

CLOSED LOOP ASPECTS OF FLUID FLOW MODEL IDENTIFICATION IN CONGESTION CONTROL¹

Krister Jacobsson* Håkan Hjalmarsson*

{krister.jacobsson|hakan.hjalmarsson}@s3.kth.se

*Automatic Control – School of Electrical Engineering, KTH,
SE-100 44 Stockholm, Sweden

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.

In network congestion control, validation experiments will with necessity be performed in closed loop since the communication protocol has to be active. Guidelines on how such experiments should be carried out in practice has until now been lacking in the literature. Departing from the theory of modeling for control, we refine a 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. Furthermore, we analyze, from a closed-loop perspective, how the network should be excited when validating such models. The resulting identification framework is used for validating the derived model using packet-level experimental data from NS-2 simulations.

Keywords: Communication Protocols, Communication Networks, Identification, Closed loop, Computer networks

1. INTRODUCTION

Exponential growth, dramatical technological development (both on the hardware side as well as the software side) and the heterogeneous and distributed nature of the Internet poses several challenging problems considering the design of future congestion control algorithms. A breakthrough in the theoretical understanding of network congestion control was made in the late 90s by Kelly and coworkers (Kelly *et al.*, 1998), who, by using the correct level of aggregation—namely fluid flow models valid for time-scales of a round-trip time (RTT) and more—presented a framework that not only admits equilibrium point analysis using a convex optimization approach but also allows for analysis and synthesis using control theory. Departing from this early work, this far, from a control perspective, the main focus has been on stability, with numerous contributions focusing on proving stability for more or less general configurations and scenarios as a result, see e.g. (Low *et al.*, 2002; Srikant, 2004; Jacobsson and Hjalmarsson, 2005a) and references therein. However, to be applicable, such stability results assumes models that

capture the dynamical properties of the true system sufficiently well. Thus model validation is an important issue for this type of models. A particular feature of this problem is that, naturally, the communication protocol has to be active during experimentation and that, hence, the experiments are performed in closed loop. Until now fluid models have mostly been validated coarsely by performing simulations that verify analytical stability bounds. Work on model validation from a system identification perspective is almost non-existent and the fallacies of closed-loop identification and model validation seem largely unknown. As an example, (Wei, 2004) and (Wang *et al.*, 2005) validate a static link model using experimental data but does not account for closed-loop effects.

A rich theory for the fundamental aspects of closed loop identification and model validation have been developed within the system identification community and the objective of this contribution is to highlight how this theory, and the associated methods, apply to fluid flow modeling. Hopefully, this will help in future development of more accurate fluid flow models and ultimately better congestion control algorithms.

We first present a refined fluid flow model and based on this model we discuss how a source/resource system should be excited when performing validation ex-

¹ This work was supported by European Commission through the project EuroNGI, by Swedish Research Council and by SCINT.

periments. Furthermore we provide illustrative identification examples—based on packet-level data from NS-2 simulations—that demonstrates our results as well as serves as practical examples on how fluid flow model validation is carried out in practice.

The paper is organized as follows. In Section 2 the mathematical abstraction of a network is derived. This model is used for analysis and validation in Section 3. Conclusions are finally given in Section 4.

2. 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. (Low *et al.*, 2002; Paganini *et al.*, 2005; Jin *et al.*, 2004).

2.1 Fluid-flow model

The bottle-necks of a generic communication 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 $l \in \{1, \dots, L\}$. The network is shared by a set of N persistent flows, competing about the offered capacity and indexed with $n \in \{1, \dots, N\}$. 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 carrying the same traffic are viewed as one link). The impact of non-bottle neck links and short lived (“mice”) traffic will be incorporated below, see Sections 2.2 and 2.3.

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 $t + \tau_{ln}^f$. This *forward delay* τ_{ln}^f 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_{ln}^b is defined in the same manner, i.e. it is the time it takes from that a packet passes 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 $\tau_n := \tau_{ln}^f + \tau_{ln}^b$. 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 $x_n(t - \tau_{ln}^f)$, are replaced by corresponding equilibrium values whereas delays appearing explicitly will be accounted for in full, see (Low *et al.*, 2002) and (Liu *et al.*, 2005) 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 $x(t) = [x_1(t), \dots, x_N(t)]^T$ and the corresponding vector

of forward delays to link l is denoted τ_l^f , with the convention that elements which correspond to $R_{nl} = 0$, i.e. not relevant for the network configuration, are set to zero. (Superscript T denotes matrix transpose.) Similarly defined, τ_n^b 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

$$y_l(t) = \sum_{n=1}^N R_{ln} x_n(t - \tau_{ln}^f) =: r_f(x(t), \tau_l^f) \quad (1)$$

where $y_l(t)$ must not exceed the associated capacity c_l in equilibrium.

Motivated by seminal work by Kelly and coworkers (Kelly *et al.*, 1998) and Low and Lapsley (Low and Lapsley, 1999), 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 $p(t) = [p_1(t), \dots, p_L(t)]^T$. The *aggregate price* received at source n is defined as

$$q_n(t) = \sum_{l=1}^L R_{ln} p_l(t - \tau_{ln}^b) =: r_b(p(t), \tau_n^b). \quad (2)$$

It is assumed that source n has access to a (possibly) corrupt version $\tilde{q}_n(t)$ of the aggregate price $q_n(t)$. A schematic representation of the closed loop system is given in Fig. 1. Here \mathcal{S} represents the source dynamics and \mathcal{L} represents the link dynamics, i.e. the pricing mechanism. The map $H : q_n(t) \rightarrow \tilde{q}_n(t)$

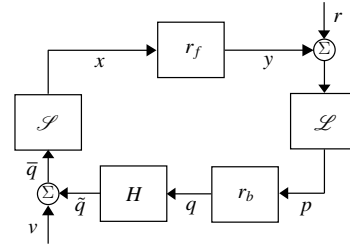


Fig. 1. Schematic representation of the delayed closed loop system of interconnected sources/resources.

depends on the used price mechanism and how it is communicated to the source. As an example, H could represent a quantizer.

2.2 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 aggregate price $q(t)$.

2.3 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 2.1. 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. 1, meanwhile keeping the network configuration fixed.

2.4 Source and link dynamics

The source dynamic should produce suitable sending rates x . As presented in (Kelly *et al.*, 1998; Low and Lapsley, 1999), 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 of explicit communication between sources. In, so called, primal/dual control (Kelly *et al.*, 1998) 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. 2. However, it can equally well be done before any processing is performed as in the right block \mathcal{S}_2 in Fig. 2. One should notice that it is important to include an anti-aliasing filter (represented by AI blocks in Fig. 2) 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 (Floyd and Henderson, 1999) or TCP Vegas (Brakmo and Peterson, 1995). 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. 2 which typically contribute to an increase in

the overall system delay, preceded by down-sampling and possibly also an anti-aliasing operation.

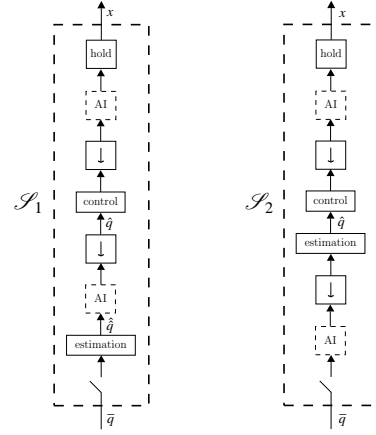


Fig. 2. A split view of the source dynamics.

2.5 Linearized system

The focus of this paper is on *local* properties of the interconnected source/resource system, i.e. small perturbations $x = x^* + \delta x$, $y = y^* + \delta y$, $p = p^* + \delta p$, $q = q^* + \delta q$ 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*.

By neglecting variations in delays appearing in variable arguments, as previously stated, we regard (1) and (2) as time invariant. Hence, the Laplace transform is applicable which yields that the linearized aggregate quantities can be expressed as

$$y(s) = R_f(s)x(s) \quad (3)$$

$$q(s) = R_b^T(s)p(s) \quad (4)$$

in the frequency domain. 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 $e^{-\tau_m^f s}$. The backward delay matrix $R_b(s)$ is obtained similarly, using backward delays $e^{-\tau_m^b s}$ instead.

Denote the 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

$$x(s) = \mathcal{K}(s)q(s) \quad (5)$$

$$p(s) = \mathcal{F}(s)y(s) \quad (6)$$

in vector form. Now: (3), (4), (5) and (6)—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.

3. CLOSED LOOP IDENTIFICATION

In this section we discuss the issue of validating models using experimental data in network congestion control. Since the communication protocol has to be active, validation experiments will with necessity be performed in closed loop. Hence, the excitation must be chosen with care to avoid misleading results. There

is an extensive literature on how to design appropriate excitation and which identification methods that can be applied, see e.g. (Ljung, 1999; Forssell and Ljung, 1999; Van den Hof and Schrama, 1995).

3.1 System excitation

It is well known that in closed-loop identification, for certain types of excitation and certain methods there is a risk of identifying the inverse of the regulator instead of the desired dynamics, or a linear combination of this dynamics and the desired dynamics. We will illustrate this for the closed loop network system in Fig. 1 with corresponding feedback configuration

$$x = \mathcal{S}(Hr_b p + v), \quad p = \mathcal{L}(r_f x + r), \quad (7)$$

when it is excited with different excitation signals.

Suppose that we are able to collect experimental data x at the sources and p at the links. From now on, for simplicity, assume that H is identity. When the system is excited through r only, we have in the Laplace domain

$$x = (I - \mathcal{S}R_b^T \mathcal{L}R_f)^{-1} \mathcal{S}R_b^T \mathcal{L}r, \quad (8)$$

$$p = (I - \mathcal{L}R_f \mathcal{S}R_b^T)^{-1} \mathcal{L}r. \quad (9)$$

By use of the matrix identity $(I + AB)^{-1}A = A(I + BA)^{-1}$ it is evident from (8) and (9) that

$$x = \mathcal{S}R_b^T (I - \mathcal{L}R_f \mathcal{S}R_b^T)^{-1} \mathcal{L}r = \mathcal{S}R_b^T p. \quad (10)$$

This leads us to the conclusion that: identifying source dynamics \mathcal{S} (concatenated with back-way delay configuration) from closed loop data $\{x, p\}$ and excitation in r only, is possible indeed; while any attempt to identify the link dynamics \mathcal{L} in the same manner will fail.

By exciting the system by means of the external signal v instead we get

$$x = (I - \mathcal{S}R_b^T \mathcal{L}R_f)^{-1} \mathcal{S}v, \quad (11)$$

$$p = (I - \mathcal{L}R_f \mathcal{S}R_b^T)^{-1} \mathcal{L}R_f \mathcal{S}v, \quad (12)$$

and furthermore, similarly to the previous case

$$p = \mathcal{L}R_f (I - \mathcal{S}R_b^T \mathcal{L}R_f)^{-1} \mathcal{S}v = \mathcal{L}R_f x. \quad (13)$$

Naturally, we observe that the link dynamics \mathcal{L} in concatenation with the forward delay matrix R_f is possible to identify in this experiment configuration; this in contrast to source dynamics \mathcal{S} , which is not.

Summing up the previous discussion: it is realized that in a scenario where individual source rates x and link prices p are collected and the network delay configuration is known (i.e. R_f and R_b) it is possible to

- i) identify the source dynamics \mathcal{S} by means of external changes in the aggregate rate y , i.e. a change in the available bandwidth via r ;
- ii) identify the link dynamics \mathcal{L} by the means of external changes in the aggregate price \bar{q} through the disturbance signal v .

To maintain consistent estimates when the system is operating in closed loop it is essential to chose identification method with care. To avoid bias when the Prediction Error Method is used it is necessary that the true dynamics is within the chosen model

structure and also that the true noise is included in the given noise description. When the external input also is known, an indirect method such as the Two-Stage Method (Van den Hof and Schrama, 1995) allows consistent estimation even without a noise model. All these methods work when there is excitation both in r and v .

3.2 Excitation in practice

Since it is not obvious how changes in the external signals r and v should be applied in practice this issue is further discussed here.

3.2.1. Changes in r In Section 2.3 the signal r is introduced to model persistent changes in the aggregate rate y at the bottle-neck links, i.e. changes in the available bandwidth viewed from the source side. This external excitation can be realized in practice in a network bed or simulations (NS-2, e.g.) by applying suitable cross traffic, e.g. by means of UDP protocols or similar. Example 3.1 later on serves as an illustration on this issue. However, we also remark that due to that packets are not perfectly smooth at packet-level, the experimenter should be aware of that changes in cross-traffic influences the noise levels of the system as well.

3.2.2. Changes in v Compared to the aggregate rate case, realizing external changes in the aggregate price q by applying cross-traffic is a formidable task. The model of Section 2 uses v to model short-term cross traffic that *occasionally* contributes to the aggregate price (e.g., buffers suddenly filling up due to trains of packets traversing the link in a delay based scheme). Changes in v must then be achieved by traffic carefully applied to non-bottle-neck links along the sources paths—however, without converting the link into a bottle-neck (which would change the whole configuration). Hence, an alternative approach must be taken when realizing other signals than what can be considered as noise. Fortunately, instead of applying cross-traffic, suitable changes in v is efficiently generated by direct manipulation of internal variables in the source algorithm. This is exemplified in Example 3.2 appearing in Section 3.3.3 where the external input is realized by imposing a step change to the price (queuing delay) estimate in the window control algorithm.

3.3 Experimental results

In the simulations we will use the time based TCP sibling FAST TCP recently proposed (Jin *et al.*, 2004) as congestion control protocol. Naturally we start by presenting a model of FAST TCP using the framework of Section 2. We refer to (Jacobsson and Hjalmarsson, 2005b) for a more rigorous derivation and model validation than given here.

3.3.1. The FAST TCP algorithm Designed for large distance high speed data transfers, FAST TCP adopts the delay-based approach of TCP Vegas (Brakmo and Peterson, 1995) to be able to attain a 'stable' equilibrium. This is, e.g., in contrast to 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 to guarantee full utilization of network resources. Additionally, FAST TCP has an equilibrium rate independent of network latency which is expressed by the constraint

$$x_n q_n = \alpha_n, \quad n = 1, \dots, N \quad (14)$$

where α_n is a parameter in the algorithm, in equilibrium. Below, the basic features of the FAST TCP algorithm are briefly described. We only consider the single source single bottle-neck link case here so subscript n is dropped for ease of notation.

The aggregate price in the algorithm is the queuing delay. Assuming that the latency d is known and denoting the k th round-trip time $\tau(k)$, 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

$$\hat{q}(k+1) = \left(1 - \frac{3}{w(k)}\right) \hat{q}(k) + \frac{3}{w(k)} \bar{q}(k) \quad (15)$$

when window is $w(k) > 12$.

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. 1, we will therefore use window sizes in the sequel. For each sender the window is adjusted according to

$$w(k+1) = (1 - \gamma)w(k) + \gamma \frac{d}{d + \hat{q}(k)} w(k) + \gamma \alpha, \quad (16)$$

where $\alpha \in \mathbb{Z}^+$ and $\gamma \in (0, 1]$ are protocol parameters and d is the network latency. Note that, in the sequel, 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 .

3.3.2. A FAST TCP model It turns out that the analysis is simplified by working in continuous time so continuous time equivalents are given for as many blocks as possible of the system. Equilibrium values are denoted with subscript 0 in the sequel.

Linearization and inverse sampling of the queue estimator algorithm (15) and the window update (16) yields

$$\dot{\hat{q}}(t) = -\frac{3}{T_q(\alpha + x_0 d)} \hat{q}(t) + \frac{3}{T_q(\alpha + x_0 d)} \bar{q}(t), \quad (17)$$

$$\dot{w}(t) = -\frac{1}{T_w} \gamma \frac{\alpha}{\alpha + x_0 d} w(t) - \frac{1}{T_w} \gamma \frac{x_0^2 d}{\alpha + x_0 d} \hat{q}(t) \quad (18)$$

respectively.

A sampler with the sampling time denoted by $T_{\tilde{w}}$ in conjunction with a zero-order-hold (ZOH) describes the signal map $w \mapsto \tilde{w}$.

The link is modeled as a continuous integrator, driven by the incoming rate at the link,

$$\dot{p}(t) = \begin{cases} \frac{\tilde{w}(t - \tau^f) - c}{d + p(t)}, & \text{if } p(t) > 0 \text{ or } \frac{\tilde{w}_n(t - \tau^f)}{d + p(t)} > c, \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

which linearized around the equilibrium $p_0 = q_0 = \alpha/x_0$ become

$$\dot{p}(t) = -\frac{x_0^2}{c(\alpha + x_0 d)} p(t) + \frac{x_0}{c(\alpha + x_0 d)} \tilde{w}(t - \tau^f). \quad (20)$$

The resulting mathematical abstraction of a single FAST TCP source sending over a single bottle-neck is represented by the block diagram of the closed-loop system shown in Fig. 3, 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 (17), (18) and (20) (excluding delay which is accounted for separately), respectively. The system in Fig. 3 cannot directly be analyzed in

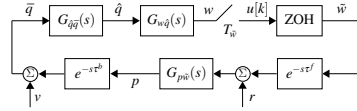


Fig. 3. Block diagram of a single FAST TCP source sending over a single bottle-neck link.

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. In simulations this simplification effectively amounts to replacing the zero-order-hold function by a continuous time delay of $T_{\tilde{w}}$.

3.3.3. Identification using packet-level data To verify the findings in Section 3.1 we use the discrete event simulator NS-2 to generate experimental packet-level data that is used to check which relationship the queue size and window size samples obey for the two different scenarios considered: i.e. (1) excitation through r ; and, (2) excitation via v , in Fig 3. This is illustrated in Example 3.1 and Example 3.2 respectively.

The experimental setting is a single FAST TCP source sending over a single bottle-neck 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 $\tau_f = 0$ ms and $\tau_b = 100$ ms. The FAST TCP parameters are set to $\alpha = 400$, $\gamma = 0.2$ and $T_{\tilde{w}} = T_w = 0.01$.

Example 3.1. A step in r of magnitude $c/5$ is applied at time $t = 10$ to the system in Fig. 3—i.e. a negative step is taken in the available bandwidth (realized by applying appropriate UDP traffic). Queue size and window size samples are collected. The solid lines in Fig. 4 shows the window size (upper plot) and queue size (lower plot) from the NS-2 simulation. The queue size data was filtered through the source control dynamics, i.e. $G_{w\hat{q}}(s)G_{\hat{q}\bar{q}}(s)e^{-s\tau^b}$, and the window size data was filtered through the queue dynamics, i.e. $G_{p\tilde{w}}(s)$ (note that $\tau^f = 0$). The resulting signals are shown in Fig. 4. It is observed that the controller dynamics yields a superior fit as predicted by the analysis in Section 3.1. We remark that x_0 is taken as the equilibrium after the occurrence of a change in conditions. ■

Example 3.2. The scenario is identical as in Example 3.1 except that the system in Fig. 3 is excited via a change in v instead of through r . It is observed that the queue dynamics then yields a superior fit as predicted by the analysis in Section 3.1. We remark

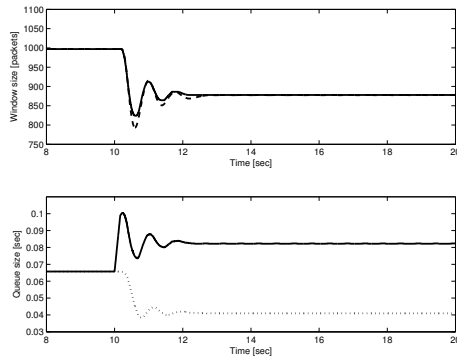


Fig. 4. Identification example (step in r). Dotted line: queue dynamics (simulated in Matlab). Dashed line: protocol dynamics (simulated in Matlab). Solid line: true dynamics (NS-2).

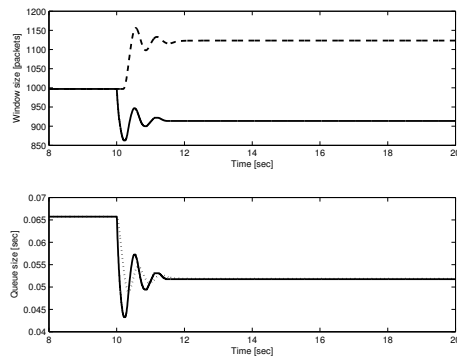


Fig. 5. Identification example (step in v). Dotted line: queue dynamics (simulated in Matlab). Dashed line: protocol dynamics (simulated in Matlab). Solid line: true dynamics (NS-2).

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. ■

It is observed, by studying Example 3.1 and Example 3.2, that the model of the source control dynamics seems to be more accurate compared to the link model. Considering sending rates it is clear that the FIFO queue is simply an integrator in the linear regime. The dynamical description between congestion window and queue size is, however, not obvious at all. Modelling of the dynamics between congestion window and the queue size in delay based schemes is still an open problem and several different models have been proposed in the literature, see e.g. (Low *et al.*, 2002; Liu *et al.*, 2005; Jin *et al.*, 2004) and references therein. Methodical evaluation of existing link models and development of new models in delay based schemes is ongoing work and will be reported elsewhere.

4. 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. 1. Also, dynamics has been

incorporated which extends the validity of the model to a wide range of operating conditions. In the context of this framework, we have analyzed how such models should be validated using network measurements. Such experiments have to be performed in closed loop since the communication protocol has to be active. Finally, we have provided illustrative identification examples based on packet-level data from NS-2 simulations that demonstrates our results as well as serves as practical examples on how theory is put into practice.

REFERENCES

- Brakmo, L. S. and L. L. Peterson (1995). TCP Vegas: end-to-end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications* **13**(8), 1465–1480.
- Floyd, S. and T. Henderson (1999). The NewReno modification to TCP’s fast recovery algorithm. RFC 2582.
- Forssell, U. and L. Ljung (1999). Closed-loop identification revisited. *Automatica* **35**(7), 1215–1241.
- Jacobsson, K. and H. Hjalmarsson (2005a). Local analysis of structural limitations of network congestion control. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. Seville, Spain.
- Jacobsson, K. and H. Hjalmarsson (2005b). Towards accurate congestion control models: Validation and local performance and stability analysis. submitted.
- Jin, C., D. X. Wei and S. H. Low (2004). FAST TCP: motivation, architecture, algorithms, performance. In: *Proceedings of IEEE Infocom*.
- Kelly, F., A. Maulloo and D. Tan (1998). Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* **49**, 237–252.
- Liu, S., T. Basar and R. Srikant (2005). Pitfalls in the fluid modeling of RTT variations in window-based congestion control. In: *Proceedings of IEEE Infocom 2005*.
- Ljung, L. (1999). *System Identification: Theory for the User*. 2nd ed.. Prentice-Hall. Englewood Cliffs, NJ.
- Low, S. H. and D. E. Lapsley (1999). Optimization flow control – I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking* **7**(6), 861–874.
- Low, S. H., F. Paganini and J. C. Doyle (2002). Internet congestion control. *Control Systems Magazine* **22**(1), 28–43.
- Paganini, F., Z. Wang, J. C. Doyle and S. H. Low (2005). Congestion control for high performance, stability, and fairness in general networks. *IEEE/ACM Trans. Netw.* **13**(1), 43–56.
- Srikant, R. (2004). *The Mathematics of Internet Congestion Control*. Birkhauser.
- Van den Hof, P. M. J. and R. J. P. Schrama (1995). Identification and control – closed loop issues. *Automatica* **31**(12), 1751–1770.
- Wang, Jiantao, David X. Wei and Steven H. Low (2005). Modelling and stability of FAST TCP. In: *Proceedings of IEEE Infocom 2005*.
- Wei, David X. (2004). Congestion control algorithms for high speed long distance TCP connections. Master’s thesis. Caltech.