Active Queue Management Controller for the High Speed TCP Protocol

Juliana de Santi  
State University of Campinas  
Campinas, Brazil  
Email: juliana.santi@students.ic.unicamp.br

Nelson L. S. Fonseca  
State University of Campinas  
Campinas, Brazil  
Email: nfonseca@ic.unicamp.br

Michele M. A. E. Lima  
Federal University of Pernambuco  
Caruaru, Brazil  
E-mail: michelelima@ufpe.br

Abstract—This paper introduces the HSTCP-H2 AQM controller, an optimal AQM controller for networks which employ the HSTCP protocol as their transport protocol. The synthesis of the controller uses a non-rational approach, in which stability and performance objectives are completely expressed as Linear Matrix Inequalities (LMIs). Results, derived via simulation, show the advantages of adopting HSTCP-H2 rather than RED in high capacity networks.

I. INTRODUCTION

In response to a series of collapses due to congestion on the Internet in the mid-'80s, congestion control was added to the transmission control protocol (TCP), thus allowing individual connections to control the amount of traffic they inject into the network. This control involves regulating the size of the congestion window (cwnd) to impose a limit on the size of the transmission window. In the most deployed TCP variant on the Internet, TCP Reno, changes in congestion window size are driven by the loss of segments. Congestion window size is increased by $\frac{1}{cwnd}$ for each acknowledgement (ack) received, and reduced to half for the loss of a segment in a pattern known as additive increase multiplicative decrease (AIMD).

Although this congestion control mechanism was derived at a time when the line speed was of the order of 56 kbs, it has performed remarkably well given that the speed, size, load, and connectivity of the Internet have increased by approximately six orders of magnitude in the past 15 years.

In long distance and high-speed networks, very large window sizes are required for the complete utilization of the transmission pipe. The window size should be roughly equal to the bandwidth delay product. The conservative approach of TCP Reno congestion control restrains the speed of growth of the congestion window, thus, preventing efficient use of the high bandwidth available. Moreover, the drastic reduction in window size after each loss event prolongs the process of achieving an efficient window size. Therefore, numerous TCP variants for high-speed networks have recently been proposed.

High Speed TCP (HSTCP) [1], one of the first variants proposed, operates in two different modes. In scenarios with loss rate higher than $10^{-3}$, HSTCP operates like TCP Reno. When congestion events are rare, it adopts a more aggressive window growth function to scale to the available bandwidth. These parameters of HSTCP are set in a way that establishes a linear relationship in a log-log scale between the sending rate and that of congestion events.

Although powerful and necessary to prevent network collapse, the congestion control mechanism of TCP Reno and TCP variants is not sufficient to avoid network congestion due to the limited control that TCP sources exert in the aggregate network traffic in the presence of unresponsive flows, since these flows do not slow down their sending rates under congestion. The efficacy of TCP congestion control mechanism relies on other mechanisms, such as Active Queue Management (AQM) to avoid congestion. Routers with AQM mechanisms notify TCP senders of incipient congestion by dropping/mark packets so that these senders can reduce their transmission rate before queue overflows and sustained packet loss occur.

Random Early Detection policy (RED) [2] is the AQM policy recommended by the Internet Engineering Task Force (IETF) for deployment in the Internet. RED estimates the average queue length, and compare it to two thresholds: $min_h$ and $max_h$. If the average queue length is less than the $min_h$, no packet is marked or dropped. If it is between $min_h$ and $max_h$, each arriving packet is marked or dropped with a certain probability value $p_d$, which increases linearly with the average queue length. Otherwise, every arriving packet is dropped.

Setting RED parameters is a major challenge. When threshold values are not correctly defined, RED can perform even worst than the traditional tail drop policy. To overcome the difficulties of tuning RED parameters, numerous studies based on heuristics have been conducted. Nonetheless, these studies neither assure that an equilibrium point can be reached nor guarantee the stability of queue length.

Research has been conduct in an attempt to derive RED configuration in a more systematic way. One of these attempts uses Control Theory to design AQM policies that ensure stability about an equilibrium point. Policies based on Control Theory consider the intrinsic feedback nature of network congestion. The transmission rates of TCP senders are adjusted according to the level of congestion, which is determined by queue occupancy. The notification of congestion to the sources is pursued via packet dropping/marking. Therefore, controllers are responsible for determining the appropriate value of drop/mark probability values to stabilize the queue.
length independent of network conditions.

This paper presents a new AQM policy, HSTCP-H2 for networks which adopt the High Speed TCP protocol as their transport protocol. The design employs optimal controllers instead of classical controllers, which are frequently sensible to model uncertainties. In the optimal control strategy the desired system behavior is formulated in terms of a cost function that is minimized. The use of an objective function allows the designer to specify exactly the control objectives, that once determined, are optimally met.

The novelty of the proposed approach lies in the use of non-rational controllers to prove stability with respect to the delay component of the system [3]. It is well-known that delay independent control presents performance limitations in the presence of long delays [4]. In the network context, resources can be more efficiently utilized if a delay-dependent approach is used. Moreover, the use of a non-rational also overcomes the difficulty involved in incorporating the matrix multiplier into the problem to prove stability with respect to the delayed component of the system as this is the major challenge in designing a rational controller for linear delay systems [3]. Furthermore, in the proposed approach stability and performance objectives are completely expressed as Linear Matrix Inequalities (LMIs), thus requiring the solution of a single convex problem for the computation of the controller parameters, leading to a very low computational cost.

The plant used in this design represents the system in greater detail, and also takes into account the presence of non-adaptive traffic such as UDP. The advantage of representing greater detail, and also takes into account the presence of non-adaptive traffic such as UDP. The advantage of representing greater detail, and also takes into account the presence of non-adaptive traffic such as UDP. The advantage of representing greater detail, and also takes into account the presence of non-adaptive traffic such as UDP.

The setting of HSTCP parameters follows the classical AIMD approach, but with different increase and decrease rate values. The HSTCP congestion window dynamics is given by:

\[ \text{ACK} : W = W + \frac{a}{W}; \quad \text{DROP} : W = W - b \times W \]

where the additive increase \((a)\) and multiplicative decrease \((b)\) parameters are function of the current congestion window size and are given by [1] [8]:

\[ b = -0.12\log(W) + 0.69; \quad a = 0.16W^{0.8} \frac{b}{2-b} \]

III. HSTCP/AQM DYNAMIC MODEL

A dynamic fluid-flow stochastic differential equation model, which captures the behavior of both HSTCP window and queue size variation was introduced in [6]; it is:

\[ W(t) = \frac{a(t)}{R(t)} - b(t)W(t)\left(W(t - R(t))\frac{a(t)}{R(t)} - W(t); \right) \]

\[ q(t) = \frac{N(t)W(t)}{R(t)} - C + \omega_q(t); \]

\[ R(t) = \frac{q(t)}{C(t)} + T_p; \]

where:

- \(W(t)\) is the mean HSTCP window size in packets;
- \(q(t)\) is the queue length in packets;
- \(R(t)\) is the round trip time (RTT) in seconds;
- \(a(t)\) is the window size increase parameter;
- \(b(t)\) is the window size decrease parameter;
- \(p(t)\) is the packet mark/drop probability;
- \(N(t)\) is the number of TCP connections;
- \(C(t)\) is the link capacity in packets/second;
- \(\omega_q(t)\) is the noise produced by UDP flows;
- \(T_p\) is the propagation delay in seconds;

Equation (3) describes the TCP window dynamics. The first term models the window additive increase, while the second term the multiplicative decrease.

Equation (4) expresses the queue size variation as the difference between the arrival rate, \(NW/R + \omega_q(t),\) and the link capacity, \(C.\) In equation (4), the term \(\omega_q(t)\) accounts the contribution of UDP flows to the queue size. UDP flows are non-adaptive, which means that they do not reduce their transmission rate under congestion.

Let \((W, q)\) be the system state to be controlled (Equations (3-5)), and \(p\) be the input of the system. The equilibrium point of this system, denoted by \((W_0, q_0, p_0)\), can be computed by making \(W(t) = 0, q(t) = 0,\) and \(R_0 = \frac{NW_0}{C_0} + T_p.\) Thus, the equilibrium point is given by:

\[ W_0 = \sqrt{\frac{a_0}{b_0}} = \frac{R_0C_0}{N_0} = \frac{C_0T + q_0}{N_0}; \]

\[ q_0 = C_0(R_0 - T) = N_0\sqrt{\frac{a_0}{b_0p_0}} - C_0T; \]

\[ p_0 = \frac{a_0N_0^2}{b_0(R_0C_0)^2} = \frac{a_0N_0^2}{b_0(C_0T + q_0)^2}; \]

where \(N(t) \equiv N_0\) and \(C(t) \equiv C_0.\) If equations (3) and (4) are linearized about the equilibrium point, resulting on:

\[ \dot{x}_1(t) = -\frac{a_0N_0}{R_0C_0} (x_1(t) + x_1(t - R_0)) - \frac{b_0C_0^2R_0}{N_0}; \]

\[ \dot{x}_2(t) = \frac{N_0}{R_0} x_1(t) - \frac{1}{R_0} x_2(t); \]
where:
\[ x_1(t) = W(t) - W_0; \quad x_2(t) = q(t) - q_0; \quad u(t) = p(t) - p_0; \]

IV. DESIGN OF AN OPTIMAL CONTROLLER FOR ACTIVE QUEUE MANAGEMENT

In this section, the congestion control system (9) is represented as a continuous time linear delay system in state space form and a non-rational approach is used to derive the HSTCP-H₂ optimal controller. The synthesis of the controller follows the approach introduced in [3] [9].

The HSTCP dynamics in system (9) can be analyzed as a function of network parameters such as the number of TCP flows, \( N_0 \), the round trip time (RTT), \( R_0 \), the link capacity, \( C_0 \), as well as in terms of the intrinsic feedback nature of AQM [6]. The action of the AQM controller, \( C(s) \), is to mark/drop packets with probability value \( p \), using the measured queue size \( q \). It should also stabilize the plant of the system, denoted by the transfer function \( P(s) \), which is irrational in \( s \) and relates the mark/drop probability affects the queue size. The linear system (9) can be expressed in state space form by the following equations:

\[
\begin{align*}
\dot{x}(t) &= A_0 x(t) + A_1 x(t - R_0) + B_w w(t); \\
z(t) &= C_0 x(t) + C_1 x(t - R_0) + D_{wz} w(t); \\
y(t) &= C_y x(t - R_0) + D_{yw} w(t);
\end{align*}
\]

(10)

where \( x(t) \) is the state vector; \( u(t) \) is the control input that represents the probability \( p(t) \); \( w(t) \) is the external noise produced by UDP loads; \( z(t) \) is the reference output, i.e., the desired output for the system and \( y(t) \) is the measured output. For feedback purposes the delay in \( u(t - R_0) \) of (9) is expressed in the output \( y(t) \) of (10).

Now, consider that the system described in (10) is connected to the following controller:

\[
\begin{align*}
\dot{x}(t) &= A_0 \hat{x}(t) + A_1 \hat{x}(t - R_0) + B_g g(t); \\
u(t) &= C_0 \hat{x}(t) + C_1 \hat{x}(t - R_0) + D_g g(t);
\end{align*}
\]

(11)

This controller can also be described in the frequency domain by the non-rational transfer function:

\[
C_{HSTCP-{H}_2}(s) = \frac{(sI - A_0 - A_1 e^{-sR_0})^{-1} e^{s(C_0 + C_1 e^{-sR_0})B} + D}{sI - A_0 - A_1 e^{-sR_0}B + D};
\]

(12)

The controller (11) was carefully designed to reproduce the structure of the plant of system (10). The goal is to determine the matrices of the controller (11) that stabilizes (10) while minimizing a certain measure of the reference output \( z(t) \). To achieve this design goal it is necessary to define the desired performance goals for the output \( z(t) \) as well as what should be measured in the output \( y(t) \).

The ideal mark/drop probability value should assure maximum transmission rates while minimizing the queue length subject to the network conditions, so that loss of packets are avoided. To achieve such goal, the matrices of system (10) are defined as:

\[
A_0 = \begin{bmatrix} -\frac{a_0 N_0}{N_0^2} & -\frac{a_0}{N_0} & 0 \\ -\frac{N_0 a_0}{N_0^2} & -\frac{N_0 - 1}{N_0} & 0 \\ -\frac{N_0}{N_0^2} & -\frac{1}{N_0} & 0 \end{bmatrix}, \quad B_w = \begin{bmatrix} 0 \\ 0.2C_0 \\ 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} -\frac{a_0 N_0}{N_0^2} & 0 \\ 0 & -\frac{N_0}{N_0^2} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
B_u = \begin{bmatrix} -\frac{N_0 a_0 C_0 N_0}{N_0^2} \\ 0 \\ 0 \end{bmatrix}, \quad B_u = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix}, \quad C_z = \begin{bmatrix} 0 & 0 & 2 \end{bmatrix}, \quad C_y = \begin{bmatrix} 0 & 1 \end{bmatrix};
\]

Matrices \( A_0, A_1 \) and \( B_u \) are obtained directly from the linearization of the system. \( A_0 \) represents the terms of the system without delay, while matrix \( A_1 \) represents the delay term. The first row of these matrices corresponds to \( \frac{\partial f_1}{\partial x} \), and the second row corresponds to \( \frac{\partial f_2}{\partial x} \), for \( i = 1, 2 \). \( B_u \) contains \( \frac{\partial f_1}{\partial y} \) in the first row, and \( \frac{\partial f_2}{\partial y} \) in the second row, where \( f_1 \) is the right-hand of Equation (3) and \( f_2 \) the right-hand of Equation (4), \( x_1 = W, x_2 = q \) and \( u = p \).

The noise existing in the system, generated by UDP flows, is controlled by \( B_u \). The chosen value, 0.2\( C_0 \), allows UDP flows to utilize up to 20% of the link capacity.

\( C_y \) translates the design goal, which is to avoid link under-utilization while minimizing the queue size and its variation. The first row is related to the queue size, and it expresses the aim of minimizing the distance between \( q \) measured and \( q_0 \). The second row expresses the objective of minimizing the queue variation. \( D_{z+} \) weighs the value of the drop/mark probability in the output. Different weight values were tested, from 0.3 to 0.9. Results were quite similar, so, 0.5 was adopted.

\( C_y \) indicates that the value of interest measured in the output is the queue size in the previous RTT. Finally, \( D_{yw} \), weighs the noise in the measured output, which is, in general, 10% of the value in matrix \( B_u \).

After defining the performance goals, the next step is to couple system (10) with the controller (11). Let \( \hat{x}(t) \) be the augmented state vector which contains the state vector \( x(t) \) and the controller state vector \( \hat{x}(t) \), \( \hat{x}(t)^{-1} = \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} \).

The connection of system (10) with the controller (11) yields the linear delay system:

\[
\begin{align*}
\dot{\hat{x}}(t) &= A_0 \hat{x}(t) + A_1 \hat{x}(t - R_0) + B_w(t); \\
z(t) &= C_0 \hat{x}(t) + C_1 \hat{x}(t - R_0) + D_{wz} w(t);
\end{align*}
\]

(13)

where:

\[
A_0 = \begin{bmatrix} A_0 & B_u C_0 \\ 0 & A_0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} A_1 + B_u \hat{D} C_y & B_u \hat{C}_1 \\ B_u \hat{D} C_y & A_1 \end{bmatrix}, \\
B = \begin{bmatrix} B_w + B_u \hat{D} D_{yw} \\ B_u \hat{D} D_{yw} \end{bmatrix}, \quad C_0 = \begin{bmatrix} C_1 & D_{zu} \hat{C}_0 \end{bmatrix}, \\
C_1 = \begin{bmatrix} D_{zu} \hat{D} C_y & D_{zu} \hat{C}_1 \end{bmatrix}, \quad D = \begin{bmatrix} D_{zu} \hat{D} D_{yw} \end{bmatrix}.
\]

To ensure stability of system (13), Theorem 4-b in [3] is used. This theorem states that a system like (13), is asymptotically stable if \( H_{yw} \leq \gamma \), if there exist symmetric and positive definite matrices \( W, Y_0 \) and \( X_0 \), and matrices \( F, R, \)

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.
and $Q_j$, for $j = 0, 1$, such that the following LMI has a feasible solution:

$$
\begin{equation}
\begin{bmatrix}
A_0 + A_1^T + X_1 & (\bullet)^T \\
A_1^T & -X_1
\end{bmatrix} < 0
\end{equation}
$$

(14)

$$
\begin{equation}
\begin{bmatrix}
W & (\bullet)^T \\
B & P_0
\end{bmatrix} > 0, \text{trace}(W) < \gamma
\end{equation}
$$

(15)

where $A_0$, $A_1$, $B$, $C_0$, $C_1$ and $P_0$ are given by:

$$
\begin{align*}
A_0 &= \begin{bmatrix} A_0 X_0 + B_u L_0 & A_0 \\ Q_0 & Y_0 A_0 \end{bmatrix}, \\
A_1 &= \begin{bmatrix} A_1 X_0 + B_u L_1 & A_1 \\ Q_1 & Y_0 A_1 + FC_y \end{bmatrix}, \\
B &= \begin{bmatrix} B_w \\ Y_0 B_w + FD_{yw} \end{bmatrix}, \\
C_0 &= \begin{bmatrix} C_x X_0 + D_{xw} L_0 \\ C_x \end{bmatrix}, \\
P_0 &= \begin{bmatrix} X_0 & I \\ I & Y_0 \end{bmatrix}
\end{align*}
$$

This convex problem was numerically solved by using the software LMI Sol Network parameters were $R_0 = 200$ millisecond, $C_0 = 8333$ packets/seconds, that corresponds to a link capacity of 1 Gb/s and average packet size of 1500 bytes and $N_0 = 10$ HSTCP flows. These parameters values were chosen to obtain an equilibrium point that gives an exceedingly small value of the loss probability $p_0$ to sustain a large window size. A feasible solution was found, and, therefore, system (13) is stable.

After finding a feasible solution, the next step is to determine the parameters of controller (11). First, arbitrary singular matrices $\hat{U}_0$ and $\hat{V}_0$ should be chosen, such that $V_0 U_0 = I - Y_0 X_0$. The matrices used were $U_0 = X_0$ and $V_0 = X_0^{-1} - Y_0$. Then, the controller parameters are determined by:

$$
\begin{equation}
\begin{bmatrix}
\hat{A}_0 & \hat{A}_1 & \hat{B} \\
\hat{C}_0 & \hat{C}_1 & \hat{D}
\end{bmatrix} = K.M.N.
\end{equation}
$$

(16)

where:

$$
\begin{align*}
K &= \begin{bmatrix} V_0^{-1} & V_0^{-1} Y_0 B_u \end{bmatrix}, \\
M &= \begin{bmatrix} Q_0 - Y_0 A_0 X_0 & Q_1 - Y_0 A_1 X_0 \\ L_0 & L_1 & F \end{bmatrix}, \\
N &= \begin{bmatrix} U_0^{-1} \\ -C_0 U_0 X_0 U_0^{-1} \\ U_0^{-1} - C_0 U_0 X_0 U_0^{-1} \end{bmatrix}
\end{align*}
$$

In the obtained solution, matrices $\hat{A}_1$ and $\hat{C}_1$ are approximately zero, and, can be ignored. As a result, the cancellation of the delay terms in the system leads to a rational controller. The delay cancellation, when possible, is the optimal solution to an $H_2$ norm minimization problem [3]. The transfer function in the frequency domain of the controller $C_{HSTCP-H2}(s)$ is given by:

$$
C_{HSTCP-H2}(s) = \frac{0.0002836 s + 0.0009829}{s^2 + 3.687e6 s + 1278e6}
$$

(17)

For a digital implementation of $C_{HSTCP-H2}$, it is necessary to choose a sampling frequency, $f_s$, so that a representation in the $z$-domain is obtained. The frequency chosen was $f_s = 8333Hz$, which is 10% of the link capacity $C_0$. $C_{HSTCP-H2}$ in the $z$-domain is, thus, given by:

$$
C_{HSTCP-H2}(z) = \frac{a z^2 + b z - c}{z^2 - d z - e} = \frac{7.361e-10 z^2 + 3.06e-13 z - 7.358e-10}{z^2 - 0.98609 z - 0.91311}
$$

(18)

\textbf{HSTCP-H2-AQM-ProbabilityFunction()}

$$
\begin{align*}
p_0 &= \alpha_0 N_0^2 / b_0 (C_0 R_0)^2; \\
p &= q_0 (c - a - b) + a * q + b * q_{old} - c * q_{old} + p_0 (1 - d - e) + d * p_{old} + e * p_{old} \\
p_{old1} &= p_{old}; \\
q_{old1} &= q_{old}; \\
p_{old} &= p_0; \\
q_{old} &= q_0;
\end{align*}
$$

\textbf{Fig. 1.} The algorithm for the computation of the value of the marking probability

\textbf{Fig. 2.} Topology used in the simulation experiments

The transfer function between $\delta p = p - p_0$ and $\delta q = q - q_0$ in (18), can be converted in a difference equation at discrete times $kT$, where $T = \frac{1}{f_s}$:

$$
\begin{align*}
\delta p_{1} (kT) &= a \delta q (kT) + b \delta q ((k - 1)T) - c \delta q ((k - 2)T) \\
&+ d \delta p_{1} ((k - 1)T) + e \delta p_{1} ((k - 2)T)
\end{align*}
$$

(19)

The algorithm for the calculation of the mark/drop probability value of H2-AQM is straightforward, and is executed at each sampling instant $1/f_s$ (Figure 1). Initially, $p_0$ is computed, based on the parameters given ($N_0$, $C_0$, $R_0$). Then, the probability value is calculated based on (19). The algorithm needs for auxiliary variables: $q_{old}$, $p_{old}$, $q_{old1}$ and $p_{old1}$, which are used to store the values of $q$, $p$, $q_{old}$ and $p_{old}$ respectively in the last RTT.

\textbf{V. NUMERICAL RESULTS}

To assess the performance of the HSTC-H2-AQM controller, the algorithm in Figure 1 was implemented in the NS simulator and the equilibrium point presented in Section III, was used. A sampling frequency value of 8333 Hz was employed to derive the digital controller, leading to values of $7.361e^{-10}$, $3.06e^{-13}$, $7.358e^{-10}$, 0.08609 and 0.9131 for the HSTCP-H2 coefficients $a$, $b$, $c$ and $d$, $e$ respectively. RED threshold parameters were set using the approach suggest in [10].

Simulation experiments were performed using the simulator ns2.29. The topology, the link capacity and propagation delay for each link are shown in Figure 2. The link between node $R_1$ and node $R_2$ is the bottleneck link. The size of the buffer is 3333 packets which is equivalent to 20% of Bandwidth Delay product [5]. A traffic generator, called TrafficGen, was employed to generate specific traffic loads. The load was varied from 0.4 to 0.9 in order to verify the robustness of
FTP traffic was generated using an exponential distribution with mean of 512 KBytes. The traffic goes from the HSTCP sources to the destination $D_1$.

The mean size of the packets generated was 1500 bytes. The receiver window value was set to a high value (100000), so that the growth of the sender window is governed only by the network and not by the receiver.

Routers are fed by both HSTCP and UDP flows. Non-responsive flows of CBR/UDP type representing up to 20% of link capacity were included to verify the robustness of the AQM controller under the presence of noise traffic. They were generated and finalized in different intervals from the UDP sources to the destination $D_2$.

Figure 3 shows the queue length as a function of the network load. As the load increase, the queue length also increase as expected. The HSTCP-H2 produced longer queues than RED did. Under loads of 0.4 the queue produced by HSTCP-H2 is 42% longer than the one produced by RED while under loads of 1.0 the queue length produced by HSTCP-H2 is almost three times of the one produced by RED. The growth of the queue length produced by HSTCP-H2 leads to better per connection performance as will be shown in the following figures. Actually, it is a consequence of the effective resource utilization produced by HSTCP-H2. The shorter queue length values produced by RED result from of its lack of flexibility to adjust to load fluctuations, yielding to poor per connection performance.

Figure 4 and Figure 5 display, respectively, the mean congestion window size given in units of MSS and the mean goodput per connection given as percentage of the link capacity both as a function of the network load. Figure 6 shows the number of connections supported in these two experiments: when using HSTCP-H2 and when using RED.

The window size produced by HSTCP-H2 is considerably larger than the one produced by RED. The window size produced by HSTCP-H2 under loads of 0.4 is 107% larger than the one produced by RED, whereas they are three times larger under loads of 0.9.

Such window sizes yield to much higher goodput per connection when the HSTCP-H2 controller is employed than when RED is employed. The per connection goodput given by HSTCP-H2 is 40% higher under loads of 0.4 than those given by RED. Under loads of 0.9, such difference can be of the order of 100%. These results make evident the advantage of using HSTCP-H2 for individual connections, in despite of producing larger queues.

Figure 7 shows the mean number of retransmissions per connection as a function of the network load. Under light to moderate loads the number of RTO’s produced by HSTCP-H2 is lower than the one produced by RED. For loads of 0.4, the mean number of RTO’s per connection is 78% lower than those given by RED. As the network load increases, this difference decreases due to the loss of a higher number of packets. Under loads of 0.9, this difference is of the order of 0.5%. However, under loads conducted in both experiments, HSTCP and RED, to generate specific loads is the same.

Moreover, HSTCP-H2 yields to a lower number of retransmission time out up to loads of 0.9 due to better utilization of the queue space. Figure 7 shows the mean number of retransmissions per connection as a function of the network load. Under light to moderate loads the number of RTO’s produced by HSTCP-H2 is lower than the one produced by RED. For loads of 0.4, the mean number of RTO’s per connection is 78% lower than those given by RED. As the network load increases, this difference decreases due to the loss of a higher number of packets. Under loads of 0.9, this difference is of the order of 0.5%. However, under loads
between 0.9 and 1.0, the long queue produced by HSTCP-H2 leads to a higher number of losses than RED does.

Although HSTCP-H2 produces longer queues than RED does, it yields to lower mean RTT values (Figure 8). Such results is a consequence of the jitter minimization criterion adopted as a design objective for HSTCP-H2. RED produces RTT values in a much wider range than HSTCP-H2 does.

Simulation using web traffic was also conducted and results reinforce the conclusions reported here.

VI. CONCLUSION

In the past years, several TCP variants have been proposed in order to overcome the inefficiency of TCP Reno to utilize the high-speed networks. HSTCP, one of the first variant proposed, is able to maintain a high utilization of resources, regardless of the capacity of the network links.

Although powerful and necessary to prevent network collapse, the congestion control mechanisms of TCP Reno and TCP variants are not sufficient to avoid network congestion due to the limited control that TCP sources exert in the aggregate network traffic in the presence of unresponsive flows, since these do not slow down their sending rates under congestion. The efficacy of TCP congestion control mechanism relies on other mechanisms, such as Active Queue Management (AQM) to avoid congestion.

This paper introduced an AQM controller derived using Optimal Control Theory for networks employing the HSTCP protocol. Results derived simulation shows that higher goodput per connection are achieved when the HSTCP-H2 controller is used when RED is employed. Moreover, lower number of retransmission time out and RTT values are obtained when HSTCP-H2 is used. These results indicate that HSTCP-H2 is a potential candidate for adoption as AQM policy in networks that use the HSTCP protocol. Simulation using different topologies have been derived and results reinforce the advantage of adopting the HSTCP-H2.

REFERENCES