Proceedings of the 17th International Symposium on Mathematical
Theory of Networks and Systems, Kyoto, Japan, July 24-28, 2006

Towards Accurate Congestion Control Models: Validation and Stability Analysis

Krister Jacobsson, Håkan Hjalmarsson and Karl Henrik Johansson
Automatic Control – School of Electrical Engineering
Royal Institute of Technology (KTH)
SE-100 44 Stockholm, Sweden
Email: {krister.jacobsson|hakan.hjalmarsson|kallej}@ee.kth.se

***Abstract*— Fluid flow models have turned out to be instrumental for analysis and synthesis of primal/dual congestion control algorithms which rely on aggregated information from a network path. In particular stability has been analyzed using such models. Departing from the theory of modeling for control, we refine the fluid flow model by augmenting the customary model of transport latencies, link price and source control with estimator dynamics and sampling properties. The impact of cross traffic and changes in network configuration is incorporated as well. As a demonstration, the modeling framework is applied to FAST TCP for which a fluid flow model is derived which through packet simulations is shown to provide accurate quantitative and qualitative information such as prediction of stability regions, behavior to cross traffic and which dynamics that influence the closed loop behavior. This model is also compared with pre-existing FAST TCP models and it is illustrated that using appropriate sampling rates and previously neglected estimation dynamics may have a large impact on the closed loop properties. We also introduce a novel link model that is validated towards packet/level data.**

I. INTRODUCTION

The tremendous complexity of the Internet makes it extremely difficult to model and analyze. However, recently significant progress in the theoretical understanding of network congestion control has been made following seminal work by Kelly and coworkers [20]. The key to success is to work at the correct level of aggregation—which is, fluid flow models with validity at longer time-scales than the round-trip time (RTT). By explicitly modeling the congestion measure signal fed back to sources, and posing the network flow control problem as an optimization problem—where the objective is to maximize the total source utility—the rate control problem can be solved in a completely decentralized manner provided that each source has a concave utility function of its own rate [20], [26].

This optimization perspective of the rate control problem has been widely adopted. It also allows for dynamical laws and the developed algorithms can be classified as: (1) primal, when the control at the source is dynamic but the link uses a static law; (2) dual, when the link uses a dynamic law but the source control is static; and (3) primal/dual, when dynamic controls are used both at the source and the links, see [23], [27], [29], for nice overviews.

This work was supported by European Commission through the project EuroNGI, by Swedish Research Council and by the project RUNES.

By appropriate choice of utility function even protocols not originally based on optimization, such as TCP Reno, can be interpreted as distributed algorithms trying to maximize the total utility [25]. Delay based protocols such as TCP Vegas [5] or FAST TCP [17] can be classified as primal/dual algorithms with queuing delay as dynamic link price.

To ensure that the system will reach and maintain a favorable equilibrium, it is important to assess the dynamical properties, such as stability and convergence, of the schemes. Instability means that the protocol is unable to sustain the equilibrium and manifests itself as severe oscillations in aggregate traffic quantities such as queue lengths.

From a control perspective, the main focus has been on stability, with numerous contributions concentrating on proving stability for more or less general configurations and scenarios as a result. Stability of some basic schemes was established already in [20], [26], but under very idealized settings, see also, e.g., [2], [22], [39]. However all results mentioned above have ignored the effect of network delay, which is critical for stability. Local stability of Reno/RED with feedback delays has been studied in [13], [28]. The stability analysis reveals that these protocols tend to become unstable when the delay increases and, more surprisingly, when the capacity increases. This has spurred an intensive research in protocols that maintain local stability also for networks with high bandwidth-delay product, see e.g., [21], [32]. Other examples of work proving local stability when taking delay into consideration are [19], [30], [36]. To be able to obtain global results but still not ignoring delay, the authors behind [8], [31], [40], use Lyapunov-Krasovskii and Lyapunov-Razumikhin functionals to establish global convergence. An alternative approach is taken in [35] where global stability of a TCP/AQM setting is analyzed via integral-quadratic constraints (IQC).

In this contribution we try to shift the focus from the stability results as such, to the *physical* modeling itself. By refining a local version of the classical primal/dual fluid flow model following Kelly's [20] framework, and interpolating our results to a window based scheme (FAST TCP and FIFO queues), we illustrate how essential a sufficiently accurate model is when making conclusions concerning stability of the true system. To authors knowledge, in practice, rigorous model validation of dynamical properties of congestion control protocols are very rare in the literature. As an example, it seems like knowledge about how a network (source/resource) system should be excited externally in the case of validation of the source- and link dynamics from measurement data, has gone unnoticed so far. Something that must be done with care if validation is performed in closed loop which is preferable for many scenarios. This issue is further elaborated on in [15] where, using the same model as here, it is illustrated how such experiments should be carried out in practice.

Also, following [20], much of the analysis has been based on fluid flow models including source controls, link price mechanisms and transport latencies. We argue that it is also important to model the dynamics of estimators typically present in the source side and the different sampling mechanisms in a network (typical congestion control mechanisms work in a number of different time scales) as they may have a large impact on the closed loop properties when considering the wide range of scenarios that can be foreseen in future networks exhibiting extreme heterogeneity in terms of bandwidth-delay products and so on. We substantiate our claims with NS-2 simulations.

The paper is organized as follows. Some preliminaries on modeling and identification are presented in Section II. In Section III a well-known model of a network is presented, refined and linearized. To substantiate our results a case study is performed in Section IV where a model of FAST TCP is developed and validated in the scope of our framework and furthermore used for stability analysis. Conclusions are given in Section V.

II. Some general remarks on modeling

In modeling and identification of complex systems, it is instrumental to consider the intended use of the model so that system properties of importance for the application are modeled with sufficient accuracy and irrelevant "details" are ignored. For example, a simulation model may have quite different properties as compared to a model suitable for control design.

Example 2.1: The system

$$G_\circ = \frac{1}{(s+1)^2(s+0.01)}$$

is modeled by

$$\hat{G} = \frac{0.5}{(s+0.04)(s+0.01)}.$$

The open loop step responses of the system and the model is given in the upper plot in Fig. 1. Clearly the model is not suitable for step-response simulation of the open loop system dynamics. However, this model allows for a

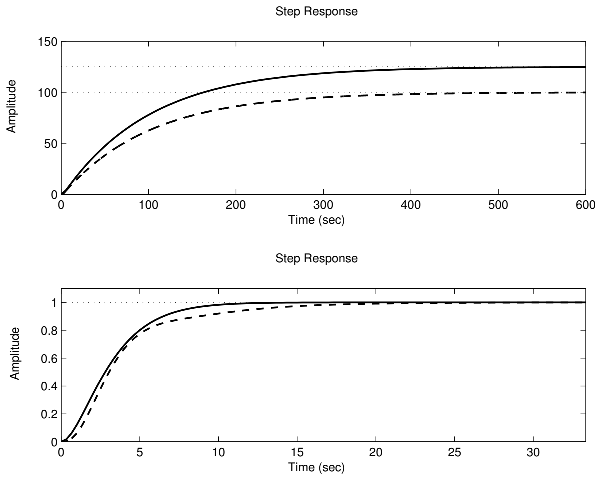

Fig. 1. Step responses in Example 2.1. Dashed line: True system. Solid line: Model. Upper: Open loop. Lower: Closed loop

reliable control design. Designing a controller $C = C(\hat{G})$ such that the designed complementary sensitivity function $T(\hat{G}) = \hat{G}C(\hat{G})/(1+\hat{G}C(\hat{G}))$ has a bandwidth of 0.4 rad/s and applying this controller to the system yields the closed loop step response in the lower plot of Fig. 1 where also the response predicted by the model is shown. We see that the agreement is quite good. ■

Over the last fifteen years or so there has been significant progress in modeling and identification when the model is to be used for control design. The present state-of-the art is summarized in [11], [12], see also [1]. In short, an accurate model is required over a frequency band covering the intended bandwidth of the closed loop. Formally this can be expressed as

$$\left\| \frac{\hat{G} - G_\circ}{\hat{G}} \; T(\hat{G}) \; S(G_\circ, \hat{G}) \right\|_\infty \ll 1, \tag{1}$$

where $C(\hat{G})$ is the controller designed on the basis of the model $\hat{G}$, where $T(\hat{G}) = \hat{G}C(\hat{G})/(1+\hat{G}C(\hat{G}))$ is the corresponding complementary sensitivity function and where $S(G_\circ, \hat{G}) = 1/(1 + G_\circ C(\hat{G}))$ is the sensitivity function for the true closed loop system. Since $T(\hat{G})$ typically is approximately unity at low frequencies and rolls off above the desired closed loop bandwidth and $S(G_\circ, \hat{G})$ exhibit the opposite behavior it follows that a good relative model fit is only required around the designed closed loop bandwidth. A prerequisite for the above discussion to be valid is that $S(G_\circ, \hat{G})$ is stable which is guaranteed if the closely related robust stability condition

$$\left\| T(\hat{G}) \; \frac{\hat{G} - G_\circ}{\hat{G}} \right\|_\infty < 1,$$

is satisfied.

Returning to Example 2.1 we see from Fig. 2 that the model fit is actually good around the desired closed loop bandwidth (0.4 rad/s) for the used model.

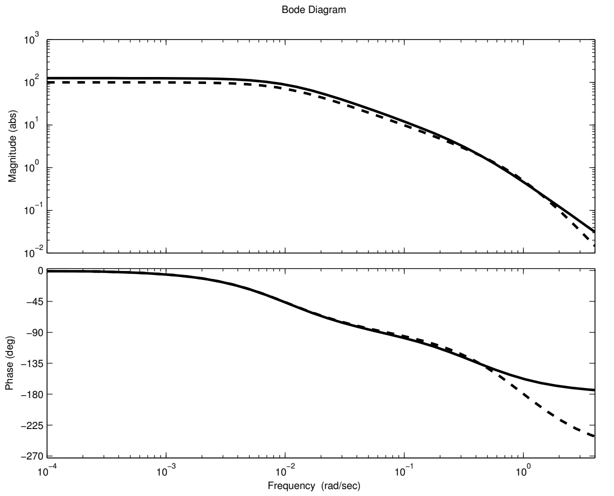

Fig. 2. Bode diagrams in Example 2.1. Dashed line: System. Upper solid line: Model. Lower solid line: Desired complementary sensitivity function.

The reason why a good model fit is not required at all frequencies is that the closed loop will be insensitive to the model accuracy at low frequencies since the controller gain typically will be large here (this is the essence of feedback control) whereas at high frequencies the controller gain will be small also resulting in low sensitivity to the modeling accuracy.

Packet based network communication systems are highly complex systems exhibiting asynchronous behavior, large number of heterogeneous nodes and non-linear behavior. The aggregation of traffic flows into fluid flows can be seen as a way of neglecting high frequency behavior well in line with the discussion above. However, as outlined above, an appropriate model must also capture the system dynamics around the desired bandwidth and in Section III we will discuss this issue in more detail.

III. Network model

In this part we present a mathematical abstraction of the network. The model and notation follows the spirit of previous work found in e.g. [14], [18], [27], [33].

A. fluid flow model

The bottlenecks in the network will be modeled as an indexed set of L resources (or links) each with an associated finite capacity c_l in packets per seconds, where $l \in \{1, \ldots, L\}$. The network is shared by a set of N persistent flows, competing about the offered capacity and indexed with $n \in \{1, \ldots, N\}$. Every flow is uniquely identified by its source-destination pair. That only bottlenecks are accounted for implies that there are always at least as many sources as links, i.e. $N \geq L$ is true by definition (adjacent links with identical capacity are viewed as one link). The impact of non-bottle neck links and short lived ('mice') traffic will be incorporated below, see Sections III-B and III-C.

To represent a certain network configuration (i.e. to associate flows with links they utilize) a *routing matrix* $R \in \mathbb{R}^{L \times N}$ is introduced. It is assumed to remain fixed and it is defined by: $R_{ln} = 1$ if link l is used by source n, and 0 otherwise. Furthermore it is implicitly assumed that R is properly posed—the configuration it represents is realizable—and that R is of full rank.

Let a packet that is sent by flow n at time t appear at link l at time $t + \tau^f_{ln}$. This *forward delay* τ^f_{ln} models the amount of time it takes to travel *from* source n *to* link l, and it accounts for total latency and queuing delays. The *backward delay* τ^b_{ln} is defined in the same manner, i.e. it is the time it takes from that a packet arrives at link l to that the corresponding acknowledgment is received at source n. The *round-trip time* associated with source n, is in this context naturally defined as $\tau_n := \tau^f_{ln} + \tau^b_{ln}$. In a buffering network round-trip time is generally time-varying since queues normally are fluctuating. We will make the simplification that time delays appearing in variable arguments, such as $x_n(t - \tau^f_{ln})$, are replaced by corresponding equilibrium values whereas delays appearing explicitly will be accounted for in full, see [27] and [24] for justification of this simplification. The validity of the model is therefore *only* in time-scales coarser than the round-trip time.

The (continuous) sending rate $x_n(t)$ in packets per second of source n at time t is related to the source congestion window $w_n(t)$ and the round-trip time as $x_n(t) := \frac{w_n(t)}{\tau_n}$ and is accurate only for the longer time scales considered. All sending rates are collected in $x(t) = [x_1(t), \ldots, x_N(t)]^T$ and the corresponding vector of forward delays to link l is denoted τ^f_l, with the convention that elements which correspond to $R_{nl} = 0$, i.e. not relevant for the network configuration, are set to zero. Similarly defined, τ^b_n is the vector of all backward delays to source n.

The aggregate flow $y_l(t)$ at link l is straightforwardly determined by the equation

$$y_l(t) = \sum_{n=1}^{N} R_{ln} x_n(t - \tau^f_{ln}) =: r_{f,l}(x(t), \tau^f_l) \qquad (2)$$

where $y_l(t)$ must not exceed the associated capacity c_l in equilibrium and collected in $y(t) = [y_1(t), \ldots, y_L(t)]^T$, so $r_f(x(t), \tau^f) = [r_{f,1}(x(t), \tau^f_1), \ldots, r_{f,L}(x(t), \tau^f_L)]^T$.

Motivated by seminal work by Kelly and coworkers [20] and Low and Lapsley [26], the congestion measure signal fed back to sources is modeled explicitly: each link has an associated congestion signal referred to as *price* $p_l(t)$ which is collected in $p(t) = [p_1(t), \ldots, p_L(t)]^T$. The *aggregate price* received at source n is defined as

$$q_n(t) = \sum_{l=1}^{L} R_{ln} p_l(t - \tau^b_{ln}) =: r_{b,n}(p(t), \tau^b_n), \qquad (3)$$

where $q_n(t)$ and $r_{b,n}(p(t), \tau^b_n)$ are individual components of the vectors $q(t)$ and $r_b(p(t), \tau^b)$ respectively.

It is assumed that source n has access to a (possibly) corrupt version $\bar{q}_n(t)$ of the aggregate price $q_n(t)$. A schematic representation of the closed loop system is given in Fig. 3. Here $\mathcal{S}$ represents the source dynamics and $\mathcal{L}$ represents the link dynamics, i.e. the pricing mechanism. The map $H : q(t) \rightarrow \tilde{q}(t)$ depends on the used price mechanism and how it is communicated to the source. H could, e.g., represent a quantizer.

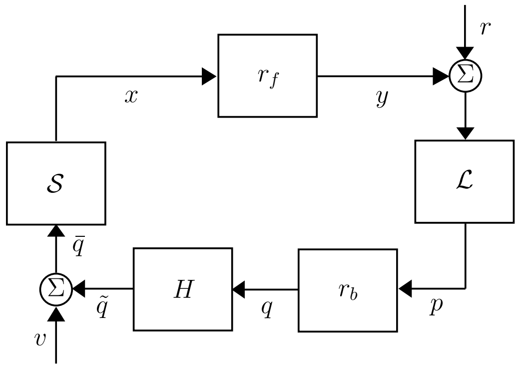

Fig. 3. Schematic representation of the delayed closed loop system of interconnected sources/resources.

B. Non-persisting sources

Only persistent sources—so called, "elephants"—are included in the fluid flow model. However, the network is also utilized by short "mice" traffic not accounted for. This additional traffic will influence individual link prices $p_l(t)$, and subsequently propagate to the persistent sources via the aggregate price $q(t)$. A natural way to model this is to treat the "mice" as noise on the link rates, or similarly as noise (filtered through link dynamics) affecting the prices. Another issue is that bottlenecks only are included in the network model. Nevertheless, also non-bottleneck links along a source's path might be exposed to "mice" cross traffic—contributing to $q_n(t)$ occasionally through the (not modeled) price mechanism. The effect of all this non-modeled traffic is accounted for in the model by the noise v on the corrupted aggregate price $\tilde{q}(t)$.

C. Changes in persistent sources

In a real setting the number of sources N is not fixed—even persistent sources connects and disconnects occasionally. From a system perspective, it is of great interest that remaining sources adapt to the new conditions suitable fast and smooth. Unfortunately, such variations in the configuration—which results in changes in the routing matrix R—are not handled by the model since R is *not* allowed to change dynamically, as stated in Section III-A. This phenomenon must instead be dealt with in an alternative way.

When a persistent source connects or disconnects it means that the links along the source's path will experience a change in the aggregate rate. This phenomenon, deriving from the configuration changes, is modeled by applying suitable signal changes in the aggregate traffic $y(t)$ represented by the signal r in Fig. 3, meanwhile keeping the network configuration fixed. Note that persistent non responsive traffic such as UDP traffic can be included in the model via r.

D. Source and link dynamics

The source dynamic should produce suitable sending rates x. As presented in [20], [26], the introduction of the concepts of aggregate rate $y_l(t)$ and aggregate price $q_n(t)$, allow specifications of dynamical source and link laws that solves the resource allocation problem in a distributed manner—that is, *without* the need for explicit communication between sources. In, so called, primal/dual control [20] each source is assumed to adjust its sending rate *dynamically* based on the aggregate price $q_n(t)$ observed. The task of the source dynamics is thus to first recover the aggregate price $q_n(t)$ from the, by cross traffic and signaling constraints, distorted signal $\bar{q}(t)$ and then to further process this signal.

Recovering $q_n(t)$ can be viewed as an estimation problem. In delay-based methods such as TCP Vegas and FAST TCP, the aggregate price is the total queuing delay along the path and the problem is to estimate this underlying delay from the round-trip time experienced by each packet. In practice the information concerning $\bar{q}(t)$ is conveyed by acknowledgments which typically arrive with a very high frequency as compared to the inherent bandwidth of the closed loop which is governed by the round-trip time or latency. From a control point of view it therefore makes sense to down sample the information. In FAST TCP and TCP Westwood this is done after the estimation as in the left block $\mathcal{S}_1$ in Fig. 4. However, it can equally well be done before any processing is performed as in the right block $\mathcal{S}_2$ in Fig. 4. One should notice that it is important to include an anti-aliasing filter (represented by AI blocks in Fig. 4) before down sampling as otherwise high-frequency components may distort the down-sampled signal. This is common practice in control systems but is often neglected in network congestion control.

The actual source control is based on the estimate $\hat{q}_n$ of the aggregate price. In a physical network the source control is realized by a dynamical window algorithm in an appropriate window based transmission control protocol such as e.g. TCP NewReno [9] or TCP Vegas [4]. The window is typically kept fixed or smoothly changed over a fixed time-interval, e.g. one round-trip time. This is modeled by the hold function in Fig. 4 which typically contribute to an increase in the overall system delay, preceded by down-sampling and possibly also an anti-aliasing operation.

The link dynamics $\mathcal{L}$ can in practice be managed by an AQM scheme such as RED [10] or REM [3]. For delay-based schemes it is simply a FIFO queue. Note that we avoid to explicitly define the entity of the price signal to keep the discussion as general as possible.

E. Linearized system

The focus of this paper is on *local* properties of the interconnected source/resource system, i.e. small perturbations

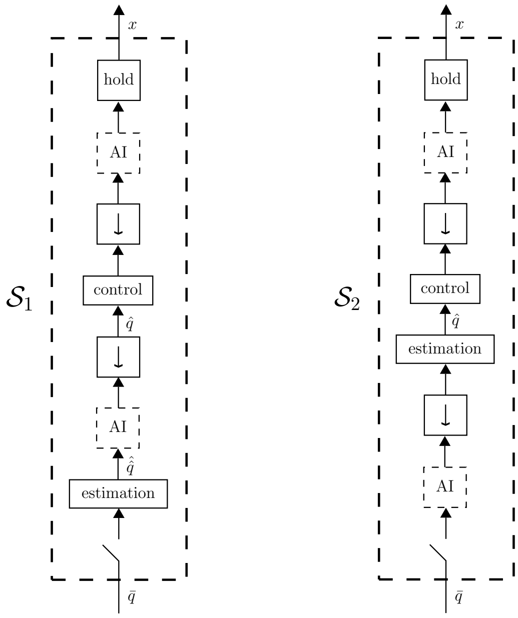

Fig. 4. A split view of the source dynamics.

$x = x^* + \delta x$, $y = y^* + \delta y$, $p = p^* + \delta p$, $q = q^* + \delta q$ around the equilibrium will be studied. Before proceeding, remark the slight abuse of notation that follows: the variables (x, y, p, q) from now on all represent *perturbations* and the elements in y and p corresponding to non-bottlenecks have been removed.

By neglecting variations in delays appearing in variable arguments, as previously stated, we regard (2) and (3) as time invariant. Hence, the Laplace transform is applicable which yields that the linearized aggregate quantities can be expressed as

$$y(s) = R_f(s)x(s) \tag{4}$$

$$q(s) = R_b^T(s)p(s) \tag{5}$$

in the frequency domain (superscript T denotes transpose). The forward delay matrix $R_f(s)$ is obtained, using the routing matrix R, by replacing unit elements by the appropriate Laplace domain forward delay $e^{-\tau_{ln}^f s}$. The backward delay matrix $R_b(s)$ is obtained similarly, using backward delays $e^{-\tau_{ln}^b s}$ instead.

Denote the individual linearized frequency domain source dynamics $K_n(s)$ and link dynamics $F_l(s)$. Define $\mathcal{K}(s) := \text{diag}(K_n(s))$ and $\mathcal{F}(s) := \text{diag}(F_l(s))$ which yields that

$$x(s) = \mathcal{K}(s)q(s) \tag{6}$$

$$p(s) = \mathcal{F}(s)y(s) \tag{7}$$

in vector form. Now (4), (5), (6) and (7) together defines the linear closed loop system, representing the interconnected source/resource system at equilibrium. It is sometimes implicitly understood—by the argument—if it is the time domain or corresponding frequency domain expression that is considered.

IV. Case study: Time-scale modeling of FAST TCP

In this section the time based TCP sibling FAST TCP recently proposed [17] is modeled and analyzed using the framework of Section III. By rigorous modeling of relevant dynamics and the different time scales that algorithms work in, a model is obtained that when validated against packet-level simulations is shown to be accurate. Analysis using this model then yields stability constraints on protocol parameters. The obtained bounds are shown to accurately reflect the behavior in packet-level simulations.

A. Protocol rationale

Designed for large distance high speed data transfers, FAST TCP adopts the delay-based approach of TCP Vegas [5] to be able to attain a 'stable' equilibrium. This is in contrast to the protocols using a loss-based approach which necessarily oscillate around the equilibrium rate. The rate (window) control estimates the end-to-end queuing delay and stabilizes its rate such that a targeted number of packets are buffered in the network—subsequently, full utilization of network resources is guaranteed. Additionally, a salient feature of FAST TCP is an equilibrium rate independent of network latency, this theoretically implies a completely α–fair protocol. For FAST TCP this is expressed by the constraint

$$x_n\, q_n = \alpha_n, \quad n = 1, \ldots, N \tag{8}$$

where α_n is a parameter in the algorithm, in equilibrium.

B. The FAST TCP algorithm

Below, the basic features of the FAST TCP algorithm are briefly described. We will drop the subscript n for ease of notation.

1) Queue size estimation: The aggregated price in the algorithm is the queuing delay. This quantity is obtained by estimating the latency d and then using round-trip time samples. The latency d is estimated by simply logging the minimum round-trip time observed, and obeying slow dynamics it seems reasonable to approximate it as a *perfectly known* static variable. Noting that this approximation does not affect the dynamical properties of the model, the latency estimation procedure in FAST TCP is not further considered in this contribution. Notice however, that any bias in this estimate will influence the equilibrium point and hence the fairness properties in practice.

With the kth round-trip time sample denoted $\tau(k)$, the queue time sample is $\bar{q}(k) = \tau(k) - d$. This is a noisy measurement of the actual price $q(t)$. Consequently, in FAST TCP $\bar{q}(k)$ is low-pass filtered according to

$$\hat{q}(k+1) = (1 - \eta(k))\,\hat{q}(k) + \eta(k)\bar{q}(k) \tag{9}$$

where

$$\eta(k) = \min\left\{\frac{3}{\tilde{w}(k)}, \frac{1}{4}\right\} \tag{10}$$

and $\tilde{w}(k)$ is defined below.

2) Window control: The sending rate of FAST TCP is implicitly adjusted via the congestion window w. Instead of considering sending rates as variables of interest as in Fig. 3, we will therefore use window sizes in the sequel. In each sender the window is adjusted according to

$$w(k+1) = (1-\gamma)w(k) + \gamma \frac{d}{d+\hat{q}(k)} w(k) + \gamma\alpha, \quad (11)$$

where $\alpha \in \mathbb{Z}^+$ and $\gamma \in (0,1]$ are protocol parameters and d is the network latency. Note that we will distinguish between the congestion window w, which is an intermediate variable, and the 'sending' congestion window $\tilde{w}$, which reflects the actual number of packets in flight. The congestion window is updated at a constant time interval, T_w. This means that down-sampling occur in the map $\hat{q} \mapsto w$, when the congestion window is large enough. This is performed without any anti-alias filtering of $\hat{q}$ first.

The sending window is periodically updated as follows: It is smoothly updated during one round-trip time and kept fixed the following round-trip time. Also this step is performed without any anti-alias filtering. The procedure is similar to the $\mathcal{S}_1$ block in Fig. 4 without AI blocks.

3) Link dynamics: FAST TCP is considered to operate in a future Internet scaled up in capacity and size with routers applying FIFO queuing policy. Mainly, in the literature there exist two different approaches in the modeling of a queue: 1) treating the queue as a static function, and 2) taking the dynamical properties over all frequencies into consideration and hence model its integrating effect. Examples of the first approach can be found in, for example, [20] and [37], and the latter in [27] and [33]. We remark that when using the window size as explicit state of the protocol the dynamics of the link buffer is different from the case of working directly with rates.

4) Algorithm summary: FAST TCP fits Fig. 3 with the source dynamics described by the $\mathcal{S}_1$ block found in Fig. 4 ignoring the anti-alias filters.

C. Model

In this section we will derive a linearized model for a single FAST TCP source sending over a single bottleneck link. A fundamental assumption in the modeling framework presented in Section III is that time delays are time invariant. However, as FAST TCP operates in a FIFO (first-in-first-out) buffering network this is not the case since queuing delay varies with time, with nested arguments in the variables as a consequence. We will follow the convention introduced and motivated in Section III-A that time delays appearing in variable arguments are replaced by equilibrium values while delays appearing explicitly will be accounted for. Addressing high capacity high latency networks, queuing delay is only a fraction of round-trip time which should benefit the considered approximation. Still, packet-level simulations are provided in Section IV-E to validate achieved results.

It turns out to simplify the analysis by working in continuous time so continuous time equivalents will be derived for as many blocks as possible of the system. Equilibrium values are denoted with subscript 0 in the sequel.

1) Link dynamics: In our simple one link, one source scenario, we will model the link as a continuous integrator, driven by the incoming rate at the link, but also include a direct term to model the immediate effect a change in the congestion window has on a saturated link,

$$\dot{p}(t) = \begin{cases} \frac{\dot{\tilde{w}}(t-\tau^f) + \frac{\tilde{w}(t-\tau^f)}{d+p(t)} - c}{c}, & \text{if } p(t) > 0 \text{ or } \dot{\tilde{w}}(t-\tau^f) \\ & + \frac{\tilde{w}_n(t-\tau^f)}{d+p(t)} > c, \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

This novel link model is developed and thoroughly validated in [16], it is obtained by a careful study of the source/link system at packet level and captures the so called self-clocking effect in window based schemes which rate based models fails to do [38]. In Section IV-E we demonstrate that the linearization of this model is accurate around the equilibrium. Below we comment on the relationship to previous work.

Linearization of (12) around the equilibrium $p_0 = q_0 = \alpha/x_0$ yields

$$\dot{p}(t) = -\frac{x_0^2}{c(\alpha + x_0 d)} p(t) + \frac{x_0}{c(\alpha + x_0 d)} \tilde{w}(t-\tau^f) + \frac{1}{c}\dot{\tilde{w}}(t-\tau^f). \quad (13)$$

Studying (13) in the frequency domain one realizes that when a source utilizes the link alone, i.e. $x_0 = c$, the map from $\tilde{w}$ to p is a pure time-delay as used in e.g. [37] and [6]. However, when *inelastic* cross traffic (such as UDP) *is* present, $x_0 < c$, a filtering effect kicks-in and the relation is not static anymore; this is also evident from the snapshot of the validation experiments presented in Section IV-E.1.

2) Estimator dynamics: Assuming $w_0 > 12$ the linearized version of the queue estimator (9), (10) is given by

$$\hat{q}(k+1) = \left(1 - \frac{3}{w_0}\right)\hat{q}(k) + \frac{3}{w_0}\bar{q}(k). \quad (14)$$

The filter in (9) is updated at every received acknowledgment. This means that the filter receives a congestion window amount of samples every round-trip time, hence the mean sampling interval becomes $T_q \approx \frac{d+q_0}{w_0}$ in equilibrium. Ignoring that the true samples are non-uniformly distributed in time, we approximate the discrete filter in the continuous domain with

$$\dot{\hat{q}}(t) = -\frac{3}{T_q(\alpha + x_0 d)}\hat{q}(t) + \frac{3}{T_q(\alpha + x_0 d)}\bar{q}(t). \quad (15)$$

where we have used that the equilibrium window is $w_0 = (d+q_0)x_0$ and (8). This approximation is accurate enough for time-scales about a magnitude greater that the ACK-inter-arrival time. Again, since high capacity links are considered this should be sufficient for the scope of this paper.

3) Window control: We will assume that the queue estimator produces unbiased estimates. Hence we can linearize (11) around the equilibrium queuing delay, which we denote by q_0. Using that the equilibrium window is $w_0 = (d+q_0)x_0$ and that $\alpha = x_0\, q_0$, the linearized window dynamics is given by

$$w(k+1) = \left(1 - \gamma \frac{q_0}{d+q_0}\right) w(k) - \gamma \frac{\alpha d}{q_0(d+q_0)} \hat{q}(k) \tag{16}$$

Considering only the case when bandwidth is high enough to guarantee that the sampling rate T_w is greater than the ACK-inter-arrival time, inverse sampling of (16) together with the approximation $\ln\left(1-\gamma\frac{q_0}{d+q_0}\right) \approx -\gamma\frac{q_0}{d+q_0}$, valid for sufficiently small $\gamma\frac{q_0}{d+q_0} > 0$ (which holds for the high bandwidth high and latency case), and (8) yields the following continuous time equivalent

$$\dot{w}(t) = -\frac{1}{T_w}\gamma\frac{\alpha}{\alpha+x_0 d}w(t) - \frac{1}{T_w}\gamma\frac{x_0^2 d}{\alpha+x_0 d}\hat{q}(t) \tag{17}$$

to (16). In (17), the alias effect that occurs in the down sampling of $\hat{q}$ is not accounted for. We believe, however, that the model captures the fundamental behavior of the congestion window update, including the separated time-scales that is present in the system as described later on.

4) Sending window update: The sending window, which represents the actual packets kept in flight, is as mentioned updated every two round-trip times. To capture this effect in the model, a sampler with the sampling time denoted by $T_{\tilde{w}}$ in conjunction with a zero-order-hold (ZOH) describes the signal map $w \mapsto \tilde{w}$.

5) Model summary: The resulting mathematical abstraction of a single FAST TCP source sending over a single bottleneck is represented by the block diagram of the closed-loop system shown in Fig. 5, where the filters $G_{\hat{q}\bar{q}}(s)$, $G_{w\hat{q}}(s)$ and $G_{p\tilde{w}}(s)$ are the frequency domain (Laplace) equivalents to (15), (17) and (13) (excluding delay which is accounted for separately), respectively. For future reference we now introduce $G = e^{-s\tau_0}G_{p\tilde{w}}(s)G_{w\hat{q}}(s)G_{\hat{q}\bar{q}}(s)$ where the equilibrium round-trip-time $\tau_0 = d + q_0 = d + \alpha/x_0$. The system in Fig. 5

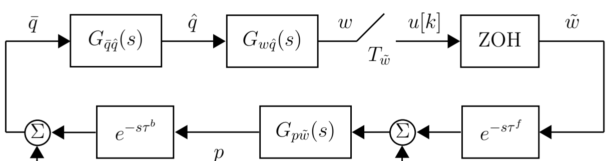

Fig. 5. Block diagram of a single FAST TCP source sending over a single bottleneck link.

cannot directly be analyzed in continuous time using the Laplace transform since it is not time invariant due to the zero-order-hold function. However, assuming that aliasing that occur is not severe at the same time as $|G(j\omega)|$ is strictly decreasing with increasing frequency we can ignore overlapping components in the Poisson summation formula [34].

This simplification effectively amounts to replacing the zero-order-hold function by a continuous time delay of $T_{\tilde{w}}$. With this approximation, we can thus work entirely in continuous time and the resulting loop transfer function (assuming negative feed-back) is given by

$$\tilde{L}(s) := -G(s)\, e^{-sT_{\tilde{w}}} = \frac{\frac{x_0 d}{\alpha} e^{-s\left(T_{\tilde{w}}+d+\frac{\alpha}{x_0}\right)} \left(\frac{\alpha+x_0 d}{x_0}s+1\right)}{\left(\frac{T_w(\alpha+x_0 d)}{\gamma\alpha}s+1\right)\left(\frac{T_q(\alpha+x_0 d)}{3}s+1\right)\left(\frac{c(\alpha+x_0 d)}{x_0^2}s+1\right)} \tag{18}$$

with $T_q = 1/x_0$.

Once again we emphasize that due to the different approximations made, packet-level simulations are needed for validation of results and model. This will be provided in Section IV-E.

Note that values of capacity c, latency d, and forward and backward delay τ^f and τ^b are determined by the particular network configuration, while α (the number of packets buffered in the network), γ, T_q (estimation algorithm sampling interval), T_w (window update interval) and $T_{\tilde{w}} > T_w$ (sending window update interval) are considered as design parameters.

D. Stability analysis

In this section, making use of the derived model of the FAST TCP/single bottleneck model, a stability analysis is performed. The accuracy of the predictions (the model) is substantiated in Section IV-E.

The simplicity of the loop gain (18) makes it tractable for stability analysis by way of the Nyquist criterion. Since the system is open loop stable it is enough to study whether the closed contour of the map (18) encircles -1 or not when input argument is traversing the imaginary axis.

Consider the case with fixed finite $\alpha \in \mathbb{Z}^+$ and $\gamma \in (0, 1]$ as in FAST TCP. Focusing on the high capacity case and with a single user so that $c = x_0$, it is realized from (18) that the pole deriving from the link dynamics in cancelled by the corresponding zero and that for sufficiently large capacity c (and hence sending rate x_0) the pole corresponding to the queue estimation approaches $-\frac{3}{d}$, while the window control dynamics is increasingly slower with increasing c. This implies that for sufficiently high capacity

$$L(j\omega) \approx \frac{\frac{cd}{\alpha}e^{-j\omega\left(T_{\tilde{w}}+\tau^f+\tau^b\right)}}{\left(\frac{T_w(\alpha+cd)}{\gamma\alpha}j\omega+1\right)} \tag{19}$$

in the frequency region of interest. Since the magnitude of the transfer function (18) is strictly decreasing with increasing frequency, the Nyquist criterion simplifies so that a necessary and sufficient condition for stability is that the phase-crossover frequency ω_p is greater than the gain-crossover frequency ω_c. The phase-crossover frequency is obtained by solving for ω_p in

$$-\left(T_{\tilde{w}}+d+\frac{\alpha}{c}\right)\omega_p-\arctan\left(\frac{T_w cd}{\gamma\alpha}\omega_p\right)=-\pi \quad (20)$$

which for sufficiently large c yields $\omega_p \approx \frac{\pi}{2(T_{\tilde{w}}+d)}$, and, similarly, solving for the gain-crossover frequency from

$$\frac{\frac{cd}{\alpha}}{\sqrt{\left(\frac{T_w cd}{\gamma\alpha}\omega_c\right)^2+1}}=1 \quad (21)$$

gives $\omega_c \approx \frac{\gamma}{T_w}$. Subsequently, it is concluded that the system is predicted to be locally stable if, as previously stated, $\omega_p > \omega_c$, which gives the following stability condition on the tunable protocol parameter γ

$$\gamma < \frac{\pi T_w}{2(T_{\tilde{w}}+d)}. \quad (22)$$

The condition (22) indicates that there is a potential danger in making the window update time proportional to the acknowledgment inter-arrival time, that is $T_w \propto \frac{1}{c}$, since the stability region then is continuously decreasing with larger capacities.

The condition (22) also shows that the sampling rates influence dynamics and ignoring them in the model may yield significantly different results. For example, (22) indicates that freezing the sending window decreases the stability region.

Remark 4.1: It can be noted that a single source single bottleneck scenario leads to that (13) reduces to a queue model consisting of a pure delay which is the model used in [37]. Furthermore, not taking estimator dynamics into consideration and using $T_\omega = 1$ and $T_{\tilde{\omega}} = 0$ in (17) above, i.e. ignoring sampling effects, the window model (17) becomes the continuous time equivalent to the window model used in [37]. Thus, modulo the difference between continuous and discrete time results, the stability bound (22) applies to the scenario in [37] as well.

E. Model validation

In this section we will validate our theoretical findings with packet-level simulations of a single FAST TCP source sending over a bottleneck link. A slightly modified version of the NS-2 implementation of FAST TCP [7] is used to generate reference data. We point out that in the original NS-2 implementation the window update mechanism is such that $w(k-1)$ appears instead of the right most $w(k)$ in (11). This yields a more low-pass behavior of the window control mechanism around the equilibrium and increases the stability region. Here, as originally described in [17], we use a window update according to (11) in all simulations. We remark that x_0 in (18) is taken as the equilibrium after the occurrence of a change in conditions.

1) Identification: After deriving the model a natural *first* step is to establish its accuracy. To verify the dynamical model of the protocol, i.e. (17) in concatenation with (15), and the link model (13) individually, we use the discrete event simulator NS-2 to generate experimental packet-level data. Notice, that the external excitation must be applied with care since the system is operating in closed loop, see [15] for a further discussion.

Two different scenarios are considered: (1) validation of the window control by excitation through r and, (2) validation of the link model via excitation in v, see Fig 5. The outcomes of the experiments are described in Example 4.1–4.3 respectively.

Example 4.1: The experimental setting is a single FAST TCP source sending over a single bottleneck link (applying FIFO queuing policy) with capacity $c = 12020$ packets/s. The round-trip latency $d = 100$ ms and the network configuration is such that $\tau_f = 0$ ms and $\tau_b = 100$ ms. The FAST TCP parameters are set to $\alpha = 400$, $\gamma = 0.1$ and $T_{\tilde{w}} = T_w = 0.01$.

At time $t = 10$ a negative step of 1202 packets/s is taken in the available bandwidth (realized by applying appropriate UDP traffic), i.e. a step r of magnitude $c/10$ in Fig. 5. Queue size and window size samples are collected. The queue size data was filtered through the source control

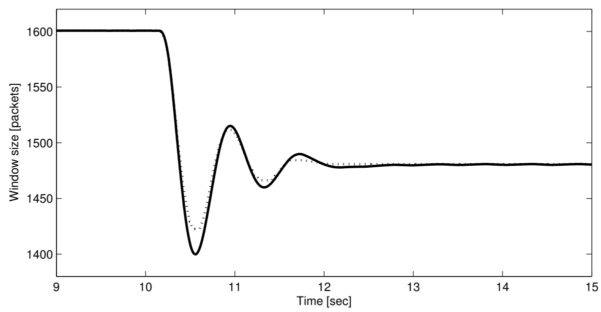

Fig. 6. Identification example (step in r). Dotted line: true dynamics (NS-2). Solid line: protocol dynamics (simulated in Matlab).

dynamics, i.e. $G_{w\hat{q}}(s)G_{\hat{q}\bar{q}}(s)e^{-s\tau^b}$. The window size signals are shown in Fig. 6, the solid line is the window size simulated in Matlab and the dotted line the 'true' window size from the NS-2 simulation. It is observed that the model fit is good. We remark that x_0 is taken as the equilibrium after the occurrence of a change in conditions. ■

Example 4.2: The experimental setting is a single FAST TCP source sending over a single bottleneck link (applying FIFO queuing policy) with capacity $c = 6010$ packets/s. The round-trip latency $d = 100$ ms and the network configuration is such that $\tau_f = 60$ ms and $\tau_b = 40$ ms. The FAST TCP parameters are set to $\alpha = 400$, $\gamma = 0.2$ and $T_{\tilde{w}} = T_w = 0.01$.

The external perturbation is similar as in Example 4.1 except that the system in Fig. 5 is excited via a step change in v instead of through r. The window size data was filtered through the queue dynamics, i.e. $G_{p\tilde{w}}(s)e^{-s\tau^f} = \frac{e^{-s\tau^f}}{c}$. The resulting signals are shown in Fig. 7. The solid line is the queue size simulated in Matlab and the dotted line the 'true' queue size from the NS-2 simulation. It is observed that the map between the effective congestion window size $\tilde{w}$ and the queue size p is indeed a pure delay as predicted by the analysis in Section IV-C.1. We remark

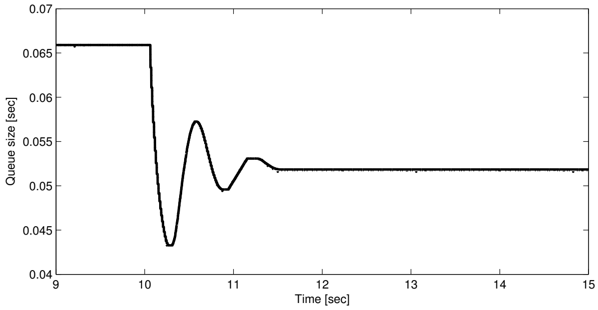

Fig. 7. Identification example (step in v). Dotted line: true dynamics (NS-2), Solid line: queue dynamics (simulated in Matlab).

that in practice the excitation is achieved through direct manipulation of the internal protocol variable $\hat{q}$—actually by applying a step with magnitude 0.025. ■

Example 4.3: The experimental setting is a single FAST TCP source sending over a single bottleneck link (applying FIFO queuing policy) with capacity $c = 60096$ packets/s which is also utilized by UDP traffic taking 80% of the total bandwidth, i.e. $x_0 = 0.20c = 12019$ packets/s. The round-trip latency $d = 150$ ms and the network configuration is such that $\tau_f = 45$ ms and $\tau_b = 105$ ms. The FAST TCP parameters are set to $\alpha = 400$, $\gamma = 0.8$ and $T_{\tilde{w}} = T_w = 0.01$.

The external perturbation is similar as in Example 4.2 but with magnitude -0.015. The window size data was

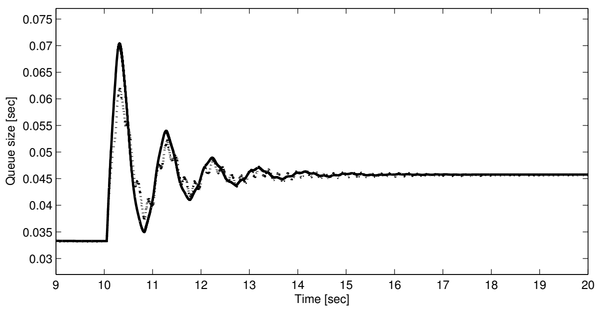

Fig. 8. Identification example (step in v). Dotted line: true dynamics (NS-2), Solid line: queue dynamics (simulated in Matlab).

filtered through the queue dynamics, i.e. $G_{p\tilde{w}}(s)e^{-s\tau^f}$. The resulting signals are shown in Fig. 8. The solid line is the queue size simulated in Matlab and the dotted line the 'true' queue size from the NS-2 simulation. It is concluded that (13) captures the behavior well. The observed high frequency ringing in the NS-2 data is due to jitter at packet level which is neglected in the fluid flow approximation and hence not captured by the model. ■

From the three previous examples together with extensive validation experiments in the same spirit not presented here, it is concluded that the model of the protocol dynamics is accurate.

2) A cross traffic scenario: In this part the local dynamical properties of the model are tested in the spirit of the framework described in Section III and [14]. In the NS-2 simulations, when the system is in equilibrium a UDP source starts to send at a constant rate over the bottleneck link. In the model in Fig. 5 this is modeled as a step in r of suitable size. In Fig. 9 and Fig. 10 the bottleneck queue length outputs from two different scenarios are provided.

A step change of magnitude a fifth of the capacity is

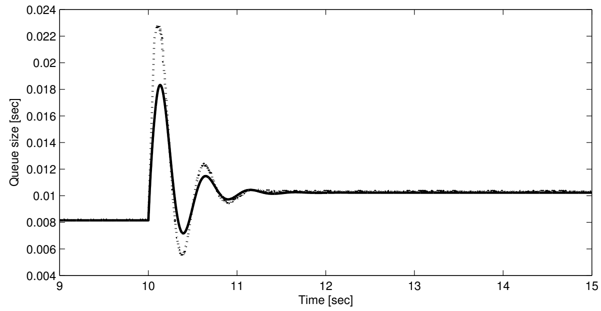

Fig. 9. Queue size plot of a single FAST TCP source sending over a single bottleneck link. At time $t = 10$s a UDP source starts to send over the bottleneck with rate $c/5$. Dotted line: NS-2 simulation. Solid line: Fluid flow model simulation (in Matlab).

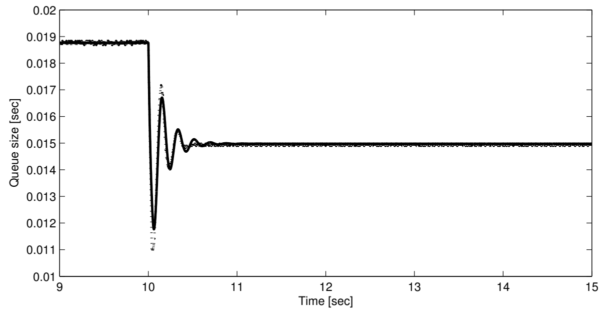

Fig. 10. Same type of plot as in Fig. 9 but with $c = 13221$ packets/s, $d = 20$ ms.

applied at time $t = 10$ s. Protocol parameters are set to $\alpha = 200$, $\gamma = 0.12$ in the first case and to to $\alpha = 200$, $\gamma = 0.5$ in the second case; also, $T_{\tilde{w}} = T_w = 0.01$ in both cases. In Fig. 9 capacity is set to $c = 24038$ packets/s and delay is configured to $d = 60$ ms, while in Fig. 10 capacity is $c = 13221$ packets/s and $d = 20$ ms. It is observed that the linear model seems to capture frequency, amplitude and decay with high precision. This is also in line with additional experiments not reported here due to space limitations.

3) Stability: We will now examine the validity of stability condition (22) derived in Section IV-D.

Example 4.4: Consider a single FAST TCP source sending over a single bottleneck link of capacity $c = 24038$ packets/s. The round-trip latency is $d = 100$ ms and the source tries to keep $\alpha = 100$ packets buffered in the network. The congestion window update time is set to $T_w = 0.01 \gg T_q$ (this is the default value in FAST TCP NS-2 version [7]) and the 'sending' congestion window is updated together with the congestion window, that is $T_{\tilde{w}} = T_w$. From (22) it is predicted that the system is stable for $\gamma < 0.14$. The system is simulated twice in NS-2 for 25 seconds, with $\gamma = 0.15$ and $\gamma = 0.13$ respectively. Fig. 11 shows the bottleneck queue size for the two NS-2 simulations. It is observed that $\gamma = 0.15$ yields an unstable

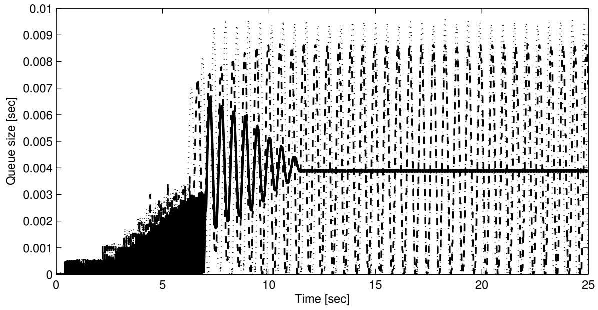

Fig. 11. Queue size plot of a single FAST TCP source sending over a single bottleneck link. Dotted line: $\gamma = 0.15$, $T_{\tilde{w}} = T_w = 0.01$. Solid line: $\gamma = 0.13$, $T_{\tilde{w}} = T_w = 0.01$. Dashed line: $\gamma = 0.13$, $T_{\tilde{w}} = 2T_w = 0.02$.

system, while for $\gamma = 0.13$ the system becomes stable. This indicates that the prediction of the stability region provided by (22), which is based on the fluid flow model, is remarkably accurate. We point out that similar results are obtained for various configurations for sufficiently high capacities.

The dashed line in Fig. 11 corresponds to a simulation similar to the $\gamma = 0.13$ case above but where the 'sending' congestion window $\tilde{w}$ is updated at half the rate of the congestion window w, i.e. $T_{\tilde{w}} = 2T_w = 0.01$. As predicted by (22) the system becomes unstable. ■

4) Impact of estimation dynamics: In the literature, estimation dynamics is often neglected, c.f. [17], leading to that the only remaining dynamics are the window control and time-delays (queue dynamics are static in this case). However, considering the single source single bottleneck configuration ($x_0 = c$) (18) indicates that estimator dynamics can not be neglected whenever $\frac{\gamma\alpha}{T_w}$ is of the same magnitude as capacity c. In fact there exist scenarios where estimator dynamics are crucial whereas window dynamics can be neglected. This is illustrated in Example 4.5.

Example 4.5: A single FAST TCP source with $\alpha = 100$, $T_{\tilde{w}} = T_w = 0.01$ and the recommended value $\gamma = 0.5$ is sending over a single bottleneck link with capacity $c = 601$ packets/s and round-trip latency $d = 100$ ms.

As $\frac{\gamma\alpha}{T_w} = 5000 > c$, the model implies that since window dynamics (17) is clearly faster than queue estimator dynamics (15), increasing γ (and making window dynamics even faster) should not influence the system dynamics to any appreciable extent.

The plot in Fig. 12 shows the queue sizes from two NS-2 simulations of this system with $\gamma = 0.5$ (dotted line) and $\gamma = 1$ (solid line) respectively. It can be observed that it is hard to distinguish between the two curves which shows that the model prediction for this behavior is accurate. ■

The accuracy of the model and the importance of taking queue estimation dynamics into account is further demonstrated in Example 4.6.

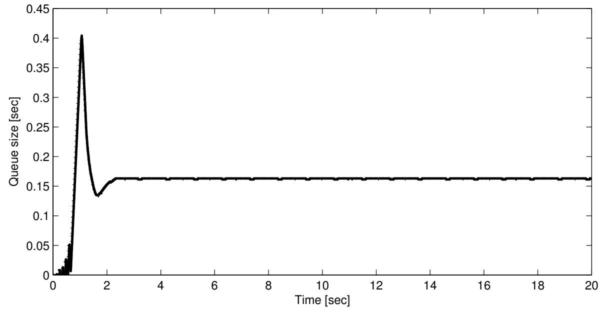

Fig. 12. Queue size plot of a single FAST TCP source sending over a single bottleneck link. Dotted line: $\gamma = 0.5$. Solid line: $\gamma = 1$.

Example 4.6: In this example the task is to predict for what delay a single FAST TCP source ($\alpha = 500$, $\gamma = 0.5$, $T_{\tilde{w}} = T_w = 0.01$) sending over a single bottleneck link with capacity $c = 2404$ packets/s is stable. This is done by checking for what d the equality $\omega_p = \omega_c$ is fulfilled, where ω_p and ω_c are obtained by solving (20) and (21), respectively. It is easily shown that $\omega_p = \omega_c$ when $d \approx 266$ ms. Performing the same operation but ignoring dynamics deriving from queue estimation in (18) yields a delay limit at $d \approx 216$ ms. The bottleneck queue of two packet-level simulations in NS-2 are displayed in Fig.13. From the figure it is concluded that the system becomes

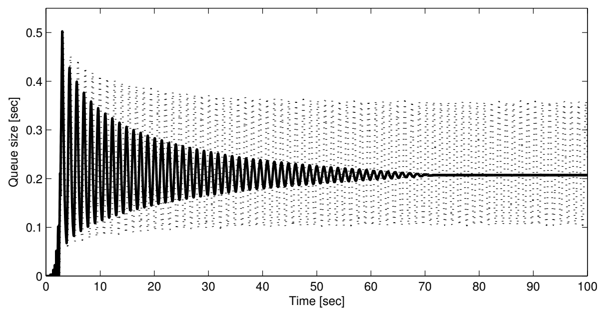

Fig. 13. Queue size plot of a single FAST TCP source sending over a single bottleneck link. Dotted line: $d = 270$ ms. Solid line: $d = 260$ ms.

unstable for $d > d^*$ where d^* is somewhere between 260 ms and 270 ms. This was also accurately predicted by the model including all present dynamics. ■

V. Conclusions

Having the congestion control application in mind, and drawing heavily from existing experience of fluid flow modeling, we have in this contribution presented a fluid flow model for packet-based communication networks. By a careful study of the signal paths we have incorporated the impact of short lived cross traffic as well as the impact of changes in the network configuration as external signals at suitable points in the block-diagram in Fig. 3. Also, dynamics has been incorporated which extends the validity of the model to a wide range of operating conditions. To this end we have included estimator dynamics, typically present in delay-based schemes, and sampling effects.

We have applied our modeling framework to FAST TCP and through packet simulations it has been shown that the resulting model provide accurate quantitative and qualitative information, such as prediction of stability regions, behavior to cross traffic and which dynamics that influence the closed loop behavior.

ACKNOWLEDGMENT

The authors would like to thank Lachlan Andrew, Tony Cui, Mikael Johansson and Niels Möller.

REFERENCES

[1] P. Albertos and A. Sala (Eds.), *Iterative Identification and Control*. Springer Verlag, 2002.

[2] E. Altman, T. Basar, and R. Srikant, "Robust rate control for ABR sources," in *IEEE Infocom*, San Francisco, CA, 1998, pp. 166–173.

[3] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, 2001.

[4] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," in *Proceedings of the conference on Communications architectures, protocols and applications*. ACM Press, 1994, pp. 24–35.

[5] L. S. Brakmo and L. L. Peterson, "TCP Vegas: end-to-end congestion avoidance on a global Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, 1995.

[6] J. Y. Choi, K. Koo, J. S. Lee, and S. Low, "Global stability of FAST TCP in single-link single-source network," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 1837–1841.

[7] T. Cui and L. Andrew, "FAST TCP simulator module for ns-2, version 1.1." [Online]. Available: <http://www.cubinlab.ee.mu.oz.au/ns2fasttcp>

[8] S. Deb and R. Srikant, "Global stability of congestion controllers for the internet," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 1055–1060, June 2003.

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

[10] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[11] H. Hjalmarsson, "From experiments to closed loop control," in *Proc. IFAC Symposium SYSID 2003*, Rotterdam, The Netherlands, 2003, plenary address.

[12] ——, "From experiment design to closed loop control," *Automatica*, vol. 41, no. 3, pp. 393–438, March 2005.

[13] C. Hollot, V. Mistra, D. Towsley, and W. B. Gong, "A control theoretic analysis of RED," in *Proceedings of IEEE Infocom 2001*, 2001.

[14] K. Jacobsson and H. Hjalmarsson, "Local analysis of structural limitations of network congestion control," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005.

[15] ——, "Closed loop aspects of fluid flow model identification in congestion control," in *IFAC Symposium on System Identification*. Newcastle, Australia: IFAC, March 2006.

[16] K. Jacobsson, H. Hjalmarsson, and N. Möller, "ACK-clock dynamics in network congestion control – an inner feedback loop with implications on inelastic flow impact," February, submitted.

[17] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: motivation, architecture, algorithms, performance," in *Proceedings of IEEE Infocom*. IEEE, March 2004.

[18] ——, "FAST TCP: motivation, architecture, algorithms, performance," Caltech, Tech. Rep. CaltechCSTR:2003.010, August 2004.

[19] R. Johari and D. Tan, "End-to-end congestion control for the Internet: delays and instability," *IEEE/ACM Transactions on Networking*, vol. 6, no. 9, pp. 818–832, 2001.

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

[21] T. Kelly, "Scalable TCP: improving performance in highspeed wide area networks," in *First International Workshop of Protocols for Fast Long-Distance Networks*, 2003.

[22] S. Kunniyur and R. Srikant, "A time-scale decomposition approach to adaptive ECN marking," *IEEE Transactions on Automatic Control*, vol. 47, no. 6, pp. 884–894, 2002.

[23] S. Liu, T. Basar, and R. Srikant, "Controlling the Internet: A survey and some new results," in *Proceedings ot the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii USA, 2003, pp. 3048–3057.

[24] ——, "Pitfalls in the fluid modeling of RTT variations in window-based congestion control," in *Proceedings of IEEE Infocom 2005*. IEEE, March 2005.

[25] S. H. Low, "A duality model of TCP and queue management algorithms," in *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, 2000.

[26] S. H. Low and D. E. Lapsley, "Optimization flow control – I: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[27] S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *Control Systems Magazine*, vol. 22, no. 1, pp. 28–43, February 2002.

[28] S. H. Low, F. Paganini, J. Wang, S. A. Adlakha, and J. C. Doyle, "Dynamics of TCP/RED and a scalable control," in *IEEE Infocom*, vol. 1, New York, June 2002, pp. 239–248.

[29] S. H. Low and R. Srikant, "A mathematical framework for designing a low-loss, low-delay Internet," *Networks and Spatial Economics*, January-February 2003, special issue on "Crossovers between Transportation Planning and Tellecommunications".

[30] L. Massoulie, "Stability of distributed congestion control with heterogeneous feedback delays," Microsoft Research, Cambridge, UK, Technical Report, 2000.

[31] F. Mazenc and S. I. Niculescu, "Remarks on the stability of a class of TCP-like congestion control models," in *IEEE Proceedings of Conference on Decission and Control*. IEEE, December 2003, pp. 5591–5594.

[32] F. Paganini, J. C. Doyle, and S. H. Low, "Scalable laws for stable network control," in *IEEE CDC*, vol. 1, Orlando, FL, December 2001, pp. 185–190.

[33] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low, "Congestion control for high performance, stability, and fairness in general networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 43–56, 2005.

[34] A. Papoulis, *Signal Analysis*. New York: McGraw Hill, 1977.

[35] M. Peet and S. Lall, "On global stability of Internet congestion control," in *IEEE Proceedings of Conference on Decission and Control*. IEEE, December 2004, pp. 1035–1041.

[36] G. Vinnicombe, "On the stability of networks operating TCP-like congestion control," in *IFAC World Congress*, Barcelona, Spain, July 2002.

[37] J. Wang, A. Tang, and S. H. Low, "Local stability of FAST TCP," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 1. IEEE, December 2004, pp. 1023–1028.

[38] J. Wang, D. X. Wei, and S. H. Low, "Modelling and stability of FAST TCP," in *Proceedings of IEEE Infocom 2005*. IEEE, March 2005.

[39] J. T. Wen and M. Arcak, "A unifying passivity framework for network flow control," *IEEE Transactions on Automatic Control*, vol. 49, no. 2, pp. 162–174, 2004.

[40] L. Ying, G. E. Dullerud, and R. Srikant, "Global stability of internet congestion controllers with heterogeneous delays," in *Proceedings of the American Control Conference 2004*, July 2004.