

Self-clocking Principle for Congestion Control in the Internet

Wuhua Hu, Gaoxi Xiao*

* Corresponding author. Tel: +65 6790 4552; fax: +65 6793 3318; email: egxxiao@ntu.edu.sg
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

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 into 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. Also the ideal SCP reveals the difficulties of developing and implementing an effective self-clocking scheme for congestion control in the Internet.

Key words: self-clocking principle (SCP), congestion control, Internet, TCP, Smith control, delay.

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 last 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; Holot, 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 Sec. 4 for the definition) or packet sending rate for avoiding congestion, the ideal SCP requires a feedback to each source of the 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 part of the paper is organized as follows. A model of the network in the absence of congestion control is established in Sec. 2. Based on this model, an ideal SCP is designed for congestion control in Sec. 3, where its implications for application are discussed. The ideal SCP is then compared with the SCP being deployed in the Internet in Sec. 4, revealing the defect of the current self-clocking scheme. Finally, Sec. 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 stand points (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 with 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 $a_{ij} \in \mathfrak{R}$ as its element in the i th row and j th column, $i = 1, 2, K, m$ and $j = 1, 2, K, n$.

Definition 1. Let $a_i = (a_{ij})_{1 \times n}$ and $b_i = (b_{ij})_{1 \times n}$, and matrices $A = (a_1^T, a_2^T, K, a_m^T)^T$ and $B = (b_1^T, b_2^T, K, b_m^T)^T$. The dot (or inner) product of A and B , denoted by $A \bullet B$, is defined as $A \bullet B = (a_1 \bullet b_1, a_2 \bullet b_2, K, a_m \bullet b_m)^T$, 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 $x_i(t) \in \mathfrak{R}$. 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 certain condition dot matrix product can *reduce* to conventional matrix product.

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

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

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

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 acknowledgements 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). 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 $\frac{L}{c}$ and $\frac{N}{c}$ 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; Wydrowski et al., 2003).

With the above characterizations, physical relations between the network variables can be established:

$$\mathcal{Q}_i(t) = \begin{cases} y_i(t) - c_i, & \text{if } q_i(t) > 0 \text{ or } y_i(t) > c_i \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

$$y(t) = R^T \bullet S^T(x(t), \frac{L}{c}), \quad (2)$$

$$f(t) = R \bullet S^T(q(t), \frac{N}{c}), \quad (3)$$

where $l = 1, 2, K, L$. Equations (1)-(2) model the physical dynamics in a transmission network without congestion control and constitute the nude network plant. Equation (3) models the feedback to each source which measures the network congestion by taking the sum of queue sizes experienced by a source on its route to the destination.

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	
$\bar{\tau} = (\bar{\tau}_{sl})_{N \times L}$: $\bar{\tau}_{sl}$ is the forward delay from source s to link l ;
$\bar{\tau} = (\bar{\tau}_{sl})_{N \times L}$: $\bar{\tau}_{sl}$ is the feedback delay from link l to source s ;
$\tau = (\tau_s)_{N \times 1}$: $\tau_s = \bar{\tau}_{sl} + \bar{\tau}_{sl}$ 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.

3. Congestion control design

With a control-theoretic approach, this section derives an ideal SCP for congestion control in the Internet.

3.1. Analysis

Equations (1)-(2) are the plant to be stabilized, $q(t)$ the variable to be controlled, $x(t)$ the manipulated variable and $f(t)$ the feedback signal. Since $x(t) \in \mathfrak{R}^N$ is the plant input, $q(t) \in \mathfrak{R}^L$ the plant output, $f(t) \in \mathfrak{R}^N$ the plant feedback, $\bar{\tau} \in \mathfrak{R}^{N \times L}$ the control input delay and $\bar{\tau} \in \mathfrak{R}^{N \times L}$ the feedback delay, the control problem described by (1)-(3) is basically a MIMO control problem with multiple input and feedback delays. The objective of congestion control design is to find a control law of specifying $x(t)$ to stabilize $q(t)$ at a setpoint of zero, by utilizing the information of the queue deviation $f(t)$ as conveyed back to the sources.

It may be expected to stabilize the queue $q(t)$ at a positive value rather than zero, as was adopted in many TCP/AQM (active queue management) schemes (Athuraliya, Low, Li, & Yin, 2001; Holot et al., 2002). However, this idea is unfavorable from a control perspective, mainly for two reasons: firstly, to eliminate steady-state queue errors an integral control must be included (Åström & Hägglund, 2005), which makes it difficult to obtain a control law guaranteeing the stability of a network with heterogeneous delays (Balakrishnan, Dukkipati, McKeown, & Tomlin, 2005; Gu, Kharitonov, & Chen, 2003; Jain & Loguinov, 2007); secondly, the additional queuing delays are undesirable in application (Low & Srikant, 2004).

An integral control makes it difficult to achieve assured stability of a network with heterogeneous delays. But without using an integral control, the actual queue length can drift from zero in the steady state due to any disturbances such as cross traffic from uncontrolled sources, leading to varying queuing delays. To eliminate such drift, an alternate solution is to reserve a fraction of the link capacity (Paganini et al., 2005; Wydrowski et al., 2003). The main idea is to replace the true capacity c_l by a *virtual capacity* $\gamma_l c_l$, where $0 < \gamma_l < 1$. This leads to the *virtual queue* dynamics:

$$\dot{q}_l(t) = \begin{cases} y_l(t) - \gamma_l c_l, & \text{if } q_l(t) > 0 \text{ or } y_l(t) > \gamma_l c_l \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

In equilibrium, the link utilization is γ_l and there is a spare bandwidth of $(1 - \gamma_l)c_l$ to dispose any emerging queue. Thus if (4) is stabilized such that $y_l(t) \rightarrow \gamma_l c_l$ as $t \rightarrow \infty$, having empty actual queues at links can be guaranteed. Hereafter the *virtual network plant*, comprised of (2)-(4), will be used to carry out the congestion control design for achieving the required control.

Note that even though the queues are empty at the equilibrium, proper buffer sizes are needed in practice to handle the transient queues. There have been a lot of research on setting efficient buffer sizes, as can be referred to (Raina, Towsley, & Wischik, 2005; Raina & Wischik, 2005; Voice & Raina, 2009; Wischik & McKeown, 2005) and the references therein.

3.2. SCP design

Suppose there are small perturbations around the equilibrium such that $x(t) = x^* + \delta x(t)$, $y(t) = y^* + \delta y(t)$, $q(t) = q^* + \delta q(t)$ and $f(t) = f^* + \delta f(t)$. At equilibrium, $y^* = (\gamma_l c_l)_{L \times 1} < c$ (element-wise comparison) and thus there are no queuing delays, implying that $\bar{\tau}$ and $\bar{\tau}$ are constant matrices. Note that non-bottleneck links maintain zero virtual queues if perturbations are sufficiently small, while all bottleneck links are affected and the virtual queues are changed. It means that $\delta q(t)$ is nonzero only for bottleneck links and the analysis can be focused on

such links. With these considerations, denote $\delta\bar{y}(t)$ and $\delta\bar{q}(t)$ as the reduced vectors, \bar{R} the reduced routing matrix, $\underline{\tau}$ and $\underline{\xi}$ and $\bar{\tau}$ respectively the reduced forward delay and backward delay and RTT matrices, by removing columns corresponding to non-bottleneck links. Let the bottleneck links be ticked as $l=1, 2, K, \bar{L}$. Linearizing (1)-(3) around a nonzero virtual queue size gives

$$\delta\bar{y}(t) = \bar{R}^T \bullet S^T(\delta x(t), \underline{\tau}), \quad (5)$$

$$\dot{\delta\bar{q}}(t) = \delta\bar{y}(t), \quad (6)$$

$$\delta f(t) = \bar{R} \bullet S^T(\delta\bar{q}(t), \underline{\xi}^T). \quad (7)$$

The main result is stated in the following theorem.

Theorem 1. *Assume that \bar{R} has full column rank and that the approximation*

$$\begin{aligned} & S^T\left(\bar{R}^T \bullet \int_0^t S^T(\delta x(v), \underline{\tau}) dv, \underline{\xi}^T\right) \\ & \approx S^T\left(\bar{R}^T \bullet \int_0^t S^T(\delta x(v), \bar{\tau}) dv, 0\right) \end{aligned} \quad (8)$$

is valid (within tolerance of the system stability margin). Let

$$K'' := \bar{R}(\bar{R}^T \bar{R})^{-1} K'(\bar{R}^T \bar{R})^{-1} \bar{R}^T, \quad (9)$$

where $K' = \text{diag}\{k'_l\} \in \mathfrak{R}^{\bar{L} \times \bar{L}}$ and $k'_l > 0$ for $l=1, 2, K, \bar{L}$. Then the control law

$$\delta x(t) = K''(-\delta f(t) - \bar{R}Q(t, \bar{\tau})), \quad (10)$$

where

$$\begin{aligned} & -\delta f(t) - \bar{R}Q(t, \bar{\tau}) = -\delta f(t) + \bar{R}\left(\bar{R}^T \bullet \int_0^t S^T(\delta x(v), \bar{\tau}) dv\right) - \bar{R}\left(\bar{R}^T \bullet \int_0^t S^T(\delta x(v), 0) dv\right) \\ & = -\delta f(t) + \bar{R} \bullet S^T\left(\bar{R}^T \bullet \int_0^t S^T(\delta x(v), \bar{\tau}) dv, 0\right) - \bar{R}\left(\bar{R}^T \bullet \int_0^t S^T(\delta x(v), 0) dv\right) \quad (\text{Lemma 2}) \\ & = -\delta f(t) + \bar{R} \bullet S^T\left(\bar{R}^T \bullet \int_0^t S^T(\delta x(v), \underline{\tau}) dv, \underline{\xi}^T\right) - \bar{R}\left(\bar{R}^T \bullet \int_0^t S^T(\delta x(v), 0) dv\right) \quad (\text{Eq. (8)}) \\ & = \bar{R}\left(-\bar{R}^T \bullet \int_0^t S^T(\delta x(v), \underline{\tau}) dv - \bar{R}^T \bullet \int_0^t (S^T(\delta x(v), 0) - S^T(\delta x(v), \underline{\tau})) dv\right) \\ & = \bar{R}(-\delta\bar{q}(t) - Q(t, \underline{\tau})) = \bar{R}\delta\vartheta(t). \end{aligned} \quad (16)$$

In the control law (10), the term $-\delta f(t)$ has a size of $N \nexists 1$ and each of its elements denotes free virtual queuing space (relative to the equilibrium size of a virtual queue) that a source observes in its routing path. Similarly, each element of the vector $Q(t, \bar{\tau})$, in a size of $\bar{L} \nexists 1$, means the amount of outstanding packets in a link (i.e., the packets moving in a link). Hence the row vector $\bar{R}Q(t, \bar{\tau})$ has each of its elements standing for the total amount of outstanding packets that a source observes in its routing path. Therefore each element of $-\delta f(t) - \bar{R}Q(t, \bar{\tau})$ indi-

$$Q(t, \bar{\tau}) = \bar{R}^T \bullet \int_0^t (S^T(\delta x(v), 0) - S^T(\delta x(v), \bar{\tau})) dv, \quad (11)$$

stabilizes the linearized system consisting of (5)-(7).

Proof. Let $\delta\vartheta(t) := -\delta\bar{q}(t) - Q(t, \underline{\tau})$, where

$$Q(t, \underline{\tau}) := \bar{R}^T \bullet \int_0^t (S^T(\delta x(v), 0) - S^T(\delta x(v), \underline{\tau})) dv. \quad (12)$$

The above definition of $Q(t, \underline{\tau})$ is in analog to the transformation used in the reduction method for handling time-delay systems (Artstein, 1982; Moon, Park, & Kwon, 2001). With $\delta\bar{q}(t)$ given in (6), it follows that

$$\dot{\delta\vartheta}(t) = -\bar{R}^T \delta x(t). \quad (13)$$

If $\delta x(t)$ is given in (10), then (13) specifically becomes

$$\dot{\delta\vartheta}(t) = -K'(\bar{R}^T \bar{R})^{-1} \bar{R}^T (-\delta f(t) - \bar{R}Q(t, \bar{\tau})). \quad (14)$$

Since $-\delta f(t) - \bar{R}Q(t, \bar{\tau}) = \bar{R}\delta\vartheta(t)$, as refers to (16), equation (14) is equivalent to

$$\dot{\delta\vartheta}(t) = -K'\delta\vartheta(t). \quad (15)$$

With $K' = \text{diag}\{k'_l\} \in \mathfrak{R}^{\bar{L} \times \bar{L}}$ and $k'_l > 0$ for $l=1, 2, K, \bar{L}$, it is obvious that $\delta\vartheta(t)$ and $\dot{\delta\vartheta}(t)$ converge to zero as t approaches infinity. Note that $\dot{\delta\vartheta}(t) \rightarrow 0$ implies $\delta x(t) \rightarrow 0$ and consequently $Q(t, \underline{\tau}) \rightarrow 0$. Together with the fact that $\delta\vartheta(t) \rightarrow 0$, it follows that $\delta\bar{q}(t) \rightarrow 0$ as $t \in \mathbb{R}^+$, i.e., the linearized system consisting of (5)-(7) is stabilized. \square

cates the maximum number of packets allowed to be injected into the network for the respective source without causing queue inflation above the equilibrium size, which can be interpreted as an effective window advertised for a source in analog to that implemented in TCP. Thus control law (10) suggests an effective window with proper scaling for each source in order to avoid congestion and ensure stable traffic transmission in the network.

In fact, control law (10) is in the very merit of the ‘self-clocking principle’ as initiated by Jacobson for congestion control in the Internet, which aims to enforce ‘conservation of packets’ that ‘a new packet isn’t put into the network until

an old packet leaves' (Jacobson, 1988). For convenience, we may call control law (10) an *ideal* SCP, differing from the SCP being deployed in TCP. As the term $\bar{R}Q(t, \bar{\tau})$ in (10) may alternatively be interpreted as the number of packets to enter into the bottleneck links as predicted by the sources, control law (10) embodies the merit of Smith predictor (or Smith control) and hence can be viewed as that for a MIMO plant as modeled by (5)-(7).

The ideal SCP requires each source to know the routing matrix of all sources, the free virtual queuing space and the outstanding packets in its individual routing path, which may all be restricted to bottleneck links. Among such information, the routing matrix would be most difficult to obtain due to rather complex distribution and time-varying nature of the sources in the Internet. Together with the extra cost of communicating the information between the sources, it imposes great difficulty for each source to know the amount of outstanding packets in its own routing path. (Note that the RTT matrix $\bar{\tau}$ may not be required because the amount of outstanding packets can usually be estimated by the records of unacknowledged packets at the sources (Jacobson, 1988; Mascolo, 1999), which avoids the rigid mathematical integration involving the RTT matrix $\bar{\tau}$.) To resolve the difficulty, novel strategies have to be developed to observe the outstanding packets in the links, which will restrict the inaccurate routing matrix to affect only the control gain. By contrast, it is likely for a source to know the free virtual queuing space in its routing path if the routers are able to embed such information into the packets that it is conveyed back to the source by the acknowledgement packets. The above insights gained from the ideal SCP are consistent with the intuition.

Remark 1. The assumption of \bar{R} having full column rank means that there are no algebraic constraints between bottleneck link flows, which rules out, for instance, the situation where all flows going through one bottleneck link also go through another. Typically, in that situation only one of the links would be a bottleneck. Therefore the assumption is generic (Paganini et al., 2005).

Remark 2. The assumption in (8) is necessary for an analytic solution to the design problem. Equation (8) means the various forward delays are approximated by the forward delay from a particular source to the bottleneck link in consideration. The approximation is valid if the forwards delays are not vastly distributed. In the special case of a per-flow buffering network, the approximation becomes exact and the SCP given in Theorem 1 recovers the SCP derived in (Mascolo, 1999, 2006).

Remark 3. We have also considered deriving an ideal SCP for Max-Net, in which the feedback signal $f(t)$ conveys the maximal (instead of the sum as in (3)) queue deviation experienced by a source on its route to the destination (Suchara et al., 2008; Wydrowski et al., 2003; Wydrowski & Zukerman, 2002). But the analysis shows that such an SCP may not exist due to the nonlinear operation of taking the maximal.

4. Discussion

Control law (10) gives an ideal SCP for congestion control in a network with FIFO buffering, which extends the results on a per-flow buffering network as reported in (Mascolo, 1999, 2006). To see the difference, we compare as follows the SCP's for FIFO and per-flow buffering networks. A generic TCP flow in a per-flow buffering network is considered for comparison.

Each TCP flow maintains two variables (Mascolo, 1999): the *Advertised Window* (AW), which measures the congestion status of the receiver buffer and the *Congestion Window* (CW) which measures the congestion status of the network. These two window variables are used to calculate the *Effective Window* (EW) which limits how much outstanding data the source can send:

$$\begin{aligned} EW_s &= \min(AW_s, CW_s) - (LastByteSent_s - LastByteAked_s) \\ &= \min(AW_s, CW_s) - OutstandingPackets_s \\ &= \min(AW_s, CW_s) - \int_{t-\tau_s}^t x_s(v) dv. \end{aligned} \quad (17)$$

The subscript $_s$ means that the variable is associated with source s . Note that the outstanding packets measure those from a single source only. Such a mechanism characterizes a Smith predictor or SCP for avoiding congestion in a per-flow buffering network (Mascolo, 1999, 2006).

By contrast, the rate of a TCP flow in a FIFO buffering network is computed as follows

$$\begin{aligned} x(t) &= K''(-f(t) - RQ(t, \tau)) \\ &= K''\left(-f(t) - R\left(R^T \bullet \int_0^t (S^T(x(v), 0) - S^T(x(v), \tau)) dv\right)\right). \end{aligned} \quad (18)$$

Here the outstanding packets $RQ(t, \tau)$ include contributions from various sources rather than a single source. This indicates the defect of the SCP being deployed in TCP. It also reveals that the conclusion made in (Mascolo, 1999, 2006) that TCP implements a Smith control misses the fact that the control law (17) actually cannot guarantee the stability of congestion control in the Internet where FIFO buffering is used. Meanwhile the ideal SCP in (18) indicates that for ideal congestion control, each source needs to know the total outstanding packets in its own routing path. This imposes an essential condition on its implementation.

Lastly, note that $\min(AW_s, CW_s)$ in (17) may be perceived as an estimate of the available queuing space $-f(t)$ in (18). The accuracy of such estimation depends on the congestion window size CW_s and hence the congestion control algorithm adopted. The algorithms vary in different variants of TCP.

5. Conclusion

Congestion control in the Internet was formulated as a MIMO control problem with multiple input and feedback delays. Based on this formulation, an ideal SCP was derived for congestion control, showing that the SCP for a

per-flow buffering network does not apply to the one with FIFO buffering. The ideal SCP indicates the SCP being deployed in the Internet cannot avoid congestion or maintain stable traffic transmission 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 indicates that for ideal congestion control, each source needs to know the routing matrix of all sources, the free virtual queuing space and the outstanding packets in its own routing path. These consist of three essential conditions on implementation of such an ideal SCP. As the current strategy is limited to estimating the outstanding packets owing to a single source, novel strategies are demanded to resolve the difficulty of estimating the total outstanding packets that a source observes in its routing path.

Meanwhile the analysis shows that it is hard to obtain a general ideal SCP in the absence of the assumption in (8), and that an exact implementation of an ideal SCP may require strong conditions. Nevertheless, by referring to the marvelous success of the SCP being deployed, certain special form or approximate implementation of an ideal SCP may be explored in the future to enhance congestion control in the Internet.

References

- Artstein, Z. (1982). Linear systems with delayed controls: A reduction. *IEEE Transactions on Automatic Control*, 27(4), 869-879.
- Åström, K.J. & Hägglund, T. (2005). *Advanced PID Control*. Research Triangle Park, NC: ISA-The Instrumentation, Systems, and Automation Society.
- Athuraliya, S., Low, S.H., Li, V.H., & Yin, Q. (2001). REM: Active queue management. *IEEE Network*, 15(3), 48-53.
- Balakrishnan, H., Dukkkipati, N., McKeown, N., & Tomlin, C.J. (2005). Stability analysis of explicit congestion control protocols. Technical Report, SUDAAR 776, Stanford University
- Dukkkipati, N. (2007). *Rate control protocol (RCP): congestion control to make flows complete quickly*. Ph.D. dissertation, Stanford University, California.
- Dukkkipati, N., Kobayashi, M., Zhang-Shen, R., & McKeown, N. (2005). Processor sharing flows in the Internet. In *Proc. of Thirteenth International Workshop on Quality of Service (IWQoS)* (pp. 271-185).
- Floyd, S. & Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1(4), 397-413.
- Gu, K., Kharitonov, V., & Chen, J. (2003). *Stability of Time-Delay Systems*. Boston: Birkhäuser.
- Hollot, C.V., Misra, V., & Towsley, D. (2002). Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Trans. on Automatic Control*, 47(6), 945-959.
- Hong, Y. & Yang, O.W.W. (2007). Design of adaptive PI rate controller for best-effort traffic in the Internet based on phase margin. *IEEE Trans. on Parallel and Distributed Systems*, 18(4), 550-561.
- Hu, W. & Xiao, G. (2009). Design of congestion control based on instantaneous queue sizes in the routers. In *Proc. of IEEE GLOBECOM* (pp. 1-6).
- Jacobson, V. (1988). Congestion avoidance and control. In *Proc. of ACM SIGCOMM '88* (pp. 314-329).
- Jacobsson, K. (2008). *Dynamic modeling of Internet congestion control*. Ph.D. dissertation, Royal Institute of Technology, Stockholm, Sweden.
- Jain, S. & Loguinov, D. (2007). PIQI-RCP: Design and Analysis of Rate-Based Explicit Congestion Control. In *Proc. of Fifteenth IEEE International Workshop on Quality of Service (IWQoS)* (pp. 10-20).
- Katabi, D., Handley, M., & Rohrs, C. (2002). Congestion control for high bandwidth-delay product networks. In *Proc. of ACM SIGCOMM* (pp. 89-102).
- Low, S.H. & Srikant, R. (2004). A mathematical framework for designing a low-loss, low-delay internet. *Networks and Spatial Economics*, 4(1), 75-101.
- Mascolo, S. (1999). Congestion control in high-speed communication networks using the Smith principle. *Automatica*, 35(12), 1921-1935.
- Mascolo, S. (2006). Modeling the Internet congestion control using a Smith controller with input shaping. *Control Engineering Practice*, 14(4), 425-435.
- Moon, Y.S., Park, P.G., & Kwon, W.H. (2001). Robust stabilization of uncertain input-delayed systems using reduction method. *Automatica*, 37(2), 307-312.
- Paganini, F., Wang, Z., Doyle, J.C., & Low, S.H. (2005). Congestion control for high performance, stability, and fairness in general networks. *IEEE/ACM Transactions on Networking (TON)*, 13(1), 43-56.
- Raina, G., Towsley, D., & Wischik, D. (2005). Part II: Control theory for buffer sizing. *ACM SIGCOMM Computer Communication Review*, 35(3), 79-82.
- Raina, G. & Wischik, D. (2005). Buffer sizes for large multiplexers: TCP queueing theory and instability analysis. In *Proc. of EuroNGI Conference on Next Generation Internet Networks* (pp. 173-180).
- Suchara, M., Andrew, L.L.H., Witt, R., Jacobsson, K., Wyrowski, B.P., & Low, S.H. (2008). Implementation of provably stable MaxNet. In *Proc. of Broadnets* (pp. 8-11).
- Voice, T. & Raina, G. (2009). Stability analysis of a max-min fair Rate Control Protocol (RCP) in a small buffer regime. *IEEE Trans. on Automatic Control*, 54(8), 1908-1913.
- Wischik, D. & McKeown, N. (2005). Part I: Buffer sizes for core routers. *ACM SIGCOMM Computer Communication Review*, 35(3), 75-78.
- Wyrowski, B., Andrew, L.L.H., & Zukerman, M. (2003). MaxNet: A congestion control architecture for scalable networks. *IEEE Communications Letters*, 7(10), 511-513.
- Wyrowski, B. & Zukerman, M. (2002). MaxNet: a congestion control architecture. *IEEE Communications Letters*, 6(11), 512-514.