Automatica 48 (2012) 425–429

Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Self-clocking principle for congestion control in the Internet[☆]

Wuhua Hu, Gaoxi Xiao [1]

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 10 May 2010
Received in revised form
14 April 2011
Accepted 27 July 2011
Available online 23 December 2011

Keywords:
Self-clocking principle (SCP)
Congestion control
Internet
TCP
Smith predictor
Delay

ABSTRACT

The self-clocking principle (SCP) of Transmission Control Protocol (TCP) had been analyzed for a network implementing a per-flow buffering scheme. The ideal SCP is yet unknown for the Internet which implements a first-in–first-out buffering scheme. This paper derives an ideal SCP for the Internet by formulating the traffic transmission control as a typical control problem and then solving it by a control-theoretic approach. The ideal SCP reveals the defect of the SCP being deployed in the Internet that it is insufficient to avoid congestion by adjusting the packet effective window based on records of the outstanding packets of a single source; instead outstanding packets from other sources also have to be counted. The ideal SCP also reveals the difficulties of developing and implementing an effective self-clocking scheme for congestion control in the Internet.

1. Introduction

Congestion control provides control and avoids congestion for traffic transmission in the Internet. Transmission Control Protocol (TCP) serves for such a purpose. Since the proposal by Jacobson in 1988 (Jacobson, 1988), TCP has experienced many changes such as TCP Tahoe, TCP Reno, TCP NewReno, TCP SACK, TCP Westwood, etc. (Dukkipati, 2007; Jacobsson, 2008). The evolution of TCP spans in the past two decades and continues today, while new directions of developing congestion control protocols emerge at the meantime (Dukkipati, Kobayashi, Zhang-Shen, & McKeown, 2005; Floyd & Jacobson, 1993; Hollot, Misra, & Towsley, 2002; Hong & Yang, 2007; Hu & Xiao, 2009; Jain & Loguinov, 2007; Katabi, Handley, & Rohrs, 2002; Low & Srikant, 2004; Paganini, Wang, Doyle, & Low, 2005).

As a component of all TCP variants, the self-clocking principle (SCP) plays an important role in enforcing the stability of transmission control (Jacobson, 1988; Jacobsson, 2008). In 1999, Mascolo concluded that the SCP implemented in TCP can be interpreted as a Smith predictor (Mascolo, 1999). This should be the first time that a theoretic understanding of the SCP had been established. Later Mascolo used the Smith principle to model the TCP congestion control algorithm deployed in the Internet (Mascolo, 2006). A key observation is that enforcing the SCP corresponds to implementing a Smith control which contributes to efficient congestion control.

The analyses by Mascolo, however, were based on the assumption that the network implements a per-flow buffering scheme. This limits its applicability in practice since the Internet adopts a first-in–first-out (FIFO) buffering scheme. With such a different buffering scheme, the situation for theoretic analysis and design becomes far more complex: in a per-flow buffering network, the dynamics of different source flows are decoupled and the network can be decomposed into a set of similar single-input–single-output (SISO) subsystems for which congestion control design can be easily carried out; in a FIFO buffering network, by contrast, dynamics of different source flows are coupled and they have collective influences on queues in any buffers. Since the flows may traverse various links and buffers, the dynamics coupling can be very complex. This makes the analysis and design of congestion control very challenging. Situation becomes even more complex while considering the existence of heterogeneous delays of the source flows. A question naturally arises: *what should an ideal SCP look like for a FIFO buffering network, say the Internet*? This paper is motivated to give an answer to this question.

We formulate the traffic transmission control into a multi-input–multi-output (MIMO) control problem with multiple input and feedback delays. Based on this formulation, an ideal SCP is then designed for avoiding congestion. The ideal SCP turns out to look like the SCP being deployed in the Internet but has significant differences. Specifically, to compute the effective window (refer to Section 4 for the definition) or packet sending rate for avoiding congestion, the ideal SCP requires a feedback to each source of the

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Hideaki Ishii, under the direction of Editor Ian R. Petersen.

E-mail addresses: hwh@ntu.edu.sg (W. Hu), egxxiao@ntu.edu.sg (G. Xiao).

[1] Tel.: +65 6790 4552; fax: +65 6793 3318.

0005-1098/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2011.11.007

Table 1
Variables characterizing the elements of a network.

Sources	
	N : the number of sources;
	$x(t) = (x_s(t))_{N\times 1}$: $x_s(t)$ is the transmission rate of source s;
	$f(t) = (f_s(t))_{N\times 1}$: $f_s(t)$ is the feedback received by source s;
Links	
	L : the number of links;
	$c = (c_l)_{L\times 1}$: c_l is the output capacity (or bandwidth) of link l;
	$y = (y_l(t))_{L\times 1}$: $y_l(t)$ is the aggregate input rate at link l;
	$q = (q_l(t))_{L\times 1}$: $q_l(t)$ is the instantaneous queue size accumulated by packets to traverse link l;
	$d = (d_l(t))_{L\times 1}$: $d_l(t)$ is the queuing delay of the new inputing packets to traverse link l;
Sources $\leftrightarrow$ Links	
	$\vec{\tau} = (\vec{\tau}_{sl})_{N\times L}$: $\vec{\tau}_{sl}$ is the forward delay from source s to link l;
	$\overleftarrow{\tau} = (\overleftarrow{\tau}_{sl})_{N\times L}$: $\overleftarrow{\tau}_{sl}$ is the feedback delay from link l to source s;
	$\tau = (\tau_s)_{N\times 1}$: $\tau_s = \vec{\tau}_{sl} + \overleftarrow{\tau}_{sl}$ is the round-trip time (RTT) of source s that traverses link l;
Routing paths	
	$R_{sl} = \begin{cases} 1 \text{ if source } s \text{ traverses link } l; \\ 0 \text{ otherwise;} \end{cases}$
	$R = (R_{sl})_{N\times L}$: the routing matrix.

total available queuing space and the total amount of outstanding packets perceived by the source in its routing path. This is in sharp contrast to the SCP being deployed in the Internet which adjusts the effective window (as corresponds to data sending rate) by using records only of outstanding packets owing to a single source. The new result thus reveals that the deployed SCP is defective and cannot avoid congestion or maintaining stable traffic transmission.

The rest of the paper is organized as follows. A model of the network in the absence of congestion control is established in Section 2. Based on this model, an ideal SCP is designed for congestion control in Section 3, where its implications for application are discussed. The ideal SCP is then compared with the SCP being deployed in the Internet in Section 4, revealing the defect of the current self-clocking scheme. Finally, Section 5 concludes the paper.

2. Network plant

For the purpose of congestion control design, models have been developed for the Internet with multiple sources, heterogeneous delays and an arbitrary topology mainly from optimization standpoints (Low & Srikant, 2004; Paganini et al., 2005; Wydrowski, Andrew, & Zukerman, 2003). Yet no such model has been established from a control-theoretic viewpoint. This section presents a concise network model by a control-theoretic approach.

2.1. Mathematical preliminaries

The superscript ^{T} on a matrix denotes its transpose; $(a_{ij})_{m\times n}$ is a handy notation of an $m \times n$ matrix with $a_{ij} \in \Re$ as its element in the ith row and jth column, $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$.

Definition 1. Let $a_i = (a_{ij})_{1\times n}$ and $b_i = (b_{ij})_{1\times n}$, and matrices $A = (a_1^T, a_2^T, \ldots, a_m^T)^T$ and $B = (b_1^T, b_2^T, \ldots, b_m^T)^T$. The dot (or inner) product of A and B, denoted by $A \bullet B$, is defined as $A \bullet B = (a_1 \bullet b_1, a_2 \bullet b_2, \ldots, a_m \bullet b_m)^T$, where $a_i \bullet b_i$ is the conventional dot product of vectors.

Definition 1 extends dot product of vectors to matrices. Some properties of *dot matrix product* are given as follows.

Lemma 1. *The dot matrix product has the following properties:*

(a) *(positivity)* $A \bullet A \geq 0$ *with equality only for* $A = 0$;
(b) *(commutativity)* $A \bullet B = B \bullet A$;
(c) *(homogeneity)* $(\alpha A) \bullet B = \alpha(A \bullet B)$;
(d) *(distributivity)* $(A + B) \bullet C = A \bullet C + B \bullet C$ *and* $D \bullet (A + B) = D \bullet A + D \bullet B$,

where '$\geq$' means being element-wise larger or equal, and α is a scalar, and A, B, C and D are matrices in appropriate dimensions.

The lemma can easily be proved by the definition. Next, a shifting operation on a vector function is defined.

Definition 2. Let $\tau = (\tau_{ij})_{m\times n}$, and $x(t) = (x_i(t))_{m\times 1}$ be a vector function where $x_i(t) : \Re \mapsto \Re$. The matrix shifting operation on the vector function $x(t)$ with respect to τ is defined as $S(x(t), \tau) = (x_i(t - \tau_{ij}))_{m\times n}$.

The next lemma shows that under a certain condition dot matrix product can *reduce* to conventional matrix product.

Lemma 2. *Let* $A = (a_{ij})_{m\times n}$ *and* $x(t) = (x_i(t))_{n\times 1}$. *Then* $A \bullet S^T(x(t), 0) = Ax(t)$, *where* $S^T(\bullet, \bullet) := (S(\bullet, \bullet))^T$ *and* $0 \in \Re^{n\times m}$.

Proof. Let $a_i := (a_{ij})_{1\times n}$ for $i = 1, 2, \ldots, m$. Then $A = (a_1^T, a_2^T, \ldots, a_m^T)^T$. Therefore

$$\begin{aligned} A \bullet S^T(x(t), 0) &= (a_1^T, a_2^T, \ldots, a_m^T)^T \bullet (x(t), x(t), \ldots, x(t))^T \\ &= \left(a_1 \bullet x^T(t), a_2 \bullet x^T(t), \ldots, a_m \bullet x^T(t)\right)^T \\ &= (a_1^T, a_2^T, \ldots, a_m^T)^T x(t) = Ax(t). \quad \square \end{aligned}$$

2.2. Nude network plant

In the Internet, there are a large number of sources sending data through links connected by routers towards their respective receivers, and the receivers feed acknowledgments back to the sources reversing the paths along which the data was sent. In this process, the physical elements mainly include sources, links, routers and receivers. These elements are characterized by variables as shown in Table 1. The routers are not of the concern as congestion is normally caused by link transmission capacities rather than routers' processing capacities (Mascolo, 1999). And the receivers are ignored for their impacts can be treated separately by flow control as deployed in TCP (Mascolo, 1999).

In the table, $\leftrightarrow$'s denote relations between sources and links. Note that N, L and R can all be time-varying. Their variations are assumed to happen at larger time-scales than the one adopted in analysis. Thus they will be viewed as constants. The delays $\vec{\tau}_{sl}$ and $\overleftarrow{\tau}_{sl}$ will be viewed as constants of propagation delays because the contributing queuing delays are almost eliminated under the designed congestion control. The above assumptions were commonly used, e.g., in Paganini et al. (2005) and Wydrowski et al. (2003).