2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

An axiomatic fluid-flow model for congestion control analysis

C. Briat
H. Hjalmarsson, K.H. Johansson, U.T. Jönsson, G. Karlsson, H. Sandberg and E.A. Yavuz

Abstract— **An axiomatic model for congestion control is derived. The proposed four axioms serve as a basis for the construction of models for the network elements. It is shown that, under some assumptions, some models of the literature can be recovered. A single-buffer/multiple-users topology is finally derived and studied for illustration.**

Index Terms— **Congestion control modeling; conservation-law; time-delay systems; state-dependent delay**

I. INTRODUCTION

Network modeling is challenging due to the very heterogeneous nature of communication networks, mixing physics, electronics and computer science. This heterogeneity coupled with intrinsic properties of the physical and mathematical laws prevent the development of an efficient bottom-up approach. This is the reason why finding macroscopic axioms capturing the critical phenomena is of interest. These axioms should provide an abstraction of the microscopic level by identifying and relating the fundamental macroscopic network parameters.

The network model is derived using four fundamental axioms. The zeroth axiom defines a consistent notion of time on which the network modeling problem is solvable and admits a scalable solution. The first axiom is a packet conservation law facilitating the derivation of models for building blocks and simplifying their mathematical expression. The second axiom defines a model for queues. Finally, the last axiom concerns the existence of a user model. We will show that each axiom has implications in the network modeling problem and, more importantly, will allow to solve yet unresolved problems, especially at user level.

The proposed model will be developed in several steps. The first one will be devoted to the modeling of lossless transmission channels directly from the first axiom. The second step is the derivation, again from the first axiom, of the so-called ACK-clocking model [1], which relates flow, flight-size and round-trip time (RTT) together. This result is of great importance in network modeling.

Based on the first and second axioms, a causal RTT expression is developed in the third step. This causal RTT expression will however require an extension of the buffer model. Indeed, there are two main limitations to the buffer model usually considered in the literature (and as stated in the second axiom). First, it does not explicitly define the queue as a FIFO queue (i.e. order preserving) in which the packets maintain their relative positions. An internal buffer description should capture this, at a flow level. Therefore, the flows should be considered as very viscous repelling liquids which do not mix. The second limitation concerns the solving of the output flow separation problem, primordial for the description of buffers interconnections and, as we shall see later, for the derivation of an exact expression for the acknowledgments flows.

Finally, the last step is devoted to the derivation of a complete user model, based on the first and third axioms. This part constitutes one the most important contributions of the paper. Indeed, the conversion of windows size into flow has been a major obstacle preventing the improvement of network models. The *static-link model* [2] assumes a static relationship between window sizes and queueing delays. It has good modeling properties in the absence of cross-traffic and when propagation delays are homogeneous. It is however rather inaccurate in more realistic scenarios. This validity domain is theoretically proved in this paper by showing that the proposed model reduces to the static-link one when some conditions are met. The *integrator link model* [3], [4], [5] improves the description by correcting the irrelevant behavior of the model when affected by cross-traffic. Yet, some characteristics of the buffer response were not well captured: the response speed and the high slope when the window size increases. More recently, the *joint-link model* [6], [7] consisting of merging the static-link and integrator-link models has been introduced. This approach improves the network model by capturing some characteristics unmodeled by the previous descriptions. It has been shown that these flow models can, in fact, be considered as approximations of the ACK-clocking model [6], [7] from which higher order approximations can also be defined. In this paper, we do not make any approximations and use the ACK-clocking model in a new fashion, leading to a new user model.

The last part of the paper will focus on the development of a general model for networks using the building blocks obtained in the paper. The modeling technique will be applied to a single-buffer/multiple-users topology for which it will be possible to show that, under some certain conditions, the static-link model can be naturally recovered. More importantly, the provided static-link model involves a state-dependent time-delay modeling the queuing delay. In this context, the local stability of the FAST-TCP protocol [8] is analyzed.

II. NETWORKS AND GRAPHS

It is convenient to introduce here the particular network graph representation considered in the paper. It is different

This work has been supported by the ACCESS project, KTH, Stockholm, Sweden. http://www.access.kth.se/, e-mail: {cbriat,hakan.hjalmarsson,gk,kallej,ulfj,hsan,emreya}@kth.se

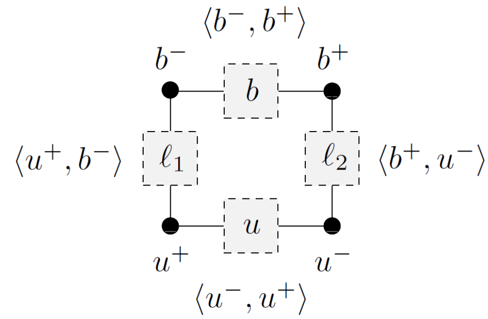

Fig. 1. Example of graph with 4 edges: one user edge $u = \langle u^-, u^+\rangle$, one buffer edge $b = \langle b^-, b^+\rangle$ and two transmission edge $\ell_1 = \langle u^+, b^-\rangle$, $\ell_2 = \langle b^+, u^-\rangle$

from the regular ones since it places all network elements on graph edges, leaving nodes with the role of connecting points, as in electrical circuits. Four types of nodes are distinguished: the input nodes u_i^-, b_j^- and output nodes u_i^+, b_j^+ for user u_i and buffer b_j respectively. The superscripts have to be understood as a temporal order of reaction or causality: the data come (-) then leave (+). We will denote any edge E of the graph by $\langle x, y\rangle$ where x and y are the input and output nodes respectively. Moreover, given any edge E, the input and output nodes are given by $\beta(E)$ and $\varepsilon(E)$ respectively.

According to these definitions, a queue edge is always denoted by $\langle b_i^-, b_i^+\rangle$, a user edge by $\langle u_i^-, u_i^+\rangle$ and a transmission edge by $\langle b_i^+, u_j^-\rangle$, $\langle u_i^+, b_j^-\rangle$ or $\langle b_i^+, b_k^-\rangle$, $i \neq k$. This is illustrated in Fig. 1. We call a circuit, say C, a path from the output to the input of a user, i.e. $C = \langle u^+, u^-\rangle$. In Fig. 1, the only possible circuit is given by $C = \langle u^+, b^-, b^+, u^-\rangle$.

III. The Networking Axioms

A. Universal Clock Existence

In an asynchronous network like the Internet, each element can be considered to have its own discrete-time clock $\mathbb{T}_i \subset \mathbb{R}_+$ where $\mathbb{T}_i$ is countable, ruling out the rhythm of protocol decisions and packets transmission. When several sources send data through the same buffer/path, a clock-coupling takes place and clock-interferences arise. Modeling this clock-coupling is of incredible complexity since the number of clocks and their interactions grow very quickly with the network complexity, leading then to a very complicated structure for the interrelated local clocks $\mathbb{T}_i$, see [6, Equations (3.7)] for example. In communication networks, clocks beat with the rhythms of acknowledgment reception rates, which are influenced in turn by network congestion; this is referred to as *ACK-clocking*[1].

An idea to resolve this complex time-structural problem is the definition of a *universal clock* $\mathbb{T}_u$ dictating a common time to the entire network. This leads us to the zeroth axiom:

Axiom 0: There exists an ideal universal clock $\mathbb{T}^u$ embedding any local clock $\mathbb{T}_i$, i.e. $\mathbb{T}^u \supset \bigcup_i \mathbb{T}_i$. $\therefore$

[1]The term *self-clocking* is also used but is less explicit.

According to the above axiom, it may seem reasonable to assimilate the universal clock $\mathbb{T}^u$ to a clock running over positive real numbers continuously, i.e. $\mathbb{T}^u \equiv \mathbb{R}_+$. This particular universal clock dramatically simplifies the modeling problem and this is the reason why it will be considered in this paper. Indeed, using such a time-scale, well-established mathematical tools can be used to model and analyze networks: real functions analysis, integration theory, dynamical systems, difference and differential equations, etc. A conclusion is that continuous-time models may be used to describe networks [9], [3]. These are generally referred to as *fluid-flow models*.

Within this framework, it is possible to provide a proper definition for flows of data. Indeed, given any edge E of the network, a nonnegative integrable flow $\phi(x, s)$, $(x, s) \in E \times \mathbb{T}^u$ and time instants $t_0, t \in \mathbb{T}^u$, $t \geq t_0$, we have

$$N_x(t, t_0) = \int_{t_0}^{t} \phi(x, s)ds \tag{1}$$

where $N_x(t, t_0)$ is a packet counter, i.e. the number of packets having passed through point x between t_0 and t is given by $N_x(t, t_0)$. The integral considered above is a very standard one, e.g. the Lebesgue integral.

B. Spatial and Temporal Conservation

The first axiom is essentially a law of conservation relating flow integration on two different domains. The overall idea is to remark that we can count the total number of packets in transit in any edge, simply by counting the number of entering packets over a certain time horizon.

Axiom 1: Given any edge E of the network, then for all $t \in \mathbb{T}^u$ there exists a time $t_0(t) \in \mathbb{T}^u$, $t_0(t) \leq t$ such that

$$\begin{array}{rcl} P_E(t) & := & \int_E \phi(\theta, t)d\theta \\ & = & \int_{t_0(t)}^{t} \phi(\beta(E), s)ds \\ & = & N_{\beta(E)}(t, t_0(t)) \end{array} \tag{2}$$

The integration over E is an abstract integral which has to be understood as a flow integration from $\beta(E)$ to $\varepsilon(E)$, that is, the number of packets $P_E(t)$ in the edge $E = \langle \beta(E), \varepsilon(E)\rangle$ at time t. $\therefore$

The axiom's main features are the domain of integration exchange and the discretization of the spatial domain to nodes only. These considerations will dramatically simplify the modeling since it is no longer necessary to consider the flows at any point $x \in E$ but only at input nodes $\beta(E)$. This is shown in the following proposition:

Proposition 1: The input flow $\phi(\beta(E), \cdot)$ and output flow $\phi(\varepsilon(E), \cdot)$ of edge E verify

$$\phi(\varepsilon(E), t) = t_0(t)'\phi(\beta(E), t_0(t)) \tag{3}$$

where we have tacitly assumed that $t_0(t)$ is differentiable.

Proof: Since $N_{\beta(E)}(t, t_0(t))$ is the current number of packets in the edge E at time t, then differentiation with respect to time provides the corresponding rate of variation

$$N_{\beta(E)}(t, t_0(t))' = \phi(\beta(E), t) - t_0(t)'\phi(\beta(E), t_0(t)).$$

Moreover, the variation of the number of packets verifies

$$N_{\beta(E)}(t,t_0(t))' = \phi(\beta(E),t) - \phi(\varepsilon(E),t)$$

which is nothing else but the difference between the input and output flows. The result follows from identification of the equalities. ■

This proposition will turn out to be very useful to derive models for transmission channels and buffers.

C. Queues are flows integrators

The axiom given below defines the behavior of queues involved, for instance, inside routers and servers. Following past works and our understanding of the problem, the integrator model for queues seems to be the most plausible.

Axiom 2: The queue dynamics of buffer i is governed by the model

$$\dot{q}_i(t) = \sum_j \phi_j(b_i^-,t) - r_i(t) \tag{4}$$

with aggregated output flow rate

$$r_i(t) = \begin{cases} c_i & \text{if } \mathcal{C}_i(t) \\ \sum_j \phi_j(b_i^-,t) & \text{otherwise.} \end{cases} \tag{5}$$

Above, q_i, c_i and $\phi_j(b_i^-,t)$ represent the queue size, the maximal output capacity and the flow of type j at the input respectively. The condition $\mathcal{C}_i(t)$ is given by

$$\mathcal{C}_i(t) := \left([q_i(t) > 0] \vee \left[\sum_j \phi_j(b_i^-,t) > c_i \right] \right). \tag{6}$$

The corresponding queuing delay can be easily deduced using the relation $\tau_i(t) = q_i(t)/c_i$. ∴

The above model can indeed be refined to capture additional features like finite maximal queue length, flow priorities, multiple output capacities, etc.

D. Users model existence

The last axiom concerns the user protocol description and the way it dynamically reacts to the congestion in the network.

Axiom 3: There exist bounded functions/functionals $\mathcal{P}_i$, $\mathcal{W}_i$ and $\mathcal{U}_i$ such that the trajectories $(z_i(t), w_i(t))$ of the following continuous-time model defined over $\mathbb{T}^u$

$$\begin{array}{rcl} \dot{z}_i(t) & = & \mathcal{P}_i(z_i(t),\mu_i(t)) \\ w_i(t) & = & \mathcal{W}_i(z_i(t),\mu_i(t)) \\ \phi_i(u_i^+,t) & = & \mathcal{U}_i(w_i(t),\phi_i(u_i^-,t)) \end{array} \tag{7}$$

match the trajectories of the asynchronous protocol (defined on $\mathbb{T}_i$) at points in $\mathbb{T}^u \cap \mathbb{T}_i$. Above, z_i, μ_i, $\phi_i(u_i^-,\cdot)$ and $\phi_i(u_i^+,\cdot)$ are the state of the protocol, the measurements, the acknowledgment flow rate and the user sending flow respectively. The window size w_i is considered here as the number of outstanding packets to track and is supposed to be differentiable. ∴

A procedure to solve the above interpolation problem for the FAST-TCP protocol is detailed in [10, Appendix C.].

IV. Transmission Channels With Constant Propagation Delay

We start by the following result obtained from Axiom 1:

Result 2: Given a lossless transmission channel, corresponding to an edge E, with constant propagation delay $T > 0$, the output flow is given by

$$\phi(\varepsilon(E),t) = \phi(\beta(E),t-T). \tag{8}$$

Proof: Following Axiom 1, the number of packets in transit $P_E(t)$ in the edge E at time $t \in \mathbb{T}^u$ obeys

$$\begin{array}{rcl} P_E(t) & = & \int_E \phi(x,t)dx \\ & = & \int_{t_0(t)}^t \phi(\beta(E),s)ds \\ & = & N_{\beta(E)}(t,t_0(t)), \end{array} \tag{9}$$

where $t_0(t) = t - T$ since the propagation delay is constant. Indeed, a packet sent at time $t - T$ will be, at time t, still in the circuit and about to leave. The result follows then from Proposition 1. ■

Using the notation defined in Section II, we can build the flow vectors $\phi(x,t)$, $x \in \bigcup_i \bigcup_j \{u_i^-, u_i^+\} \cup \{b_j^-, b_j^+\}$ using the 'col' operator:

$$\phi(x,t) \quad = \quad \mathrm{col}_{k=1}^{\sigma(x)} \left[\phi_k(x,t)\right] \tag{10}$$

where $\sigma(x)$ is the number flows using node x.

We are now in position to introduce the transmission channels operators.

Definition 3: The above vectors are related by the transmission channels operators $\mathcal{R}_x$, $x \in \{ub, bu, bb\}$ as

$$\begin{bmatrix} \phi(u^-,t) \\ \phi(b^-,t) \end{bmatrix} = \mathcal{R} \begin{bmatrix} \phi(u^+,t) \\ \phi(b^+,t) \end{bmatrix} + \mathcal{D}\delta(t) \tag{11}$$

where

$$\mathcal{R} \quad = \quad \begin{bmatrix} 0 & \mathcal{R}_{ub} \\ \mathcal{R}_{bu} & \mathcal{R}_{bb} \end{bmatrix} \quad \text{and} \quad \mathcal{D} \quad = \quad \begin{bmatrix} 0 \\ \mathcal{D}_b \end{bmatrix}.$$

The matrix $\mathcal{R}_x$, $x \in \{ub, bu, bb\}$, contains 0 entries except for those corresponding to an existing transmission channel. In this case, the entry contains a single delay operator $\mathscr{D}_T$, where T is the constant propagation delay corresponding to the transmission channel. The vector $\delta(t)$ consists of cross-traffic flows entering buffers. The full-rank matrix $\mathcal{D}_b$ contains 0 and 1 only.

V. The ACK-Clocking model

The ACK-clocking model [1] is certainly the most important consequence of Axiom 1. This model characterizes the *flight-size*[2] $\digamma_i(C_i,t)$ of a user i at any time $t \in \mathbb{T}^u$ over a closed circuit $C_i = \langle u_i^+, u_i^- \rangle$. The importance of the ACK-clocking model lies in the semantic it adds to the model by relating the RTT, flow and flight-size together. From this result, it will be possible to derive a number of important properties and rules for the network.

[2]The number of outstanding packets.

Result 4 (ACK-Clocking): The ACK clocking model is given by

$$\begin{aligned}\digamma_i(C_i, t + RTT_i\{t\}) &= \int_{C_i} \phi(\theta, t + RTT_i\{t\})d\theta \\ &= \int_{t}^{t+RTT_i\{t\}} \phi_i(u_i^+, s)ds\end{aligned} \tag{12}$$

where $RTT_i\{t\}$ is the RTT of a packet sent at at time t in the circuit C_i by user i. ▲

Proof: We assume here that the circuit is lossless for simplicity. The flight-size is indeed a spatial integration of flows since the number of packets in transit is equal to the spatial integral of the flows over the circuit. Thus, following Axiom 1, it is possible to convert the spatial integration into a temporal one provided that we can determine its integration domain. To obtain it, we use the notion of RTT and suppose that a data sent by user i in the circuit C_i at time t has a round-trip-time given by $RTT_i\{t\}$. This means that the data sent between t and $t + RTT_i\{t\}$ are unacknowledged and thus, still in the circuit. Hence, the corresponding temporal integration has bounds t and $t + RTT_i\{t\}$. ■

VI. RTT EXPRESSION AND INTERNAL BUFFER MODEL

In the light of the discussion above, it turns out that a model for the RTT is necessary in order to characterize and compute the flight size $\digamma_i(C_i, t+RTT_i\{t\})$. A step forward towards a RTT expression is the analysis of queuing delays in buffers.

A. Forward/Backward delays and Causal RTT expression

Since RTT directly depends on the queuing delays, the computation of it essentially means calculating the queueing delays. The results are taken from a previous work of us [11] with the difference that we relate them here to Axiom 2.

Considering the buffer model (4) with queueing delay $\tau_i(t)$, we define the *forward delay operator* $f_i(t) := t+\tau_i(t)$ which maps, at a flow level, any 'flow' input time t to the corresponding 'flow' output time $f_i(t)$. Although it is easy to derive and understand, this operator leads to a noncausal RTT expression which is not desirable. To observe this, let us consider a closed circuit C with N queues, indexed from 1 to N. The indices 0 and $N+1$ are used to denote the input and output of the circuit respectively. Given a packet input time t, the corresponding packet output time t_C is given by:

$$\begin{aligned}t_C(t) &= \mathscr{F}_C(t) \\ \mathscr{F}_C &= \mathcal{R}_{N,N+1}^{-1} \circ f_N \circ \mathcal{R}_{N-1,N}^{-1} \circ f_{N-1} \circ \dots \\ &\quad \circ \mathcal{R}_{2,3}^{-1} \circ f_2 \circ \mathcal{R}_{1,2}^{-1} \circ f_1 \circ \mathcal{R}_{0,1}^{-1} \circ ev\end{aligned} \tag{13}$$

where $\mathcal{R}_{i,j}$ is the entry of the routing matrix $\mathcal{R}$ corresponding to the transmission channel between the network elements i and j, ev is the evaluation map and $\circ$ the composition operator.

It is clear that $\mathscr{F}_C$ is noncausal since it requires the knowledge of future information, which is not available. In order to solve this problem, the *backward delay operator* [11] expressing the input time t as a function of the output time t_C, is considered instead. Hence, the problem reduces to inverting the forward delay operator. The existence of this inverse operator and some of its properties are recalled below:

Result 5 ([11]): The operator f_i is invertible if and only if the input flow of the corresponding buffer is almost everywhere positive. ▲

Result 6 ([11]): The functions f_i and g_i obey:

$$\begin{aligned}f_i(t) &= t + \tau_i(t) \\ g_i(t) &= t - \tau_i(g_i(t)) \\ g_i'(t) &= \begin{cases} c_i \left(\displaystyle\sum_{k=1}^{\sigma(b_i^-)} \phi_k(b_i^-, g_i(t)) \right)^{-1} & \text{if } \mathcal{C}_i(g_i(t)) \\ 1 & \text{otherwise} \end{cases} \\ f_i'(t) &= \begin{cases} c_i^{-1} \displaystyle\sum_{k=1}^{\sigma(b_i^-)} \phi_k(b_i^-, t) & \text{if } \mathcal{C}_i(t) \\ 1 & \text{otherwise} \end{cases}\end{aligned}$$

where $f'(t)$ stands for the the upper right Dini derivative of $f(t)$, i.e. $D^+[f](t) = \limsup_{h\downarrow 0} h^{-1}(f(t+h) - f(t))$. ▲

Using the backward delay operators g_i, the packet sending time t can be computed from the reception time t_C through the causal expression:

$$\begin{aligned}t(t_C) &= \mathscr{B}_C(t_C) \\ \mathscr{B}_C &= \mathcal{R}_{0,1} \circ g_1 \circ \mathcal{R}_{1,2} \circ g_2 \circ \mathcal{R}_{2,3} \circ \dots \\ &\quad \circ g_{N-1} \circ \mathcal{R}_{N-1,N} \circ g_N \circ \mathcal{R}_{N,N+1} \circ ev.\end{aligned} \tag{15}$$

Example 7: In the single-user/single-buffer case, the expressions reduce to

$$\begin{aligned}t(t_C) &= g(t_C - T_b) - T_f \\ t_C &= t + T_f + T_b + \tau(t + T_f) \\ RTT\{t\} &= t_C - t \\ &= t_C - g(t_C - T_b) + T_f \\ &= T_b + T_f + \tau(g(t_C - T_b))\end{aligned} \tag{16}$$

where T_f and T_b are the forward and backward propagation delays corresponding to $\mathcal{R}_{0,1}$ and $\mathcal{R}_{1,2}$.

Using the backward expression of the RTT, it easy to obtain the following result:

Result 8: The flight size obeys

$$\begin{aligned}\digamma_i(C_i, \mathscr{F}_{C_i}(t)) &= \int_{t}^{\mathscr{F}_{C_i}(t)} \phi(u_i^+, s)ds \\ \digamma_i(C_i, t) &= \int_{\mathscr{B}_{C_i}(t)}^{t} \phi(u_i^+, s)ds.\end{aligned} \tag{17}$$

▽

This is a direct consequence of Axiom 1 (through the ACK-clocking model) and Axiom 2. However, the problem is not completely resolved yet since it is still rather unclear how to compute the queuing delays along a given circuit. Indeed, calculating the queueing delays requires the knowledge of all the buffers input flows, and hence requires a way of splitting the upstream buffer aggregated output flows into distinct 'atomic' flows. Otherwise, a modular description of buffers interconnections is not possible.

B. FIFO Buffer output flow separation

Without further consideration on the queue type, there exists an infinite number of ways to separate the aggregated output flow directly from the queuing model of Axiom 2. When a FIFO queue (i.e. order preserving) is considered, it turns out that the output flow separation problem is easily solvable using Axioms 1 and 2. The FIFO characterization and output flow separation problems have been fully solved in [11]. In this section, we will recall and explain these results and relate them to Axioms 1 and 2.

Result 9 ([11]): Let us consider the queueing model (4). The output flow corresponding to the input flow $\phi_\ell(b_i^-,t)$ is given by

$$\begin{array}{rcl} \phi_\ell(b_i^+,t) & = & g_i'(t)\phi_\ell(b_i^-,g_i(t)) \\ & = & \left\{ \begin{array}{ll} \dfrac{c_i\phi_\ell(b_i^-,g_i(t))}{\sum_j^{\sigma(b_i^-)}\phi_j(b_i^-,g_i(t))} & \text{if } \mathcal{C}_i(g_i(t)) \\ \phi_\ell(b_i^-,t) & \text{otherwise} \end{array} \right. \end{array} \tag{18}$$

Proof: The proof is available in [11] and is based on the analysis of the contribution of each input flow to the queue size. Then, using Axioms 1 and 2, it is possible to split the aggregate output flow into atomic output flows that correspond to each input flow. ■

This model deserves interpretation: the output flows consist of a scaling and shifting of the input flows. The delay accounts for the high flow viscosity and captures the queue FIFO behavior, at a flow level. This model also tells that the output flow $\phi_\ell(b_i^+,t)$ corresponding to the input flow $\phi_\ell(b_i^-,t)$ is expressed as a (delayed) ratio of the input flow $\phi_\ell(b_i^-,t)$ over the total input flow that entered the buffer at the same time. Hence, the output flows are equal to the relative flows proportions, scaled-up by the output capacity in order to utilize the available bandwidth. This nonlinear expression for the output flows describes the flow- and clock-coupling phenomena discussed in Section III-A.

Definition 10: The buffer operator $\mathcal{B}_i$ with $\sigma(b_i^-)$ input flows is defined as

$$\begin{array}{rccc} \mathcal{B}_i & : & \mathbb{R}^{\sigma(b_i^-)} & \rightarrow & \mathbb{R}^{\sigma(b_i^-)} \\ & & \phi(b_i^-,t) & \rightarrow & \phi(b_i^+,t) \end{array} \tag{19}$$

where the output flows and the buffer state are governed by (4) and (18). Using these operators, we can build a matrix of operators $\mathcal{B}$ connecting the $\phi(b^-,t)$'s to the $\phi(b^+,t)$'s as

$$\phi(b^+,t) = \mathcal{B}\phi(b^-,t) \tag{20}$$

where $\mathcal{B} = \mathrm{diag}_i\{\mathcal{B}_i\}$.

VII. Complete user model - Window control

As explained in the introduction, the user modeling problem is partially an open question and a solution, based on Axioms 1 and 3, is proposed here. The distinct notions of flight-size, ACK-flow and window-size are clarified first and associated with each other. Then, the problem of computing users sending flow $\phi(u_i^+,t)$ is solved. Finally, the window-to-flight-size conversion problem, accounting for flight-size rate of variation constraints, is addressed.

A. ACK-clocking dynamics and user flow computation

The approach exposed here is based on the ACK-clocking model of [6], obtained in Section V from Axioms 1 and 2.

Result 11: Let us consider a circuit $C_i = \langle u_i^+, u_i^- \rangle$. Then the ACK-flow the user u_i receives is given by

$$\phi(u_i^-,t) = \mathscr{B}'_{C_i}(t)\phi_i(u_i^+,\mathscr{B}_{C_i}(t)). \tag{21}$$

▽

Proof: The key idea is to remark that $\digamma_i(C_i,t) = N_{u_i^+}(t,\mathscr{B}_{C_i}(t))$. Hence, using Proposition 1 and noting that the ACK-flow is the leaving flow from the circuit $\phi(u_i^-,t)$, we have the result. ■

Note that differentiation of (17) also yields

$$\phi_i(u_i^+,t) = \digamma_i'(C_i,t) + \mathscr{B}'_{C_i}(t)\phi_i(u_i^+,\mathscr{B}_{C_i}(t)) \tag{22}$$

meaning that, to maintain the same flight size, the user has to naturally send data at the same rate as receiving ACK packets: this is exactly ACK-clocking but expressed at a flow level. By flow integration, we can easily recover the 'packet-level ACK-clocking'.

B. User flow and windows size

We clarify here the relation between the user sending flow $\phi(u_i^+,t)$ and its window size $w_i(t)$. First, recall that the window size corresponds to the desired flight-size while the flight-size is the current number of packets in transit. The window size is then a *reference* to track while the flight size is the *controlled output*. The *control input* is the user sending flow for which constraints must be considered.

Indeed, when the window size increases, the user can immediately send a burst of new packets to equalize the flight- and windows-sizes. In such a case, we can ideally assimilate them to be equal (and so are their derivatives). The small delay corresponding to the protocol reaction time can be easily incorporated in the constant part of the RTT. The problem is, however, slightly more difficult when the window size decreases at a rate below a certain threshold depending on the received ACK-flow. In such a case, we cannot withdraw packets from the network and the only thing we can do is waiting for new ACK packets until the flight size becomes equal to the window size. Therefore, while the slope of the flight-size is unbounded from above, it is basically bounded from below. In [6], a rate-limiter is used to control the slope of the flight size but is rather limited due to the absence of any ACK-flow model and the time-varying nature of the lower bound value. We provide here an explicit approach based on a hybrid modeling of the user behavior.

According to the above discussion, the flight-size must obey

$$\digamma_i(C_i,t)' = \left\{ \begin{array}{ll} \dot{w}_i(t) & \text{if } \mathcal{T}_i(t) \\ -\phi(u_i^-,t) & \text{otherwise} \end{array} \right. \tag{23}$$

where $\mathcal{T}_i(t)$ is a condition which is true when increasing the flight-size is allowed and false otherwise.

Result 12: The flight-size $\digamma_i(C_i, t)$ satisfies (23) if the user sending rate is defined as

$$\phi(u_i^+, t) = \begin{cases} \dot{w}_i(t) + \phi(u_i^-, t) & \text{if } \mathcal{T}_i(t) \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

where $\mathcal{T}_i(t) = \big([\pi_i(t) = 0] \wedge [\dot{w}_i(t) + \phi(u_i^-, t) \geq 0]\big)$ and

$$\dot{\pi}_i(t) = \begin{cases} 0 & \text{if } \mathcal{T}_i(t) \\ \dot{w}_i(t) + \phi(u_i^-, t) & \text{otherwise.} \end{cases} \tag{25}$$

Moreover, this model is the simplest one. ▲

Proof: The virtual buffer π_i, taking nonpositive values, measures the number of ACK packets to retain in order to balance the flight- and window-sizes. When the virtual buffer has negative state, i.e. $\pi_i(t) < 0$, the coming ACK-packets have to be retained until the state reaches 0. Once zero is reached, the user can now start sending again until the window size decreases too fast, i.e. $\dot{w}_i(t) < -\phi(u_i^-, t)$. Substitution of the user sending rate defined by (24) and (25) in (22) yields the flight-size behavior (23). To see that the model is minimal, it is enough to remark that both conditions in $\mathcal{T}_i(t)$ are necessary. ■

The state of the extended user model consists of both the state of the congestion control z_i and the virtual buffer π_i. The user behavior also depends on the measurements $\mu_i(\varkappa_t)$, functions of the overall network state $\varkappa$; this state will be discussed in more details in Section VIII.

We are now in a position to derive the users operators from the equations (24).

Definition 13: The users operators $\mathcal{U}_i(w_i) : \mathbb{R}_+ \to \mathbb{R}_+$ mapping the ACK-flow $\phi(u_i^-, t)$ to the sending flow $\phi(u_i^+, t)$ is given by

$$\phi(u^+, t) = \mathcal{U}(w)\phi(u^-, t) \tag{26}$$

where $\mathcal{U}(w) = \mathrm{diag}_i \{\mathcal{U}_i(w_i)\}$ and $\mathcal{U}_i$ satisfies (7).

VIII. General Network model

We have developed transmission channels, buffers and users models in Sections IV, VI and VII respectively. We summarize in this section the obtained results in a compact form involving dynamical systems and operators:

$$\begin{array}{rcl} \dot{\varkappa}(t) & = & \mathcal{N}(\varkappa_t, \phi(u^-, t), \phi(b^-, t)) \\ \digamma_i(C_i, t) & = & \int_{\mathscr{B}_{C_i}(t)}^t \phi(u_i^+, s) ds \end{array} \tag{27}$$

along with

$$\begin{bmatrix} \phi(u^+, t) \\ \phi(b^+, t) \\ \phi(u^-, t) \\ \phi(b^-, t) \end{bmatrix} = \mathcal{M} \begin{bmatrix} \phi(u^+, t) \\ \phi(b^+, t) \\ \phi(u^-, t) \\ \phi(b^-, t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathcal{D}_b \end{bmatrix} \delta(t)$$

where

$$\mathcal{M} = \begin{bmatrix} 0 & 0 & \mathcal{U}(w) & 0 \\ 0 & 0 & 0 & \mathcal{B} \\ 0 & \mathcal{R}_{ub} & 0 & 0 \\ \mathcal{R}_{bu} & \mathcal{R}_{bb} & 0 & 0 \end{bmatrix} \tag{28}$$

and $\varkappa = \mathrm{col}(\tau, z, \pi)$ is the state of the network. The hybrid users and queues dynamics are described by the nonlinear discontinuous functional $\mathcal{N}$ obtained from equations (4), (5), (7) and (25). The notation $\varkappa_t$ is here to emphasize that the evolution of the network state depends on past state values [12]. Note that adjoining the flight-size expression is critical to obtain a finite number of equilibrium points. Indeed, since the user flow is computed from the derivative of the flight-size, the equilibrium information is lost and can only be recovered from the original expression of the flight-size. At equilibrium we indeed have $\digamma_i^* = w_i^* = RTT_i^* \phi_i^*$ where RTT_i^* and ϕ_i^* are equilibrium values for the RTT and the sending flow of user u_i respectively.

This model thus takes the form of a *descriptor nonlinear hybrid time-delay system with state-dependent and constant delays* about which many theoretical questions are open: well-posedness, existence of solutions, uniqueness of solutions, stability of solutions, etc. Note also that in this paper, we have not discussed about the delays derivative related constraints whose violation may lead to severe well-posedness problems [13]. Some simple topologies have been considered in [11] where it is shown that delay-derivative may exceed one under certain conditions. For the moment, it is unclear whether for arbitrary topologies and under certain reasonable conditions, the delays perceived by the users always have derivative smaller than one.

IX. The Single-Buffer/Multiple-User Topology with FAST-TCP protocol

In this section, we consider a single-buffer/multiple-users topology interconnected by lossless transmission channels. The forward and backward propagation delays of user i are denoted by T_i^f and T_i^b respectively. We propose to use the FAST-TCP model as user model

$$\dot{w}_i(t) = \gamma \left[-\frac{\tau(g_i(t))}{T_i + \tau(g_i(t))} w_i(t) + \alpha \right] \tag{29}$$

where $w_i(t)$, $T_i = T_i^f + T_i^b$ and $g_i(t) = g(t - T_i^b)$ are the window size, the propagation delay and the backward queuing delay respectively.

A. General Model

The general model is given by (27) with (29) and

$$\begin{array}{rcl} \dot{\tau}(t) & = & \begin{cases} c^{-1}\eta(t) + \delta(t) - 1 & \text{if } \mathcal{C}(t) \\ 0 & \text{otherwise} \end{cases} \\ \dot{\pi}_i(t) & = & \begin{cases} 0 & \text{if } \mathcal{T}_i(t) \\ \dot{w}_i(t) + \phi(u_i^-, t) & \text{otherwise.} \end{cases} \\ \digamma_i(t) & = & \int_{g_i(t) - T_i^f}^t \phi(u_i^+, \theta) d\theta \\ \phi(u_i^+, t) & = & \begin{cases} \dot{w}_i(t) + \phi(u_i^-, t) & \text{if } \mathcal{T}_i(t) \\ 0 & \text{otherwise} \end{cases} \\ \phi(u_i^-, t) & = & \begin{cases} \frac{c\phi(u_i^+, g_i(t) - T_i^f)}{c\delta(g_i(t)) + \sum_j \phi(u_j^+, g_i(t) - T_j^f)} & \text{if } \mathcal{C}(g_i(t)) \\ \phi(u_i^+, t - T_i^b - T_i^f) & \text{otherwise} \end{cases} \\ \eta(t) & = & \sum_{i=1}^N \phi(u_i^+, t - T_i^f) \end{array} \tag{30}$$

where $\delta(t)$ denotes the normalized cross-traffic $\delta(t) \in [0, 1)$. The equilibrium point of this model is unique and is given by

$$w_i^* = \alpha\left(1+\frac{T_i}{\tau^*}\right), \quad \tau^* = \frac{N\alpha}{c(1-\delta^*)}$$

where $T_i = T_i^f + T_i^b$, $\delta^* \in [0,1)$ and N are the equilibrium normalized cross-traffic, the propagation delay and the number of users respectively. Note that the equilibrium point of the network is both fair and efficient [14]. In what follows, we will show that this model degenerates into models of the literature when some particular conditions are met.

B. Homogeneous delays and no-cross traffic - The static link model

In the case of homogeneous delays, i.e. $T_i^f = T^f$, $T_i^b = T^b$, $i = 1, \ldots, N$, and absence of cross-traffic, i.e. $\delta \equiv 0$, model (30) reduces to

$$\begin{array}{rcl} \dot{\tau}(t) & = & \left\{\begin{array}{ll} c^{-1}\eta(t) - 1 & \text{if } \mathcal{C}(t) \\ 0 & \text{otherwise} \end{array}\right. \\ \dot{\pi}_i(t) & = & \left\{\begin{array}{ll} 0 & \text{if } \mathcal{T}_i(t) \\ \dot{w}_i(t) + \phi(u_i^-, t) & \text{otherwise.} \end{array}\right. \\ F_i(t) & = & \int_{g_b(t)-T^f}^{t} \phi(u_i^+, \theta)d\theta \\ \phi(u_i^+, t) & = & \left\{\begin{array}{ll} \dot{w}_i(t) + \phi(u_i^-, t) & \text{if } \mathcal{T}_i(t) \\ 0 & \text{otherwise} \end{array}\right. \\ \phi(u_i^-, t) & = & \left\{\begin{array}{ll} \frac{c\phi(u_i^+, g_b(t)-T^f)}{\sum_j \phi(u_j^+, g_b(t)-T^f)} & \text{if } \mathcal{C}(g_b(t)) \\ \phi(u_i^+, t - T^b - T^f) & \text{otherwise} \end{array}\right. \\ \eta(t) & = & \sum_{i=1}^{N} \phi(u_i^+, t - T^f) \\ g_b(t) & = & g(t - T^b). \end{array} \tag{31}$$

Assuming the buffer is always congested (i.e. both $\mathcal{C}(t)$, $\mathcal{C}(g_b(t))$ hold true) and all the users are active (i.e. the $\mathcal{T}_i(t)$'s are true) we obtain

$$\dot{\tau}(t) = c^{-1}\sum_i \dot{w}_i(t - T^f) \tag{32}$$

which is the static-link model. Integrating the above equation from 0 to t we obtain

$$\tau(t) = c^{-1}\sum_i w_i(t - T^f) - \varpi \tag{33}$$

where we assumed $\tau(0) = 0$, $w_i(0) = 0$, $i = 1, \ldots, N$ and ϖ is a constant to be determined. This constant can be determined by considering the equilibrium relationship between the queueing delay and the window sizes and we get $\varpi = T = T^b + T^f$. The static-link model of [2] is then retrieved. Thus, according to the proposed model (30), the static-link model is valid whenever

- the buffers are permanently congested, i.e. $\mathcal{C}(t)$ and $\mathcal{C}(g_b(t))$ hold true;
- the propagation delays are homogeneous, i.e. $T_i^f = T^f$, $T_i^b = T^b$, $i = 1, \ldots, N$;
- the cross-traffic is absent, i.e. $\delta \equiv 0$;
- the users are not in ACK-retaining mode, i.e. $\mathcal{T}_i(t)$ holds true for all $i = 1, \ldots, N$.

It is interesting to note that these conditions are necessary and sufficient for the static-link model validity. The first one ensures that the queue is a bottleneck and is always congested. The second condition makes the sum of the ACK-flow contribution in the user sending flows equal to the queue output capacity, so that only the windows derivatives remain in the delay dynamical model. Note that in presence of heterogeneous delays, the sum may exceed the queue maximal output capacity. The absence of cross-traffic also ensures that only the windows derivatives remain. Finally, if the user is in ACK-retaining mode, then it does not send any flow and the window derivative term disappears from the model.

By substituting the above static-link model (33) into the user model (29) with homogeneous delays. This yields the new model

$$\dot{w}_i(t) = \gamma\left[\left(\frac{cT}{\sum_j w_j(g_b(t) - T^f)} - 1\right)w_i(t) + \alpha\right]. \tag{34}$$

This model is very similar to the ones obtained in previous works [2] at the difference that the delay is state-dependent. A thorough study of the above model is made in Section X.

C. Homogeneous delays and cross-traffic

When a cross-traffic is added to the problem, the overall picture is changed. The cross-traffic will act as a bandwidth limiter both in the networking and control terminology. Indeed, a nonzero $\delta(t)$ will reduce the maximal output capacity c, creating then sort of 'varying-output-capacity' $c(1-\delta(t))$ which will reduce the bandwidth perceived by the users.

Result 14: To see the bandwidth reduction in the control sense, note that the queue model rewrites (in the congested mode)

$$\begin{array}{rcl} \dot{\tau}(t) & = & c^{-1}\sum_i \dot{w}_i(t - T^f) + \delta(t) - \varphi(t) \\ \varphi(t) & = & \dfrac{c\delta(g_{bf}(t))}{c\delta(g_{bf}(t)) + \sum_j \phi_j(u_j^+, g_{bf}(t) - T^f)} \end{array} \tag{35}$$

where $\varphi(t)$ is the output flow at time t corresponding to the cross-traffic and is responsible of the bandwidth reduction.

Proof: Since the buffer is always congested and the users are not in ACK-retain mode, then we have

$$\phi_i(u_i^+, t) = \dot{w}_i(t) + \phi_i(u_i^-, t) \tag{36}$$

and

$$\phi_i(u_i^-, t) = \frac{c\phi_i(u_i^+, g_b(t) - T^f)}{c\delta(g_b(t)) + \sum_j \phi_j(u_j^+, g_b(t) - T^f)}. \tag{37}$$

Substituting the above expressions in the queue model and noting that $\sum_j \phi_j(u_j^-, t - T^f) + \varphi(t) = c$ yields the result. ∎

Assume the network is at equilibrium (constant cross-traffic) and one of the windows increases. Then the corresponding flow grows and then, after one RTT, the value of φ decreases in turn, slowing down the queue response.

X. Stability analysis of a simple topology

We consider the case of homogeneous propagation delays and no cross-traffic described in Section IX-B.

A. Equilibrium point

Proposition 15: The equilibrium point of model (34) is unique and given by

$$\begin{aligned} w^*(N) &= \frac{cT}{N} + \alpha \in (\alpha, cT+\alpha] \\ \tau^*(N) &= \frac{N\alpha}{c} \in [\alpha/c, +\infty) \end{aligned} \tag{38}$$

where $T = T^b + T^f$. ∇

Proof: Simple computations show that the equilibrium point verifies the set of symmetric equations

$$\left(\frac{cT}{\sum_j w_j^*} - 1\right) w_i^* + \alpha = 0, i = 1, \ldots, N. \tag{39}$$

Hence the equilibrium window sizes are identical, i.e. $w_i^* = w^*$, $i = 1, \ldots, N$, and are given by

$$w^* = \frac{cT}{N} + \alpha. \tag{40}$$

The equilibrium value of the queueing delay is easily obtained from (33). ■

B. Local stability analysis

Theorem 16: The network model (34) is locally exponentially stable if one of the following statements hold:

- $T < \tau^*$,
- $T \geq \tau^*$ and $\tau^* + T < \tau_c$ where

$$\tau_c = \frac{1}{\gamma}\sqrt{\frac{N\alpha + cT}{N\alpha - cT}} \arcsin\left(\sqrt{1 - \left(\frac{\tau^*}{T}\right)^2}\right). \tag{41}$$

Proof: The linear model [15] around the equilibrium point determined above is given by

$$\dot{\tilde{w}}(t) = A\tilde{w}(t) + B\tilde{w}(t-h) \tag{42}$$

where $\tilde{w}(t) = w(t) - w^*\mathbb{1}_N$, $h = \tau^* + T$, $T = T^f + T^b$, $A = \text{diag}[-\mu]$, $B = -\theta\mathbb{1}_N\mathbb{1}_N^T$, $\tau^* = Nw^*/c - T$, $\mu = \gamma\dfrac{N\alpha}{N\alpha + cT}$, $\theta = \gamma\dfrac{cT}{N(N\alpha + cT)}$ and $\mathbb{1}_N$ is a N-dimensional column vector with entries equal to 1. The B matrix has rank one, it hence interesting to exploit this structure to simplify the problem analysis. The delay-independent stability of the equilibrium model (42) can be analyzed by looking at the H_∞-norm of the transfer function $H(s) := -\theta\mathbb{1}_N^T(sI - A)^{-1}\mathbb{1}_N$ which is given by $||H||_{H_\infty} = \dfrac{cT}{N\alpha}$. The system is delay-independent stable if and only if $cT < N\alpha$ [16]. If $cT \geq N\alpha$, then the system is not delay-independent stable and the delay acts on stability.

To analyze delay-dependent stability, let us consider the change of variables

$$\begin{aligned} y_1(t) &= \textstyle\sum_k \tilde{w}_k(t) \\ y_{i+1}(t) &= \tilde{w}_{i+1}(t) - \tilde{w}_i(t) \end{aligned} \tag{43}$$

for $i = 1, \ldots, N-1$. The derivatives of the y_i's are given by

$$\begin{aligned} \dot{y}_1(t) &= -\mu y_1(t) - N\theta y_1(t-h) \\ \dot{y}_{i+1} &= -\mu y_{i+1}(t) \end{aligned} \tag{44}$$

for $i = 1, \ldots, N-1$ and we have decoupled the different variables. Since $\mu > 0$, the stability of the last $N-1$ ordinary differential equations is ensured. It is therefore only necessary to analyze the stability of y_1. Using similar results as in [17], it is possible to show that the system (44) is delay-dependent stable provided that $h < \tau_c$ where τ_c is given in (41). ■

It is interesting to note that when the equilibrium delay is larger than the total propagation delay, the equilibrium is locally exponentially stable. Hence, increasing the number of users has a positive effect on stability. Increasing α also has a positive effect on stability.

References

[1] K. Jacobsson, L. Andrew, A. Tang, K. H. Johansson, H. Hjalmarsson, and S. Low, "ACK-clock dynamics: Modeling the interaction between ACK-clock and network," in *27th IEEE Conference on Computer Communications (INFOCOM)*, Phoenix, Arizona, USA, 2008, pp. 181–185.

[2] J. Wang, D. X. Wei, and S. H. Low, "Modelling and stability of FAST TCP," in *28th IEEE Conference on Computer Communications (INFOCOM)*, 2005, pp. 938–948.

[3] C. Hollot, V. Misra, D. Towsley, and W. B. Gong, "A control theoretic analysis of RED," in *20th IEEE Conference on Computer Communications (INFOCOM)*, Tel-Aviv, Israel, 2001, pp. 1510–1519.

[4] S. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE Control Systems Magazine*, pp. 28–43, February 2002.

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

[6] K. Jacobsson, "Dynamic modeling of internet congestion control," Ph.D. dissertation, KTH School of Electrical Engineering, 2008. [Online]. Available: https://eeweb01.ee.kth.se/upload/publications/reports/2008/TRITA-EE_2008_020.pdf

[7] N. Möller, "Window-based congestion control," Doctoral thesis, KTH, Stockholm, Sweden, 2008. [Online]. Available: https://eeweb01.ee.kth.se/upload/publications/reports/2008/TRITA-EE_2008_001.pdf

[8] D. Wei, C. Jin, S. Low, and S. Hedge, "FAST TCP: Motivation, architecture, algorithms, performance," *IEEE/ACM Transactions on Networking*, vol. 14(6), pp. 1246–1259, 2006.

[9] D. V. Lindley, "The theory of queues with a single server," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, pp. 277–289, 1952.

[10] K. Jacobsson, L. Andrew, A. Tang, S. Low, and H. Hjalmarsson, "An improved link model for window flow control and its application to FAST TCP," *IEEE Transactions on Automatic Control*, vol. 54(3), pp. 551–564, 2009.

[11] C. Briat, H. Hjalmarsson, K. H. Johansson, G. Karlsson, U. T. Jönsson, and H. Sandberg, "Nonlinear state-dependent delay modeling and stability analysis of internet congestion control," in *49th IEEE Conference on Decision and Control, Atlanta, USA*, 2010.

[12] J. K.Hale and S. M. V. Lunel, *Introduction to Functional Differential Equations*. New York, USA: Springer-Verlag, 1991.

[13] W. Michiels and E. Verriest, "A look at fast varying and state dependent delays from a system theory point of view," K.U. Leuven," Internal Report, 2011. [Online]. Available: http://www.cs.kuleuven.be/publicaties/rapporten/tw/TW586.abs.html

[14] H. Sandberg, H. Hjalmarsson, U. Jönsson, G. Karlsson, and K. H. Johansson, "On performance limitations of congestion control," in *48th IEEE Conference on Decision and Control*, Shanghai, China, 2009, pp. 5869–5876.

[15] K. L. Cooke and W. Huang, "On the problem of linearization for state-dependent delay differential equations," *Proceedings on the American Mathematical Society*, vol. 124(5), pp. 1417–1426, 1996.

[16] K. Gu, V. Kharitonov, and J. Chen, *Stability of Time-Delay Systems*. Birkhäuser, 2003.

[17] S.-I. Niculescu, *Delay effects on stability. A robust control approach*. Springer-Verlag: Heidelbeg, 2001, vol. 269.