario consisting of 20 delay-based HS-TCP flows and 10 intermittent standard HS-TCP flow (we repeat analogous on-off switching test from Section III-A. Fig. 18 illustrates the results of such an experiment both in terms of queueing delay and packet loss ratio. It can be clearly seen that the algorithm is able to detect the presence of loss based flows and then, once these flow are off, is able to revert back to the low queueing delay operation.

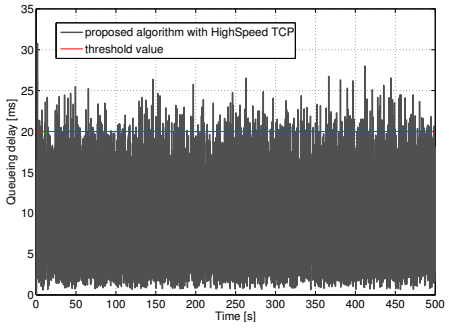


Fig. 15. Delay-based HS-TCP in a single bottleneck scenario: queueing delay for 30 flows.

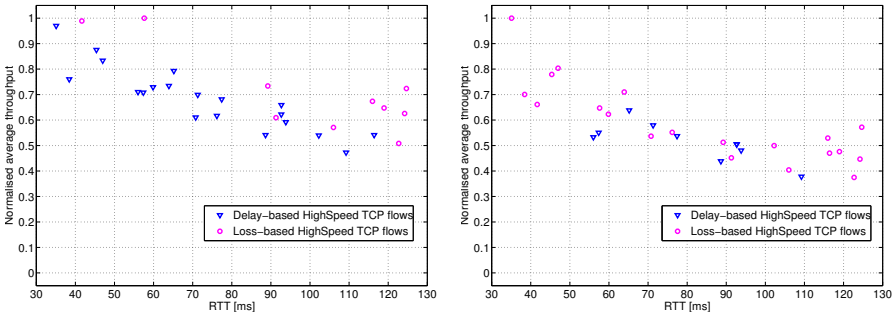


Fig. 16. Coexistence of delay-based HS-TCP flows and standard HS-TCP flows in terms of the normalised average throughput for the following mixes of flows: (a) the (20,10); and (b) the (10,20) mix.

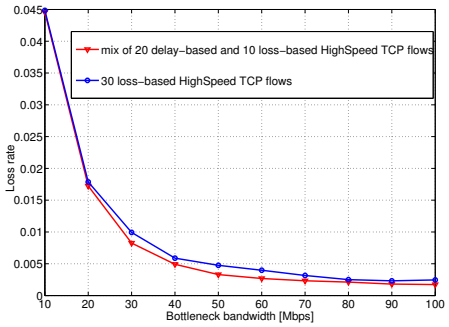


Fig. 17. Comparison in terms of loss rate: 20 delay-based HS-TCP flows coexisting with 10 standard HS-TCP flows, and 30 standard HS-TCP flows.

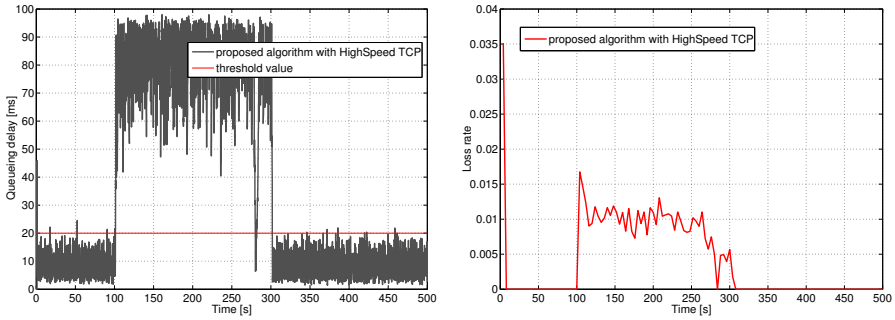


Fig. 18. Coexistence of 20 delay-based HS-TCP flows with 10 loss-based HS-TCP flows (standard HS-TCP) switching on and off: (a) queueing delay at the bottleneck; and (b) loss rate.

VI. Conclusion

In this paper we have presented a method that can be used to ensure that delay-based AIMD flows operate as loss-based flows when loss-based flows are present in the network, allowing fair coexistence with their loss-based counterparts, and otherwise revert to delay-based operation mode. We demonstrated both in experiments and analytically that the proposed solution guarantees the aforementioned features, and that appropriate adjusting of the maximum equilibrium loss rate permits wide application of C-TCP, covering a range of different scenarios, e.g., fast networks. Finally, to conclude the description of presented algorithm we denote a number of potential limitations of C-TCP. First, our algorithm works best in multiplexed environments with standing queues. In situations where this assumption is not valid, some unfairness may be introduced when loss based flows are present. A crucial part of the algorithm is the assumption that all flows use the same per-packet back-off probability function and sense the same queueing delay. If this assumption is not valid, unfairness can be introduced. Also, the mathematical analysis presented in Section IV is only valid in situations when the fluid model of TCP provides a description of the queueing dynamics. However, the qualitative description of the mechanism provided in Section II is valid, and is independent of the fluid model.

Acknowledgement

The authors would like to thank Fabian Wirth from the Institut für Mathematik, Universität Würzburg, Germany for help with the stability proof for the multiple bottleneck scenarios (Section IV).

References

[1] V. Jacobson, "Congestion avoidance and control," in *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*. New York, NY, USA: ACM, 1988, pp. 314–329.

[2] R. Jain, "A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks," *ACM Comp. Commun. Rev.*, vol. 19, no. 5, pp. 56–71, 1989.

[3] A. Tang, J. Wang, S. Low, and M. Chiang, "Equilibrium of heterogeneous congestion control protocols," in *Proc. of the 24th IEEE INFOCOM Conference*, vol. 2, March 2005, pp. 1338– 1349.

[4] L. Brakmo, S. O'Malley, and L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in *Proc. of ACM SIGCOMM Conference*, 1994, pp. 24–35.

[5] L. Brakmo and L. Peterson, "TCP Vegas: End to end congestion avoidance on a global internet," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, October 1995.

[6] Z. Wang and J. Crowcroft, "Eliminating periodic packet losses in the 4.3-Tahoe BSD TCP congestion control algorithm," *ACM Comp. Commun. Rev.*, vol. 22, no. 2, pp. 9–16, 1992.

[7] W. Feng and S. Vanichpun, "Enabling compatibility between TCP Reno and TCP Vegas," in *Proc. of the Symp. on Appl. and the Internet*, 2003, pp. 301– 308.

[8] K. Srijith, L. Jacob, and A. Ananda, "TCP Vegas-A: Solving the fairness and rerouting issues of TCP Vegas," in *Proc. of*

22nd IEEE International Performance, Computing, and Communications Conference (IPCCC), April 2003, pp. 309–316.
[9] U. Hengartner, J. Bolliger, and T. Gross, "TCP Vegas revisited," in *Proc. of the 19th IEEE INFOCOM Conference*, 2000, pp. 1546–1555.
[10] H. Choe and S. Low, "Stabilized Vegas," in *Proc. of the 22nd IEEE INFOCOM Conference*, vol. 3, March-April 2003, pp. 2290–2300.
[11] S. Low, L. Peterson, and L. Wang, "Understanding Vegas: a duality model," *Journal of the ACM*, vol. 49, no. 2, pp. 207–235, March 2002.
[12] R. Jain and K. Ramakrishnan, "Congestion avoidance in computer networks with a connectionless network layer: concepts, goals and methodology," *Computer Networking Symposium*, pp. 134–143, 1988.
[13] J. Wang, D. X. Wei, and S. Low, "Modelling and stability of FAST TCP," in *Proc. of the 24th IEEE INFOCOM Conference*, vol. 2, 2005, pp. 938– 948.
[14] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound TCP approach for high-speed and long distance networks," 2005, Technical Report - Microsoft Research MSR-TR-2005-86.
[15] D. Leith, J. Heffner, R. Shorten, and G. McCullagh, "Delay-based AIMD congestion control," in *Proc. of 5th Int'l Workshop on Protocols for FAST Long-Distance Networks (PFLDnet)*, 2007.
[16] D. Leith, R. Shorten, G. McCoullagh, L. Dunne, and F. Baker, "Making available base RTT for use in congestion control applications," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 429–431, June 2008.
[17] S. Bhandarkar, A. Reddy, Y. Zhang, and D. Loguinov, "Emulating AQM from end hosts," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 349–360, October 2007.
[18] K. Kotla and A. Reddy, "Making a delay-based protocol adaptive to heterogeneous environments," in *Proc. of 16th International Workshop on Quality of Service (IWQoS 2008)*, June 2008, pp. 100–109.
[19] J. Martin, A. Nilsson, and I. Rhee, "Delay-based congestion avoidance for TCP," *IEEE/ACM Trans. Netw.*, vol. 11, no. 3, pp. 356–369, June 2003.
[20] R. Prasad, M. Jain, and C. Dovrolis, "On the effectiveness of delay-based congestion avoidance," in *Proc. of 2nd Int'l Workshop on Protocols for FAST Long-Distance Networks (PFLDnet)*, 2004.
[21] G. McCullagh, "Exploring delay based TCP congestion control," Master's thesis, Hamilton Institute, NUI Maynooth, Ireland, 2008.
[22] C. Kellett, R. Shorten, and D. Leith, "A review of delay-based congestion control," 2006, Cisco Report - Distributed to project partners, March 2006.
[23] A. Tang, J. Wang, S. Low, and M. Chiang, "Equilibrium of heterogeneous congestion control: existence and uniqueness," *IEEE/ACM Trans. Netw.*, vol. 15, no. 4, pp. 824–837, 2007.
[24] A. Tang, X. Wei, S. Low, and M. Chiang, "Equilibrium of heterogeneous congestion control: Optimality and stability," *to appear in IEEE/ACM Trans. Netw.*
[25] Ł. Budzisz, R. Stanojević, R. Shorten, and F. Baker, "A strategy for fair coexistence of loss and delay-based congestion control algorithms," *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 555–557, July 2009.
[26] Ł. Budzisz, R. Stanojević, A. Schlote, R. Shorten, and F. Baker, "On the fair coexistence of loss- and delay-based TCP," in *17th International Workshop on Quality of Service*, 2009.
[27] "The network simulator ns-2," http://www.isi.edu/nsnam/ns/.
[28] D. Bauso, L. Giarre, and G. Neglia, "AQM stability in multiple bottleneck networks," in *Proc. of IEEE International Conference on Communications, 2004 (ICC 2004)*, vol. 4, June 2004, pp. 2267–2271.
[29] L. Wang, L. Cai, X. Liu, X. S. Shen, and J. Zhang, "Stability analysis of multiple-bottleneck networks," *Computer Networks*, vol. 53, no. 3, pp. 338–352, 2009.
[30] A. Schlote, "Stability of TCP models," Diplomarbeit, Julius-Maximilians-Universität Würzburg, November 2009.
[31] D. Hinrichsen and A. Pritchard, *Mathematical Systems Theory I. Modelling, state space analysis, stability and robustness*. Berlin: SpringerVerlag, 2004, vol. 48.
[32] S. Floyd, "RFC 3649, HighSpeed TCP for large congestion windows," http://www.ietf.org/rfc/rfc3649.txt, December 2003.

PLACE PHOTO HERE

Łukasz Budzisz (S'05-M'09) received the M.S. degree in Electronics and Telecommunication from the Technical University of Łódź, Poland, in 2003, and the Ph.D. degree in Signal Theory and Communications from the Technical University of Catalunya, Spain, in 2009. He is currently a research fellow at Hamilton Institute, Ireland. His research interest include network congestion control, mobility and wireless networks.

PLACE PHOTO HERE

Rade Stanojević Biography text will be provided with the camera-ready version.

PLACE PHOTO HERE

Arieh Schlote received his Diploma in mathematics from the University of Würzburg, Germany, in 2010. He is now a PhD student at the Hamilton Institute, Ireland, where he works on stability issues of communication networks. His research interests include stability theory of dynamical systems and linear algebra.

PLACE PHOTO HERE

Robert Shorten Biography text will be provided with the camera-ready version.

PLACE PHOTO HERE

Fred Baker Biography text will be provided with the camera-ready version.