# ACK-Clocking Dynamics: Modelling the Interaction between Windows and the Network

Krister Jacobsson*, Lachlan L. H. Andrew†, Ao Tang†, Karl H. Johansson*, Håkan Hjalmarsson*, Steven H. Low†
*KTH, Stockholm, SE-100 44, Sweden; †California Institute of Technology, Pasadena, CA 91125, USA

***Abstract*—A novel continuous time fluid flow model of the dynamics of the interaction between ACK-clocking and the link buffer is presented. A fundamental integral equation relating the instantaneous flow rate and the window dynamics is derived. Properties of the model, such as well-posedness and stability, are investigated. Packet level experiments verify that this new model is more accurate than existing models, correctly predicting qualitatively different behaviors, for example when round trip delays are heterogeneous.**

## I. Introduction

The Transmission Control Protocol (TCP) is the predominant transport protocol of the Internet today, carrying about 83% of the total traffic volume [1]. Since Jacobson's work on the Tahoe release of BSD Unix in 1988 [2], many modifications and replacements have been proposed [2–9] to meet the demands of a modern Internet scaled up in size and capacity.

The research effort on congestion control has been considerable after 1988. However, most proposed algorithms are *window based*, meaning that a source explicitly controls a window size, that is the number of packets that are sent before the sender must wait for an acknowledgment packet. Research has focused on how to determine that window size.

Today's TCP NewReno [3] (with or without SACK [4]) is in principal similar to its predecessor TCP Tahoe, relying on packet losses as a congestion indicator to trigger a rapid decrease in the window size, and trusting that flows will see appropriately matched loss rates to ensure fairness. Another widely discussed source of congestion information is the delay experienced by packets [7], [8]. There exist many experimental TCP proposals ranging between purely loss-based versions like CUBIC [5] and H-TCP [6], and purely delay based schemes like TCP Vegas [7] and FAST TCP [8], with many algorithms that use both delay and loss as congestion measures such as TCP Africa [9] and TCP Illinois [10].

All of these rely on detailed dynamics of instantaneous rates and network queue sizes, either to determine which flow's packet is being received at the exact time a packet is dropped, or to determine the precise queuing delays. In window based schemes, ACK-clocking governs these sub-RTT phenomena. Despite its importance, the dynamics of the window mechanism is still not well understood.

### A. Window-based transmission control

A schematic picture of the control structure for window-based transmission control is displayed in Figure 1. The dynamics of the endpoint protocol are represented by the three blocks: transmission control, window control, and congestion estimator. The system consists of an inner loop and an outer loop. In the outer loop, the window control adjusts the transmission window size based on the estimated congestion level of the network. This congestion level is estimated based on the ACKs, which carry implicit (often corrupted) information in the form of duplicate, missing and delayed ACKs.

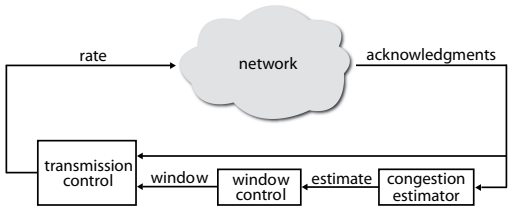


Fig. 1. System view in window-based congestion control.

### B. ACK-clocking

The dynamics of the inner loop is given by so called ACK-clocking. The transmission of new packets is controlled or "clocked" by the stream of received ACKs by the transmission control. A new packet is transmitted for each received ACK, thereby keeping the number of outstanding packets, i.e. the window, constant. More sophisticated traffic shaping could also be considered, but we do not consider such dynamics in this paper.

The design of the outer loop, i.e., the window adjustment mechanism, has received ample attention in the literature [2–9]. The properties of ACK-clocking are often ignored. For example, ACK-clocking has stabilizing properties in itself. Furthermore, ACK-clocking operates at a per-packet time-scale. This makes it better suited to handle short-term queue fluctuations than the outer-loop, that typically adjusts the window on a round trip time basis.

### C. Network fluid flow modeling

To ensure that the network will reach and maintain a favorable equilibrium, it is important to assess its dynamical properties such as stability and convergence. Instability means that small fluctuations due to varying cross traffic are amplified, and manifests itself as severe oscillations in aggregate traffic quantities, such as queue lengths. Following the seminal

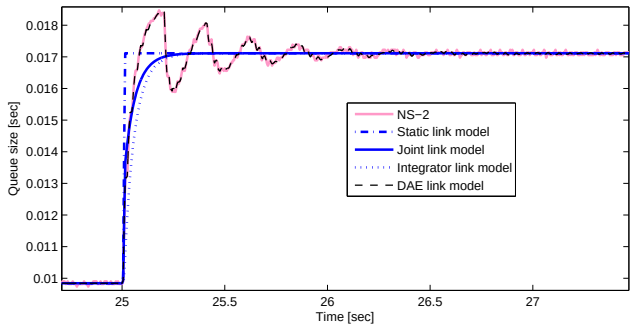


Fig. 2. The new model presented in this paper is able to capture a dynamic step response much better than traditional models in the literature. The graph shows the queue size. Two window based flows with propagation delays $d_1 = 10\,\text{ms}$ and $d_2 = 190\,\text{ms}$ are sharing the bottleneck link. The window of the first source is subject to a step after 25 s.

work by Kelly [11] there have been numerous studies on network stability. Network fluid flow models, where packet level information is discarded and traffic flows are assumed to be smooth in space and time, have shown to be useful in such analysis, for example in [8,12–16]. The validity of results concerning dynamical properties, however, rely heavily on the accuracy of the models. Models with fundamentally different dynamical properties have been used to model ACK-clocking in window-based schemes (often referred to as "the link"), i.e., the inner loop in Fig. 1. In [12–15] an "integrator" link model is used, integrating the approximate excess rate on the link. On the other hand, in [17] transients are ignored and a "static" link model is proposed. Furthermore, in [18] a "joint" link model combining the immediate and long term integrating effect is proposed and used for stability analysis in [16]. The motivating example in Section II illustrates the limited accuracy of these models. This is further elaborated on in a companion paper [19], which highlights the need for incorporating important microscopic sub-RTT effects in macroscopic fluid flow models.

A new link model which captures these sub-RTT effects is derived in Section III. Properties of this model are found in Section IV, and it is rigorously validated in Section V. Conclusions are drawn in Section VI.

## II. A MOTIVATING EXAMPLE

Consider a system of two window based flows sending over a single bottleneck link with capacity 100 Mbit/s, with 1040 byte packets, where the sources' window sizes are kept constant, i.e., the outer loop in Fig. 1 is disabled. The round trip delays excluding the queuing delay are $d_1 = 10\,\text{ms}$ and $d_2 = 190\,\text{ms}$. The window sizes are initially $w_1 = 210$ and $w_2 = 1500$ packets respectively. After convergence, at 25 seconds, $w_1$ is increased step-wise from 210 to 300 packets. The solid pink line in Fig. 2 shows the bottleneck queue size (in seconds) when this scenario is simulated in NS-2, exhibiting significant oscillation in the queue. This is in contrast to the dash-dotted, solid and dotted blue lines in Fig. 1, showing predictions made by existing models of the inner loop dynamics (see [16] for a discussion). They all predict smooth convergence similar to first order filter step responses (with varying time constants), the reasons for which will be discussed in Section IV-F. The dashed black line shows the continuous time fluid model derived in this paper, it shows almost perfect agreement with the packet level simulation, even at sub-RTT time scales.

The analysis of the dynamic properties of a window based system based upon any of the previous models may yield qualitatively different results than those from the more accurate model proposed here, especially for TCPs responding in part to queuing delay [7]–[9]. This is also confirmed in [19].

## III. MODELLING

### A. Preliminaries

A network is modeled as consisting of $L$ links with capacities $c_l$ and time varying queuing delays $p_l(t)$, $l = 1, \ldots, L$. Traffic consists of $N$ flows, with $w_n(t)$ the time varying number of packets "in flight" (sent but not acknowledged). The instantaneous rate at which traffic from flow $n$ enters link $l$ is $x_{l,n}(t)$, or $x_n(t)$ in the single-link case. The round trip time between the time a packet of flow $n$ enters link $l$ and the time that the "resulting" packet transmitted in response to the acknowledgment of that packet enters link $l$ is denoted $\tau_{l,n}(t)$. It consists of a fixed component $d_n$ and a time varying component due to queuing delays. In the single-link case, $\tau_n(t) = d_n + p(t)$.

Link $l$ carries cross traffic $x_{l,c}(t)$ which is not window controlled. Cross traffic is assumed for simplicity to not use more than one link and is not included in the routing.

Packets are assumed to be transmitted greedily and in FIFO order at links, which reflects the reality of the current Internet.

### B. The Single Source Single Bottleneck Case

Consider first the simplest case of a single window flow control source sharing a single link with non-window cross traffic of known rate. In this section, the subscripts will be dropped for clarity, and forward propagation delay is assumed without loss of generality to be zero.

*1) Instantaneous rate:* To discover what can be known about the instantaneous transmission rate based on knowledge of the window size, consider an arbitrary time $t$. Packets transmitted up to time $t$ will be acknowledged by time $t+\tau(t)$, and thus the number of packets "in flight" at time $t+\tau(t)$, namely $w(t+\tau(t))$, will exactly equal those packets transmitted in the interval $(t, t+\tau(t)]$. That is,

$$\int_t^{t+\tau(t)} x(T)dT = w(t+\tau(t)). \tag{1}$$

(This equation was introduced in passing in [20], but not pursued.) Most models approximate the integral in (1) by a product, yielding $x(t) \approx w(t)/\tau(t)$; one exception is [18] which instead considered an embedded discrete time sequence $t_{k+1} = t_k + \tau(t_k)$, yielding an exact mean rate of $x_k = w_k/\tau(t_{k-1})$ over the interval $(t_{k-1}, t_k]$. Fig. 3 shows how (1) can be interpreted as a sliding window of such averages.

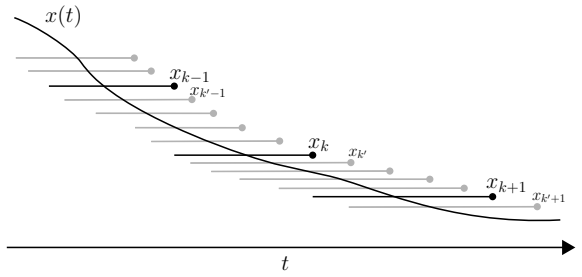


Fig. 3. The input rate of the source into the queue. The sequences $x_k, x_{k'}, \ldots$ which represents averages over an interval $(t_k, t_k + d + p(t_k)]$, $(t_{k'}, t_{k'} + d + p(t_{k'})]$, $\ldots$ are known; we seek the function $x(t)$.

Differentiating (1) with respect to time gives

$$(1 + \dot{\tau}(t))x(t + \tau(t)) - x(t) = (1 + \dot{\tau}(t))\dot{w}(t + \tau(t)). \quad (2)$$

Rearranging, and shifting the time point, gives

$$x(t) = \frac{x(t - \tau(\tilde{t}))}{1 + \dot{\tau}(t - \tau(\tilde{t}))} + \dot{w}(t) \quad (3)$$

where $\tilde{t}$ solves $t = \tilde{t} + \tau(\tilde{t}) = \tilde{t} + d + p(\tilde{t})$. Note that the rate at time $t$ is not determined solely by the window and RTT, but depends on the rate one RTT previously; that is the origin of the sub-RTT rate dynamics studied in [19].

*2) ACK-clocking model:* In terms of rates, a link buffer is simply an integrator, integrating the excess rate at the link (modulo static non-linearities present in the system, such as non-negativity constraints and packet drops). Thus, having defined the instantaneous rate $x(t)$, the buffer dynamics are naturally given by

$$\dot{p}(t) = \frac{1}{c}\left(x_n(t) + x_c(t) - c\right). \quad (4)$$

The whole system is described by the delay *Differential Algebraic Equation* (DAE) defined by (1) and (4).

### C. Multiple Sources, One Bottleneck

When multiple flows share the bottleneck, there are $N$ constraints analogous to (1), expressing the rate $x_n(t)$ for each flow; which is combined with an integration similar to (4). Without loss of generality, assume zero forward propagation delay and we have the DAE model

$$\dot{p}(t) - \frac{1}{c}\left(\sum_{n=1}^{N} x_n(t) + x_c(t) - c\right) = 0, \quad (5a)$$

$$\int_t^{t+\tau_n(t)} x_n(T)\, dT - w_n(t + \tau_n(t)) = 0, \quad (5b)$$

for $n = 1, \ldots, N$. The model is supported by numerical results in Section V.

### D. The Multiple Sources Multiple Bottlenecks Case

Consider now a network with multiple bottlenecks. The objective is to find a highly accurate model, perhaps at the expense of simplicity. The result is a reference model, which can be a starting point for tractable approximations (c.f. Section IV-F).

Let $R = (r_{ln})$ be the $N \times L$ routing matrix, with $r_{ln} = 1$ if link $l$ is used by flow $n$, and 0 otherwise. Bidirectional links are modeled as distinct unidirectional links.

The first row of (5) then becomes

$$c_l \dot{p}_l(t) = \sum_{n=1}^{N} R_{l,n} x_{l,n}(t) + x_{c;l}(t) - c_l. \quad (6a)$$

It remains to determine $x_{l,n}(t)$ analogously to (1). This case is significantly more complex since packets experience delays at different instants of time at each link.

Let $\tau_{l,n}(t)$ be the round trip time from when a packet from source $n$ arrives at link $l$ to the arrival at link $l$ of the "resulting" packet—the packet sent as a result of the acknowledgment of the first. Similarly, let $\tau^f_{l,n}(t)$ be the time from when a packet released from source $n$ reaches link $l$ and

$$w_{l,n}(t + \tau^f_{l,n}(t)) = w_n(t). \quad (6b)$$

The instantaneous rate $x_{l,n}(t)$ then satisfies

$$\int_t^{t+\tau_{l,n}(t)} x_{l,n}(T)\, dT = w_{l,n}(t + \tau_{l,n}(t)). \quad (6c)$$

It remains to calculate $\tau_{l,n}(t)$ and $\tau^f_{l,n}(t)$. To do this, it is necessary to keep track of the order of the links along each source's path. Let $\vec{p}_{l,n}(t)$ be a column vector of the same dimension as the number of links in the $n$th sources path, say $L_n$. The elements of $\vec{p}_{l,n}(t)$ are the queue sizes in the path of source $n$, ordered from the point of view of source $n$ and of link $l$. Thus the first element corresponds to the queue size of link $l$, the second element the queue size of the link *downstream* of link $l$ in the $n$th source's path, and so on, and finally the last element corresponds to the link queue *upstream* of link $l$. The $i$th element in a vector $\vec{p}_{l,n}(t)$ is denoted $\vec{p}_{l,n,i}(t)$.

The ordered propagation delay $\vec{d}_{l,n}$ can be defined similarly as for the queuing delays. So $\vec{d}_{l,n,i}$ represents the propagation delay between link $l$ and the link $i-1$ hops after $l$ on path $n$, and where by convention $\vec{d}_{l,n,L_n+1} = d_n$. Note that $\vec{d}_{l,n,1} = 0$, and, if $l$ is the $k$th and $l'$ is the $k'$th link on path $n$ where $k' > k$ (link $l'$ is downstream link $l$), then $\vec{d}_{l,n,1+k'-k} + \vec{d}_{l',n,L_n+1-(k'-k)} = d_n$ by definition.

Let $l$ be the $m(l,n)$th link on path $n$ and let $\hat{\tau}_{l,n,i}(t)$ be the delay such that a packet which arrives at link $l$ at time $t$ arrives at the link $i-1$ hops after $l$ on path $n$ at time $t + \hat{\tau}_{l,n,i}(t)$. (Strictly, the packet which arrives may be an acknowledgment or a "resulting" packet.) The total delay, including the queuing at each link, is then

$$\hat{\tau}_{l,n,i}(t) = \vec{d}_{l,n,i} + \sum_{k=1}^{i-1} \vec{p}_{l,n,k}(t + \hat{\tau}_{l,n,k}(t)). \quad (6d)$$

The interval of integration in (6c) is then simply

$$\tau_{l,n}(t) = d_n + \sum_{i=1}^{L_n} \vec{p}_{l,n,i}(t + \hat{\tau}_{l,n,i}(t)) = \hat{\tau}_{l,n,L_n+1}(t). \quad (6e)$$

Similarly, the forward delay linking $w_{l,n}(t)$ with $w_n(t)$ is

$$\tau^f_{l,n}(t) = \hat{\tau}_{\ell(n),n,m(l,n)+1}(t), \tag{6f}$$

where $\ell(n)$ is a "link" located at the source of flow $n$, introduced to model propagation delay between the source and the first (bottleneck) link included in the routing.

In summary, the model of ACK-clocking dynamics for a system of $N$ window based sources utilizing a network of $L$ links is given by (6). The accuracy of the model is investigated in Section V-B.

## IV. Analysis

This section shows the uniqueness of the equilibrium of the general model (6), and then for the single link case proves that the queuing delays are locally asymptotically stable but the rates may possibly have sustained oscillations.

### A. Equilibrium

ACK-clocking can be interpreted as a congestion control algorithm applied at the source, with queuing delay as price signal fed back from the network, and with tuning parameter $w$ (window size). In this context, we are able to apply the utility optimization framework to characterize the equilibria in the following theorem.

*Theorem 1:* For given positive vectors $w$, $d$ and $c$, the equilibrium rates $x^*$ of the ACK-clocking model (6) are unique, and if $R$ is full rank, then the queuing delays $p^*$ are also unique.

*Proof:* A feasible equilibrium point $(x^*, p^*)$ satisfies $\sum_{n=1}^N R_{l,n}x^*_n \le c_l$ for all $l, n$ and

$$x^*_n := w_n/(d_n + q^*_n), \quad q^*_n = \sum_{l=1}^L R_{l,n}p^*_l, \quad p^*_l \ge 0.$$

The parameters $w_n, d_n, c_l$ are fixed.[1] The equilibrium point can be expressed as

$$\sum_{l=1}^L R_{l,n}p^*_l = q^*_n = \frac{w_n}{x^*_n} - d_n. \tag{7}$$

Let

$$U_n(x^*_n) = w_n \log(x^*_n) - d_n x^*_n \tag{8}$$

which is strictly concave. Note that (7) is the Karush-Kuhn-Tucker condition to the convex program

$$\max_{x\ge 0} \quad \sum_{n=1}^N U_n(x_n), \quad \text{s.t.} \quad Rx \le c \tag{9}$$

with compact feasible set. Thus there exists a unique optimal solution $x^*$, see [11,22], and by (7), a unique $q^*$. Assume there exist two optimal queuing delay vectors $p^*$ and $\tilde{p}^*$, then

$$R^T(p^* - \tilde{p}^*) = q^* - q^* = 0. \tag{10}$$

If $R$ has full row rank, then the columns of $R^T$ are linearly independent and thus $p^* = \tilde{p}^*$. Therefore, if $R$ has full row rank then the equilibrium $(x^*, p^*)$ is unique. ∎

[1] c.f. a similar proof in e.g. [8]

### B. Linearization around equilibrium

In order to study the stability, let us linearize (5) around its equilibrium $(p, w, x, x_c)$. Following the convention that time delays in variables' arguments are modeled by their equilibrium values yields, for $n = 1, \dots, N$,

$$\dot{p}(t) - \sum_{n=1}^N x_n(t)/c - x_c(t)/c = 0, \tag{11a}$$

$$x_n\dot{p}(t) - \dot{w}(t+\tau_n) + x_n(t+\tau_n) - x_n(t) = 0. \tag{11b}$$

Here variables now denote small perturbations. Taking the Laplace transform gives an explicit expression of the sources' queue input rates

$$x_n(s) = \frac{s}{e^{-s\tau_n} - 1}\left(x_n e^{-s\tau_n}p(s) - w_n(s)\right). \tag{12}$$

Thus the linear ACK-clocking dynamics are described by

$$\left(c + \sum_{n=1}^N x_n \frac{e^{-s\tau_n}}{1 - e^{-s\tau_n}}\right) p(s) = \sum_{n=1}^N \frac{w_n(s)}{1 - e^{-s\tau_n}} + \frac{1}{s}x_c(s). \tag{13}$$

Modeling non-zero forward propagation delay, $\tau^f_n$, is achieved simply by multiplying $w_n(s)$ by $e^{-s\tau^f_n}$ in (13). The linear model is validated in Section V-A2 and used for analysis below.

### C. Stability

As pointed out in [2], window flow control is stable in the sense that signals remain bounded. The following theorem shows the stronger result that the linearized single bottleneck dynamics (13) relating the windows $w$ to the queue $p$ are asymptotically stable, ruling out persistent oscillations in these quantities, at least locally. Let $\mathbb{C}^+$ be the open right half plane, $\{z : \mathrm{Re}(z) > 0\}$, and $\bar{\mathbb{C}}^+$ be its closure, $\{z : \mathrm{Re}(z) \ge 0\}$.

*Theorem 2:* For all $0 < x_n \le c$, $\tau_n > 0$, $n = 1, \dots, N$, the function $G_{pw} : \bar{\mathbb{C}}^+ \to \mathbb{C}^{1\times N}$ whose $i$th element is given by

$$G_{pw_i}(s) = \frac{1}{(1 - e^{-s\tau_i})\left(c + \sum_{n=1}^N x_n \frac{\exp(-s\tau_n)}{1-\exp(-s\tau_n)}\right)}, \tag{14}$$

is stable.

*Proof:* It is sufficient to confirm that [21]:

(a) $G_{pw}(s)$ is analytic in $\mathbb{C}^+$;

(b) for almost every real number $\omega$,

$$\lim_{\sigma\to 0^+} G_{pw}(\sigma + j\omega) = G_{pw}(j\omega);$$

(c) $\sup_{s\in\bar{\mathbb{C}}^+} \bar{\sigma}(G_{pw}(s)) < \infty$

where $\bar{\sigma}$ denotes the largest singular value.

Conditions (a) and (b) are satisfied if they hold element-wise. Furthermore

$$\sup_{s\in\bar{\mathbb{C}}^+} \bar{\sigma}(G_{pw}(s)) = \sup_{s\in\bar{\mathbb{C}}^+} \sqrt{\bar{\lambda}(G_{pw}(s)G^*_{pw}(s))}$$

$$= \sup_{s\in\bar{\mathbb{C}}^+} \sqrt{\sum_{i=1}^N G_{pw_i}(s)\overline{G}_{pw_i}(s)} \le \sum_{i=1}^N \sup_{s\in\bar{\mathbb{C}}^+} |G_{pw_i}(s)|. \tag{15}$$

Thus, condition (c) holds if

$$\inf_{s\in\bar{\mathbb{C}}^+} |1/G_{pw_i}(s)| > 0. \tag{16}$$

It is therefore sufficient to establish (a), (b) and (c) for the $i$th transfer function element $G_{pw_i}(s)$.

Start with the boundedness condition (c). It is sufficient to show that there is no sequence $s_l = \sigma_l + j\omega_l \in \bar{\mathbb{C}}^+$ with $\lim_{l\to\infty} |1/G_{pw_i}(s_l)| = 0$. This will be established by showing that the limit evaluated on any convergent subsequence is greater than 0. Consider a subsequence with $\sigma_l \to \sigma$, $\omega_l \to \omega$.
**Case 1**, $\sigma = \infty$: $1/G_{pw_i(s_l)} \to c > 0$.
**Case 2**, $\sigma \in (0, \infty)$: By the triangle inequality,

$$|1 - e^{-s_l\tau_i}| \geq \left|1 - |e^{-s_l\tau_i}|\right| \to 1 - e^{-\sigma\tau_i} > 0. \tag{17}$$

Furthermore, $1/(e^{s_l\tau_n} - 1)$ lies on the circle with center $1/(A_l^2 - 1) + j0$ and radius $A_l/(A_l^2 - 1)$, where $A_l = |e^{s_l\tau_n}|$. Thus $\lim_{l\to\infty} \mathrm{Re}(1/(e^{s_l\tau_n} - 1)) \geq -1/(e^{\sigma\tau_n} + 1)$, hence

$$\lim_{l\to\infty} \mathrm{Re}\left(c + \sum_{n=1}^{N} \frac{x_n}{e^{s_l\tau_n} - 1}\right) \geq c - \sum_{n=1}^{N} \frac{x_n}{e^{\sigma\tau_n} + 1} =$$
$$c - \sum_{n=1}^{N} x_n + \sum_{n=1}^{N} \frac{x_n e^{\tau_n\sigma}}{e^{\tau_n\sigma} + 1} \geq \sum_{n=1}^{N} \frac{x_n}{1 + e^{-\tau_n\sigma}} \geq \sum_{n=1}^{N} \frac{x_n}{2} > 0. \tag{18}$$

Multiplying (17) and (18) gives $\lim_{l\to\infty} |1/G_{pw_i}(s_l)| > 0$.
**Case 3**, $\sigma = 0$: Note that $\mathrm{Re}(1/(e^{j\omega_l\tau_n} - 1)) = -1/2$, so

$$\lim_{l\to\infty} \mathrm{Re}\left(c + \sum_{n=1}^{N} \frac{x_n}{e^{(\sigma_l + j\omega_l)\tau_n} - 1}\right) = c - \sum_{n=1}^{N} \frac{x_n}{2}$$
$$\geq c - \frac{c}{2} > 0. \tag{19}$$

Thus $\lim_{l\to\infty} |1/G_{pw_i}(s_l)| \neq 0$ except possibly when the first factor of (14) $1 - e^{-s_l\tau_i} \to 0$, which occurs when $\omega\tau_i = 2\pi m$, $m \in \mathbb{Z}$. Let

$$\mathbf{I}_n = \begin{cases} 1 & \text{if } m\tau_n/\tau_i \in \mathbb{Z}, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Now

$$\lim_{s\to j2\pi m/\tau_i} |1/G_{pw_i}(s)|$$
$$= \lim_{s\to j2\pi m/\tau_i} \left| c(1 - e^{-s\tau_i}) + x_i + \sum_{\substack{n=1 \\ n\neq i}}^{N} x_n e^{-s\tau_n} \frac{1 - e^{-s\tau_i}}{1 - e^{-s\tau_n}} \right|$$
$$= x_i + \sum_{\substack{n=1 \\ n\neq i}}^{N} x_n \frac{\tau_i}{\tau_n} \mathbf{I}_n > 0, \tag{21}$$

using L'Hôpital's rule in the second step when $\mathbf{I}_n = 1$. Thus $\lim_{l\to\infty} |1/G_{pw_i}(s_l)| > 0$ for all sequences $s_l$ in $\bar{\mathbb{C}}^+$ for which the limit exists, whence (16) holds, and thus (c).

Furthermore, since $1/G_{pw_i}(s) \neq 0$, $G_{pw_i}(s)$ is also non-singular in $\bar{\mathbb{C}}^+$, and therefore analytic as its components are analytic. (Constants as well as the exponential function are entire, i.e., analytic in $\mathbb{C}$; also note that sums, differences, and products of analytic functions are analytic; quotients of analytic functions are analytic except where the denominator equals zero.) This establishes (a). Condition (b) holds since $G_{pw_i}(s)$ is analytic in $\bar{\mathbb{C}}^+$. ■

### *D. Uniqueness of rates*

The results presented until now hold for any $x(t)$ satisfying (11), leaving open the question of uniqueness. It is possible for the windows not to define unique rates, due to sub-RTT burstiness. Consider a network in which two flows with equal RTTs $\tau$ share a bottleneck link of capacity $C$, and each has window $C\tau/2$. If the flows alternate between sending at rate $C$ for time $\tau/2$ and sending at rate 0 for $\tau/2$, and if the "on" periods of flow 1 coincide exactly with the "off" periods of flow 2, then the total rate flowing into the bottleneck link is constant, and (11) is satisfied. It is also satisfied if both sources send constantly at rate $C/2$.

For a single bottleneck, the rates will be unique unless one flow has a RTT which is a rational multiple of another flow's RTT. (Note that if RTTs are drawn randomly, then this will occur with probability 0.)

To see this, note that sustained oscillations in the rate for a constant window correspond to marginally stable (pure imaginary) poles of (11). Taking the Laplace transform of (11) and eliminating $p$, gives

$$\mathrm{diag}(se^{s\tau_k})w(s) = \left(\frac{1}{c}\mathrm{diag}(x_k)E + \mathrm{diag}(e^{s\tau_k} - 1)\right) x(s), \tag{22}$$

where $E_{k,l} = 1$ for all $k, l = 1, \ldots, N$. Since $\mathrm{diag}(se^{s\tau_k})$ is never singular for $s \neq 0$, the poles of (22) are the non-zero values of $s$ for which the coefficient of $x(s)$ is singular. The only imaginary values for which this occurs are when $s\tau_i = j2\pi b$ and $s\tau_k = j2\pi a$ for some $i, k = 1, \ldots, N$ and integers $a$ and $b$.

For $N = 2$ flows with equal mean rates $x_1 = x_2 = C/2$, the linearized rates are given by

$$\frac{x_1(s)}{w_1(s)} = \frac{s(2 - e^{-s\tau_2})}{2 - e^{-s\tau_1} - e^{-s\tau_2}} \tag{23a}$$
$$\frac{x_2(s)}{w_1(s)} = \frac{-se^{-s\tau_2}}{2 - e^{-s\tau_1} - e^{-s\tau_2}}. \tag{23b}$$

Note that $x_1(s) = -x_2(s)$ for $s = j2\pi b/\tau_i$. This highlights the fact that the sustained oscillations maintain a constant rate flowing into the bottleneck link. Conversely, for any periodic function $x_1(t)$ with period $\tau_i/b$, perturbations about the mean with $x_2(t) = -x_1(t)$ will satisfy (11).

Since any ratio of round trip times can be approximated arbitrarily closely by a rational number, it might seem that this sustained oscillation would be common. Several factors may contribute to it not having been reported regularly. Firstly, the periodicity would be interrupted by changes in the window due to congestion control. Secondly, many studies report window sizes, queue sizes or rates estimated as the ratio of window over RTT. Thus, this effect may have occurred in many experiments in which it was not reported. Thirdly, although

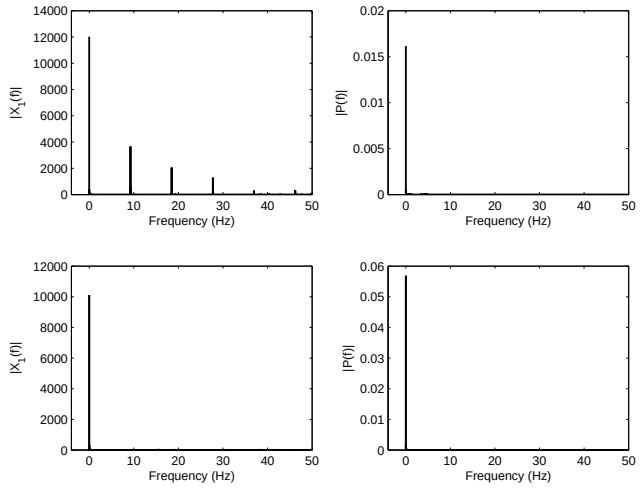


Fig. 4. Single-sided amplitude spectrum for the rate of source 1 (left) and bottleneck queue (right). Upper plots: $a = 2, b = 1$. Lower plots: $a = 2079, b = 1352$.

any rational ratio of RTTs will lead to sustained oscillations, those oscillations may be at a high frequency and attributed to "packet level noise". Oscillation occurs at $\min(a, b)$ times per RTT for the smaller RTT flow. Even if $\tau_1/\tau_2$ is not exactly rational, or is a ratio of large integers, slowly decaying oscillations may exist corresponding to an approximation $\tau_1/\tau_2 \approx b/a$ for smaller $a$ and $b$. For $\epsilon = b\tau_2 - a\tau_1$, there is a pole with $\sigma + j\omega \approx -(2\pi\epsilon)^2/(\tau_1+\tau_2)^3 + j2\pi(a+b)(\tau_1+\tau_2)$.

*Example 1:* Consider two window based flows sharing a bottleneck link with capacity $c = 100$ Mbit/s, with 1040 byte packets, and an equilibrium queuing delay of $p = 8.16$ ms. First, a scenario with $w_1 = 650$ packets, $w_2 = 2w_1$, $d_1 = 100$ ms, $d_2 = 208.16$ ms is simulated in NS-2. For this case $\tau_2 = 2\tau_1 = 216.32$ ms. This rational ratio $a = 2, b = 1$ suggests sustained oscillations at frequencies $f = kb/\tau_1 \approx (9.25k)$ Hz where $k \in \mathbb{Z}^+$. The upper left plot in Fig. 4 shows the single-sided amplitude spectrum (computed by the FFT) of the sending rate of source 1, sampled every 5 ms. The spikes in the plot agree with our prediction. The upper right hand plot of the amplitude spectrum of the queue size lacks such sustained oscillations; it is stable in line with Theorem 2. (While the individual sending rates oscillate, their sum does not, yielding a stable link buffer.)

The two lower plots are for a similar scenario, with instead $d_2 = 158.16$ ms. Since $\tau_1/\tau_2 = 1352/2079$, the sustained oscillations will be at $f = kb/\tau_1 = (12500k)$ Hz. The approximations $b/a = 2/3$ has a decay time constant of approximately 1/8 s, while higher order approximations give frequencies off the scale of this graph. Accordingly, the amplitude spectrum of the source rate lacks spikes except at the zero mode. Again the queue is non-oscillatory.

### *E. Relation between network window and congestion window*

Throughout this paper, the term "window" has referred to the *network window*, the number of packets outstanding in the network. This need not be the same as the window flow control's desired *congestion window*, because a reduction in the desired window size cannot immediately withdraw packets from the network.

Recall that the network window $w$ is defined from the equality

$$\int_t^{t+d+p(t)} x(T)\, dT = w(t + d + p(t)), \quad t \geq 0,$$

where $x$ is the instantaneous sending rate, $p$ the queuing delay, and $d$ the propagation delay. Let $0 < t_1 < t_2 < \dots$ be the time instances when an ACK arrives at the sender. Denote the sequence of consecutive instances a round-trip time apart as $0 < \mathbf{t}_1 < \mathbf{t}_2 < \dots$, and suppose that $\mathbf{t}_1 = t_1$, i.e., the first packet was sent at $t = 0$. Note that the set of round-trip times $\{\mathbf{t}_\ell\}$ is a subset of the ACK times $\{t_k\}$.

Given the congestion window $\bar{w}$, the corresponding sending rate $x$ is not uniquely defined. The congestion window only determines what the *average* rate over a round-trip time should be. It is common that if $\bar{w}$ is increased by a certain number of packets these packets are instantaneously put on the network, while if $\bar{w}$ is decreased the decrease on the network depends on when the next few ACKs arrive to the sender. Many other implementations of the congestion window changes are possible, e.g., smoothing a window increase over a certain time interval. Obviously, the relation between the actual sending rate and the congestion window depends on the protocol implementation. Next, we derive some fundamental bounds on the difference between $\bar{w}$ and $w$.

In general, it is hard to obtain a better bound on the pointwise difference between $\bar{w}$ and $w$ than

$$\sup_{t\geq 0} |\bar{w}(t) - w(t)| \leq \sup_{t\geq 0} \bar{w}(t) = \bar{w}_{\max},$$

where $\bar{w}_{\max}$ denotes the largest congestion window over a session. (Note that The inequality follows simply from the previous argument: if $\bar{w}$ is decreased, then $w$ is not instantaneously decreased but decreases only as the next ACKs arrive to the sender.

Let us next consider the $L_1$ norm of the difference between $\bar{w}$ and $w$:

$$\|\bar{w} - w\|_{L_1} = \sup_{T>0} \frac{1}{T} \int_0^T |\bar{w}(t) - w(t)| dt.$$

It is thus a measure of how close $\bar{w}$ and $w$ are in average. Let us first assume that $\bar{w}$ is updated only at the round-trip times $\mathbf{t}_\ell$, $\ell = 1, 2, \dots$. Note that for $t \in (\mathbf{t}_\ell, \mathbf{t}_{\ell+1})$, it holds that $w(t) = w(\mathbf{t}_\ell) + (t - \mathbf{t}_\ell)w'(\xi)$ for some $\xi \in (\mathbf{t}_\ell, \mathbf{t}_{\ell+1})$. Hence, $|w(\mathbf{t}_\ell) - w(t)| \leq (\mathbf{t}_{\ell+1} - \mathbf{t}_\ell)w'(\xi)$. With $w'_{\max} = \sup_{\ell=1,2,\dots} w'(\xi(\ell))$ and $\mathrm{RTT}_{\max} = \sup_{\ell=1,2,\dots}(\mathbf{t}_{\ell+1} - \mathbf{t}_\ell)$, we have thus $|w(\mathbf{t}_\ell) - w(t)| \leq \mathrm{RTT}_{\max}\mathrm{w}'_{\max}$. Then,

$$\begin{aligned}\|\bar{w} - w\|_{L_1} &= \sup_{T>0} \frac{1}{T} \int_0^T |w(\mathbf{t}_{\ell(t)}) - w(t) dt \\ &\leq \sup_{T>0} \frac{1}{T} \int_0^T \mathrm{RTT}_{\max}\mathrm{w}'_{\max}\mathrm{dt} \\ &\leq \mathrm{RTT}_{\max}\mathrm{w}'_{\max}.\end{aligned}$$

The approximation error between $\bar{w}$ and $w$ is thus of the order of the round-trip time.

If on the other hand, $\bar{w}$ is updated at each ACK arrival $t_k$, $k = 1, 2, \ldots$, then for $t \in (t_k, t_{k+1})$, it holds that $w(t) = w(t_k) + (t - t_k)w'(\xi)$ for some $\xi \in (t_k, t_{k+1})$. Similar to above we obtain

$$\begin{aligned}\|\bar{w} - w\|_{L_1} &= \sup_{T>0} \frac{1}{T} \int_0^T |w(t_{k(t)}) - w(t)dt \\ &\leq \sup_{T>0} \frac{1}{T} \int_0^T \mathrm{ACK_{max} w'_{max} dt} \\ &\leq \mathrm{ACK_{max} w'_{max}},\end{aligned}$$

where $\mathrm{ACK_{max}} = \sup_{\mathrm{k=1,2,\ldots}}(\mathrm{t_{k+1}} - \mathrm{t_k})$. In this case, the approximation error between $\bar{w}$ and $w$ is thus of the order inter-arrival time of the ACKs, which is often much smaller than the round-trip time.

To summarize, we have proven the following theorem.

*Theorem 3:* The following relations between the congestion window $\bar{w}$ and the network window $w$ hold:

- if $\bar{w}(t)$ is updated at $t \in \{\mathbf{t}_\ell\}_{\ell=1,2,\ldots}$, then $\bar{w}(\mathbf{t}_\ell) = w(\mathbf{t}_\ell)$ and $\|\bar{w} - w\|_{L_1} \leq \mathrm{RTT_{max} w'_{max}}$;
- if $\bar{w}(t)$ is updated at $t \in \{t_k\}_{k=1,2,\ldots}$, then $\bar{w}(t_k) = w(t_k)$ and $\|\bar{w} - w\|_{L_1} \leq \mathrm{ACK_{max} w'_{max}}$.

### F. Relation to existing models

The model may be simplified by approximating the integral equation (1) defining the instantaneous rate, and the $N$ integral constraints in (5). Let $H_t(z) = \int_t^z x(T)dT - w(z)$. By (1), $H_t(t + \tau(t)) = 0$. Standard approximations to $H_t(z)$ yield several popular models. Intuitively, better approximations of the constraint should lead to greater model accuracy. Note however that, due to coupling between the constraints (1) and the integration (4), even though we are able to quantify the accuracy of the approximation of $H_t(z)$, more rigorous analysis is needed to formalize the resulting accuracy in the queuing delay $p$ for the different models. This is left for future work, and thus the discussion here is heuristic.

*1) Ratio models:* Most common models take $x_n(t) \approx w(t-\Delta_a)/\tau(t-\Delta_b)$, for some choice of $\Delta_a$ and $\Delta_b$ [12–16]. Applying the right-side rectangle rule to $H_t(t + \tau(t))$ gives $x_n(t + \tau(t)) \approx w_n(t + \tau(t))/\tau_n(t) + \mathcal{O}(\tau)$ whence

$$x_n(t) \approx w_n(t)/\tau_n(t - \tau_n(\tilde{t})) \tag{24}$$

where $\tilde{t}$ satisfies $\tilde{t} + \tau_n(\tilde{t}) = t$. This is similar to the integrator model shown in [17] to be overly pessimistic for large RTTs. More accurate numerical quadrature rules can also be applied. For example, the trapezoidal rule gives a recursive rule

$$x_n(t + \tau_n(t)) \approx 2w_n(t + \tau_n(t))/\tau_n(t) - x_n(t). \tag{25}$$

Note that rules such as the midpoint rule which do not evaluate the end point of the interval, $x(t+\tau_n(t))$, will yield non-causal models, with $x(t)$ dependent on a future value of $w(t + \ldots)$.

By further assuming in (24) that the deviation from the equilibrium rates are negligible, $x_n(t) = x_n + \delta x_n(t) \approx x_n$, we get a static update of the queue in terms of window updates as suggested in [17].

*2) "Joint" models:* Taylor expansion of $H_t$ around $t$ yields

$$\begin{aligned}0 &= H_t(t + \tau(t)) = H_t(t) + H'_t(t)\tau(t) + \mathcal{O}(\tau^2) \\ &= -w(t) + (x(t) - \dot{w}(t))\tau(t) + \mathcal{O}(\tau^2).\end{aligned} \tag{26}$$

Dividing by $\tau_n(t)$ gives the rate used by the "joint link model" [18] as an $\mathcal{O}(\tau)$ approximation

$$x_n(t) \approx w_n(t)/\tau_n(t) + \dot{w}_n(t). \tag{27}$$

Ignoring the $\dot{w}_n(t)$ in (26) gives $x_n(t) \approx w_n(t)/\tau_n(t)$. If $\dot{w}_n = \mathcal{O}(\tau_n)$ then this is again an $\mathcal{O}(\tau_n)$ approximation, albeit less accurate than (27); otherwise it is $\mathcal{O}(\dot{w}_n)$.

Taking higher order terms in the Taylor expansion of $H(t+\tau(t))$ gives more accurate models. However, this leads to high order ODE models.

*3) Models by Padé approximations:* An alternative is to study the linearized model in the Laplace domain (13), and use, for example, different orders of Padé approximations to $e^{-s\tau_n}$. In this context a (0,0) Padé approximation (i.e. $e^{-s\tau_n} \approx 1$) gives the "static link model" introduced in [17], while the "joint link model" [18] corresponds to a (0,1) approximation. By a (1,0) approximation, a time-scaled ratio model is achieved, c.f. [12–16]. A suitable order of approximation can be chosen, and a nonlinear ODE may then be "reverse engineered" to approximate the DAE model. This approach is used with good accuracy in the linear validation example in Section V-A2.

All of the above models are based on small $\tau$ approximations. However, $\tau(t)$ need not be small; in particular $\tau(t)$ does not approach zero in the fluid limit of many packets. Thus, (1) should be used whenever it results in a tractable problem formulation, such as the analysis of loss synchronization and stability of delay based protocols in [19].

## V. Model Validation

In this section the model derived in Section III is validated. The model is simulated in Simulink, and the simulation output is compared with packet level data achieved using NS-2. Note that in all experiments we only execute positive changes of the window $w(t)$ (remember it represents the packets "in flight" here). This is to decouple the dynamics of the studied mechanism from the dynamics of the inherited traffic shaping. Recall that a negative change is dependent on the rate of received ACKs.

### A. Single link network

*1) Nonlinear model:* For the single link case we refer to the motivating example in Section II due to limited space. The solid pink line in Fig. 2 shows the queue size when the system is simulated in NS-2, the dashed black line the DAE model (5). The model fits almost perfectly.

*2) Linearized model:* Two window based flows are sending over a bottleneck link with capacity $c = 100$ Mbit/s. There is no non-window based cross traffic, so $x_c = 0$. Initially, $w_1 = 60$ packets and $w_2 = 2000$ packets, with packet size $\rho = 1040$ byte. Furthermore, $d_1 = 10$ ms and $d_2 = 190$ ms,

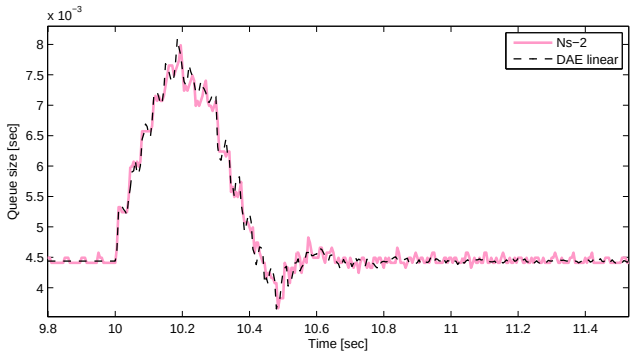


Fig. 5. Validation example. Solid line: NS-2 simulation. Dashed line: Continuous time DAE model (5).

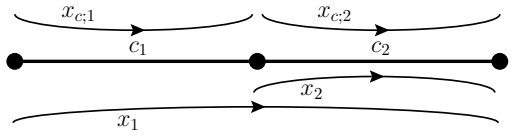


Fig. 6. Network configuration validation example

with no forward delay. The system is started in equilibrium, and $w_1$ is increased by 10 at $t = 10$ s, and 300 ms later it is decreased back to 60. The solid line in Fig. 5 shows the queue size when the system is simulated in NS-2, the dashed line the linear approximation (13). The model is good, so the linear approximation seems valid. (In the simulation of (13), a Padé approximation of order (17,17) of the exponential functions has been used.)

### B. Multiple link network

The multi flow multi link ACK-clocking model (6) is validated using a scenario with two flows sending over a network of two bottleneck links (indexed 1 and 2). The configuration is according to Figure 6. The first flow utilizes both links, and in the view of this source, the first link is upstream the second link. The second flow is sending over the second link only. Furthermore, there may exist non window based cross traffic sending over the individual links. For all simulations $c_1 = 80$ Mbit/s, $c_2 = 200$ Mbit/s, $d_1 = 100$ ms, $d_2 = 200$ ms, and packet size $\rho = 1040$ byte. Furthermore, the first source is located at the first link and thus $\ell(1) = 1$, while the second flow is located at a non-bottleneck "link" $\ell(2)$ upstream the second link (modeling forward propagation delay). Configuration is such $\vec{d}_{1,1,2} = \vec{d}_{2,1,2} = 50$ ms and $\vec{d}_{\ell(2),2,2} = 50$ ms. The system is perturbed from equilibrium at $t = 15$ s by a positive step change in one of the sources window of magnitude 50 packets. The queue sizes of the simulated DAE model (simulated in Simulink) is compared with NS-2 data.

*1) Case 1: no cross traffic:* No traffic except the two window based sources are present, so $x_{c;1} = x_{c;2} = 0$. Furthermore $w_1^0 = 2100$ and $w_2^0 = 3900$ packets. In Fig. 7 the system is perturbed from equilibrium by a step change in the window of the first source. We observe that it is only, in the

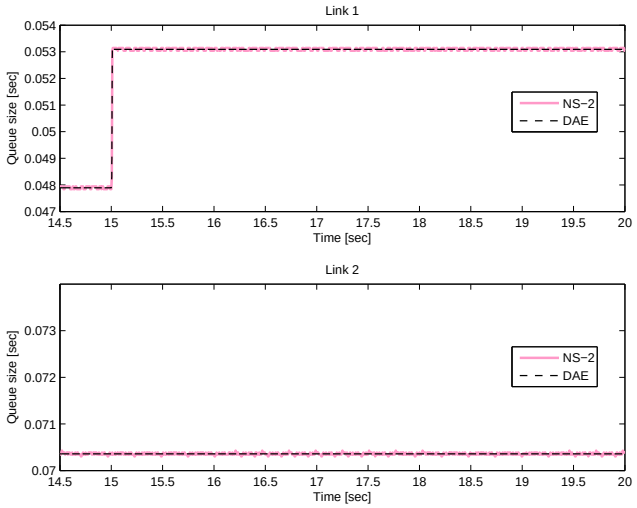


Fig. 7. Simulation example. No cross traffic. Step change in window 1. Solid line: NS-2. Dashed line: DAE model.

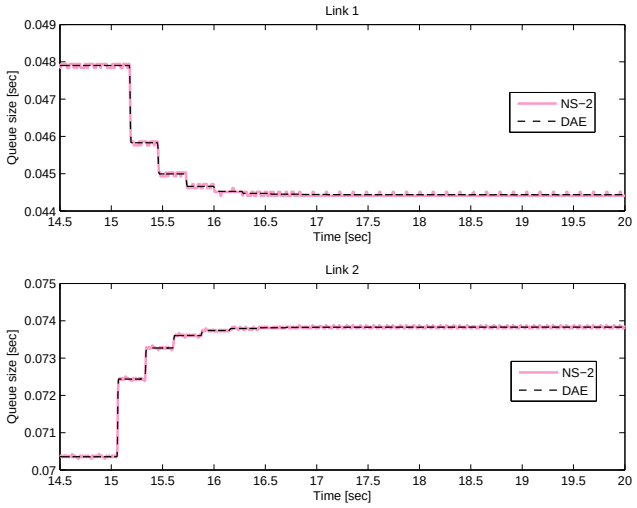


Fig. 8. Simulation example. No cross traffic. Step change in window 2. Solid line: NS-2. Dashed line: DAE model.

view of the first source, the upstream queue that is affected. This is due to that the traffic travelling from link 1 to link 2 is saturated by the capacity of link 1. This blocking property is captured by the model.

On the other hand, from Fig. 8, which corresponds to a scenario when the window of the second source is changed, both queues are affected even though link 2 is downstream link 1 from the first source point of view. This is because that each source actually is operating in closed loop, and that the ACK rate of the first source is affected by the change in the queue size of link 2. Moreover we observe that the model fit is very good, the discrepancies are of the magnitude $\rho/c_l$ and hence seem to be due to quantization. The burstiness in the link buffer is captured.

*2) Case 2: cross traffic on link 1:* In this scenario UDP cross traffic is sending over link 1 utilizing half the capacity,

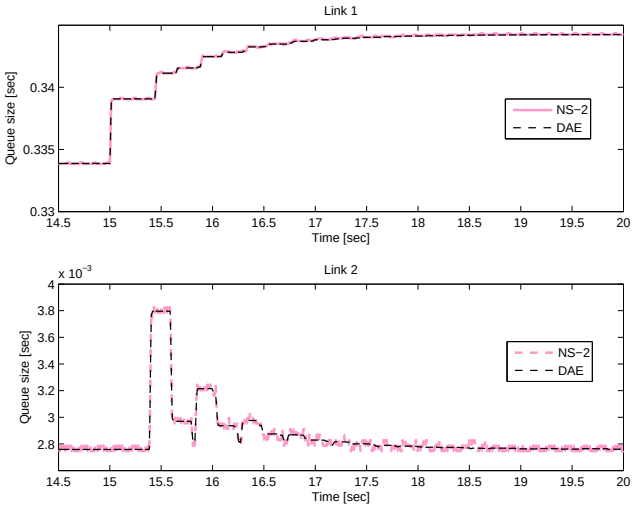


Fig. 9. Simulation example. Cross traffi c on link 1. Step change in window 1. Solid line: NS-2. Dashed line: DAE model.

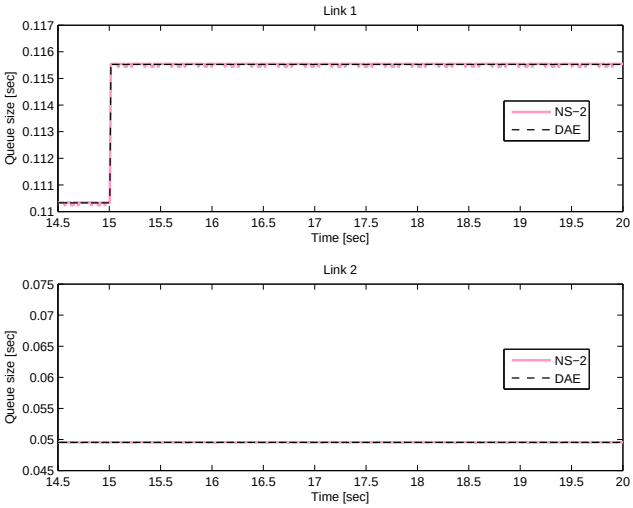


Fig. 11. Simulation example. Cross traffi c on link 2. Step change in window 1. Solid line: NS-2. Dashed line: DAE model.

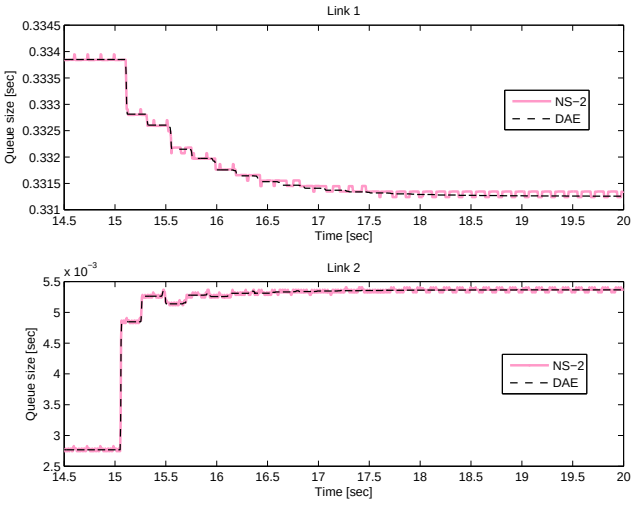


Fig. 10. Simulation example. Cross traffi c on link 1. Step change in window 2. Solid line: NS-2. Dashed line: DAE model.

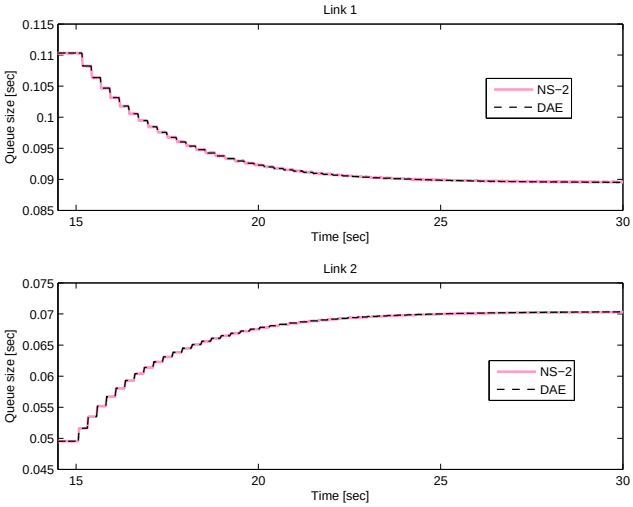


Fig. 12. Simulation example. Cross traffi c on link 2. Step change in window 2. Solid line: NS-2. Dashed line: DAE model.

i.e., $x_{c;1}(t) = c_1/2$; and initially $w_1^0 = 2100$ and $w_2^0 = 3900$ packets. The plot in Fig. 9 displays the queue sizes when the congestion window of the first source is changed. While Fig. 10 shows when a step is applied to the second sources window. Note that the change of the first source affects the second buffer size for this case since the flow between the links are not saturated anymore on a shorter time scale.

*3) Case 3: cross traffic on link 2:* In this scenario UDP cross traffic is sending over link 2 and utilizes half the capacity, i.e., $x_{c;2}(t) = c_2/2$; and initially $w_1^0 = 2500$ and $w_2^0 = 600$ packets. The plot in Fig. 11 displays the queue sizes when the congestion window of the first source is increased step wise. As in the first simulation case, the second queue is not affected. The explanation is analogous. The plot in Fig. 12 corresponds to the case when the second source is perturbed. Here both queues are affected by the window perturbation, and the transient is significant for this case.

*4) Case 4: cross traffic on both links:* In this scenario, UDP sources are sending over both links independently, each of them utilizing half the capacity of the link, i.e., $x_{c;1}(t) = c_1/2$ and $x_{c;2}(t) = c_2/2$. We also initially have $w_1^0 = 1000$ and $w_2^0 = 1500$ packets. The plot in Fig. 13 displays the queue sizes when the congestion window of the first source is increased. While Fig. 14 shows when a step is applied to the second sources window.

In summary, the model shows very good agreement with the packet level data. It captures sub-RTT effects such as burstiness besides those more long term behaviors.

## VI. Conclusion

We have rigorously analysed the dynamics of ACK-clocking in window-based congestion control, deriving a new fluid flow model. The model is shown in packet level simulations to be very accurate and qualitatively different from its predecessors.

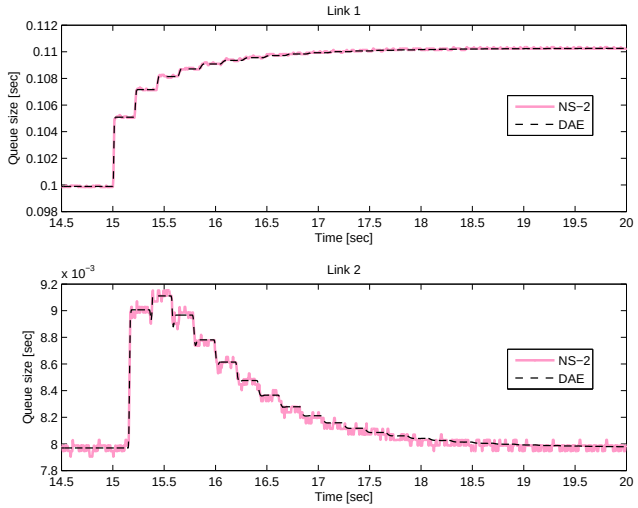


Fig. 13. Simulation example. Cross traffic on both links. Step change in window 1. Solid line: NS-2. Dashed line: DAE model.

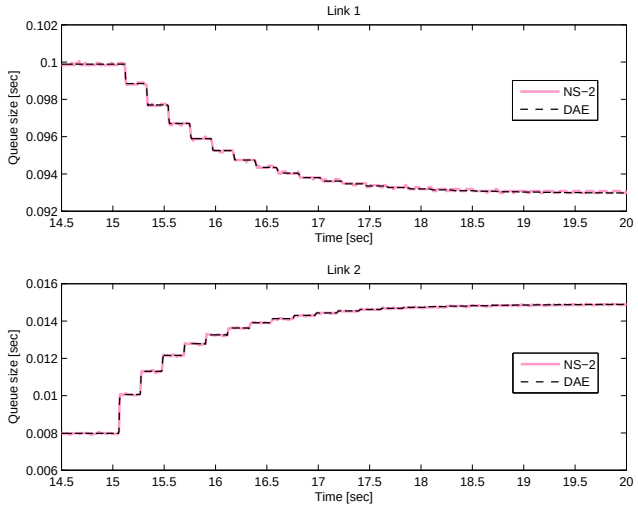


Fig. 14. Simulation example. Cross traffic on both links. Step change in window 2. Solid line: NS-2. Dashed line: DAE model.

We define the instantaneous rate each window based source inputs to the individual links by a fundamental integral equation. This is in contrast to the customary approach for approximating the window based sources' sending rates, and is the key in the modeling. We prove that the system has a unique equilibrium. Furthermore we show that a linear approximation of the model around the equilibrium is asymptotically stable from window to the queue. All existing models in the literature are shown to be certain approximations to this exact new model. This procedure also provided us with insight how to derive other simplified models.

A natural application of the model is stability analysis of window based congestion control algorithms. Since the model captures sub-RTT burstiness it can be used to analyze, e.g., loss synchronization. Analyzing how such microscopic effects influences macroscopic properties is future work, although exciting initial steps are given in a companion paper [19]. It also remains to explore the implications of the model for general networks.

## Acknowledgments

The authors thank Niels Möller for useful discussions. This work was supported by the KTH ACCESS Linnaeus Centre, the Swedish Research Council, the Swedish Foundation for Strategic Research, and the European Commission through the Network of Excellence HYCON and the Integrated Project RUNES; by NSF under grants 0435520 and EIA-0303620.

## References

[1] M. Fomenkov, K. Keys, D. Moore, and K. Claffy. Longitudinal study of Internet traffic in 1998-2003. In *WISICT '04: Proc. Winter Int. Symp. Info. Commun. Technol.*, 2004.

[2] V. Jacobson. Congestion avoidance and control. *ACM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 314–329, 1988.

[3] S. Floyd and T. Henderson. The NewReno modification to TCP's fast recovery algorithm. RFC 2582, April 1999.

[4] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP selective acknowledgements options RFC 2018, October 1996.

[5] L. Xu, K. Harfoush and I. Rhee. Binary Increase Congestion Control for Fast Long-Distance Networks. In *Proc. IEEE INFOCOM*, 2004.

[6] D. J. Leith, R. Shorten. H-TCP Protocol for High-Speed Long Distance Networks. In *Proc. PFLDnet*, 2004.

[7] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson. TCP Vegas: new techniques for congestion detection and avoidance. In *Proc. ACM SIGCOMM*. 1994, pp. 24–35.

[8] D. Wei, C. Jin, S. H. Low, and S. Hegde. FAST TCP: motivation, architecture, algorithms, performance. *IEEE/ACM Trans. Networking*, December 2006.

[9] R. King, R. Baraniuk and R. Riedi. TCP-Africa: An Adaptive and Fair Rapid Increase Rule for Scalable TCP. In *Proc. IEEE INFOCOM*, 2005.

[10] S. Liu, T. Basar and R. Srikant. TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks. In *Proc. First Int. Conf. on Perform. Eval. Methodol. Tools (VALUETOOLS)*, 2006.

[11] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *J. Op. Res. Soc.*, 49:237–252, 1998.

[12] E. Altman, C. Barakat, and V. Ramos. Analysis of AIMD protocols over paths with variable delay. In *Proc. IEEE INFOCOM*, 2004.

[13] F. Baccelli and D. Hong. AIMD, fairness and fractal scaling of TCP traffic. In *Proc. IEEE INFOCOM*, 2002.

[14] C. Hollot, V. Misra, D. Towsley, and W. B. Gong. A control theoretic analysis of RED. In *Proc. IEEE INFOCOM*, Anchorage, AK, April 2001, pp. 1510–1519.

[15] S. H. Low, F. Paganini, and J. C. Doyle. Internet congestion control. *IEEE Control Systems Magazine*, 22(1):28–43, Feb. 2002.

[16] A. Tang, K. Jacobsson, L. L. H. Andrew and S. H. Low. An accurate link model and its application to stability analysis of FAST TCP. In *Proc. IEEE INFOCOM*, 2007.

[17] J. Wang, D. X. Wei, and S. H. Low. Modeling and stability of FAST TCP. In *IMA Volumes in Mathematics and its Applications*, Volume 143: Wireless Communications. Springer Science, 2006.

[18] K. Jacobsson, H. Hjalmarsson, and N. Möller. ACK-clock dynamics in network congestion control – an inner feedback loop with implications on inelastic flow impact. In *Proc. IEEE Conf. Decision Control*, 2006.

[19] A. Tang, L. L. H. Andrew, K. Jacobsson, K. Johansson, S. H. Low and H. Hjalmarsson. Window Flow Control: Macroscopic Properties from Microscopic Factors In *Proc. IEEE INFOCOM*, 2008. [online] Available ⟨http://netlab.caltech.edu/∼lachlan/abstracts/ACanalysisTR.pdf⟩.

[20] J. Mo, R. La, V. Anantharam, and J. Walrand. Analysis and comparison of TCP Reno and TCP Vegas. In *Proc. IEEE INFOCOM*, 1999.

[21] G. E. Dullerud and F. Paganini *A Course in Robust Control Theory*. Springer, 2000.

[22] S. H. Low and D. E. Lapsley. Optimization flow control – I: Basic algorithm and convergence. *IEEE/ACM Trans. Networking*, 7(6):861–874, 1999.