Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Towards Satisfying an Application Level Quality Measure via a Neural Network TCP-like Protocol

Christos N. Houmkozlis and George A. Rovithakis

Abstract— **A new TCP-like protocol is proposed in this paper aiming at regulating the packet round trip to satisfy a user imposed desired application sending time. For this purpose linear in the weights neural networks are employed to construct an on-line model for the estimated round trip depending on source rate (control input) and congestion level which is directly measured from the network via monitoring the state of the ECN bit for each packet acknowledgement. When the network has enough resources to support the user requirement, the neural network based rate control algorithm guarantees the uniform ultimate boundedness of the achieved round trip with respect to an arbitrarily small neighborhood of the desired round trip. In the unfortunate case of high congestion, the controller reduces the rate to the point of congestion escaping. To improve the achievable sending time, introducing the notion of communication channels, whose number is regulated via an appropriately designed algorithm, parallelizes the transmission process. We assume the nonexistence of uncontrollable traffic flows (e.g., UDP) as well as negligible propagation delays. Simulation studies illustrate the approach, highlighting various performance measures (i.e. fairness, buffer level, router utilization) and verifying theoretical analysis.**

I. INTRODUCTION

Control problems in communication networks are generally nonlinear, dynamic, and complex. However, they have become increasingly important owing to the explosive expansion and growth of traffic in the Internet. Congestion control in the Internet is a significant problem, which has been the main subject of intensive studies over the last decade [2],[6],[10],[13]. Congestion occurs in the Internet when the users of the network collectively demand more resources like bandwidth and buffer space, than the network has to offer.

Congestion control is a distributed algorithm to share network resources among competing sources. It consists of two components: a source algorithm that dynamically adjusts rate (or window size) in response to congestion in its path, and router algorithm that updates, implicitly or explicitly, a congestion measure and sends it back to sources that use the router. On the current Internet, the source algorithm is carried out by TCP, and the router algorithm is carried out by (active) queue management (AQM) schemes such as DropTail or RED [4]. Different protocols use different methods to measure congestion, e.g., TCP Reno uses loss probability as congestion measure, while TCP Vegas [11] uses queuing delay.

C. Houmkozlis and G. Rovithakis are with Dept. of Electrical and Computer Engineering, Aristotle University of Thessaloniki 54124, Thessaloniki, GREECE (houm,robi)@eng.auth.gr

Source algorithms can be roughly categorized into two classes: rate-based algorithms [3] and window-based algorithms [18]. A rate-based algorithm directly controls the transmition rate of the connection, based on feedback from the network and on measurements taken at the source. A window-based algorithm adjusts the congestion window size, which is the maximum number of outstanding packets, in order to control the trasmition rate and backlog inside the network, associated to the connection.

Nonlinear optimization techniques have also been used to address the rate adaptation problem. Their target is to determine the source rate that leads to the maximization or minimization of a utility function subject to network constrains. Indicative works include [7],[10],[12].

Neural nets have been applied in the congestion control of ATM networks [5],[19] and TCP networks[21],[22].

In this paper we present a new TCP-like protocol based on sending time, which is an application level quality measure. We consider a source that has to transmit a prespecified amount of constant size packets to a receiver, at a user imposed desired sending time. Assuming that a single packet is transmitted every round trip, a desired round trip time is calculated. Exploiting the approximation capabilities of the linear in the weights neural networks, an on-line dynamic model of the estimated packet round trip time is derived, depending on source rate (control input) and congestion level. The latter is calculated via monitoring the ECN bit of each received packet acknowledgment. The packet round trip model is nonlinear with respect to the control input (source rate). Assuming that the network has enough resources to support the user requirement on sending time, a neural network based rate control algorithm is proposed to guarantee the uniform ultimate boundedness of the round trip with respect to an arbitrarily small set of the desired round trip, thus achieving user requirements. In the unfortunate scenario of high congestion, in which the packet round trip may grow without bound, the controller is designed to reduce the transmition rate to the point of congestion escaping.

To improve the achievable sending time, the transmition process is parallelized by introducing the notion of communication channels (equivalent virtual sources), each having own rate controller. The source packets are divided into constant size groups. Based on a neural network model of the group sending time, the number of channels is automatically updated to guarantee the uniform ultimate boundedness of the group sending time with respect to an arbitrarily small neighbourhood of its corresponding desired value, provided

0-7803-9568-9/05/$20.00 ©2005 IEEE

the network, at that time slot, is capable of supporting such a goal. The desired group sending time is not fixed a priori. In fact it may vary for each group, trying to speed up the transmition process in the absence of congestion, by admitting lower values. The channel selection algorithm developed, operates at a higher level than the source rate controller and only after a group transmition is completed. The number of channels used, is kept constant during group transmition. Besides the improvement on achievable sending time, the number of channels influence directly the desired round trip time used by the source rate controllers to transmit the packets within a group. On the network side, we assume the non-existence of uncontrollable traffic flows (e.g., UDP) as well as negligible propagation delays. The theoretically developed TCP-like protocol is tested via illustrative simulation studies, highlighting various performance measures (i.e., fairness, buffer level, router utilization) and verifying theoretical analysis.

The paper is organized as follows. In Section 2, we review some basic definitions, we formulate the problem and state the necessary assumptions. The proposed rate control algorithm is analyzed in Section 3. Section 4, presents the number of channels selection algorithm, while simulation studies are performed in Section 5. Finally, we conclude in Section 6.

II. Problem formulation and preliminaries

A. Linear in the weights Neural Networks

Neural networks with a linear in the weights property will extensively be used throughout this paper. Mathematically, can be expressed by:

$$y = WS(u) \tag{1}$$

where $y \in R^n$ is the neural net output, the input is $u \in R^m$, W is a L-dimensional vector of synaptic weights and $S(u)$ is a $L \times n$ matrix of regressor terms. The regressor terms may contain high order connections of sigmoid functions [15], radial basis functions (RBFs), with fixed centers and widths [17],[14], shifted sigmoids [1], thus forming High Order Neural Networks (HONNs), RBFs and Shifted Sigmoidal Neural Networks respectively. Another class of linear in the weights neural nets is also the CMAC network which mainly uses B-splines in $S(u)$ [9].

A very important property shared by the aforementioned neural network structures is the following (see also the references above).

Property 1. *For every continuous function $f(u) : R^n \rightarrow R^m$, there exist an integer L and optimal weight values W^* such that for every $\epsilon > 0$, $\sup_{u\in\Omega} |f(u) - W^{*T}S(u)| \leq \epsilon$, $\forall u \in \Omega$ where $\Omega \subset R^m$ is a compact region.*

In other words, if the number of regressor terms L is sufficiently large, then there exist weight values W^* such that $W^{*T}S(u)$ can approximate $f(u)$ to any degree of accuracy, in a compact region. In general $\sup_{u\in\Omega} |f(u) - W^{*T}S(u)| \leq \epsilon$ becomes smaller as L increases. The reason for focusing on the linear in the weights neural networks, instead of other network structures, (e.g., multilayer neural networks), is owing to the fact that basic system properties like stability and robustness are less difficult to be derived; yet the generality is not harmed owing to Property 1.

B. Problem Formulation

Consider a network which consists of $C = \{1, 2, , m\}$ nodes and $L = \{1, 2, , l\}$ links. A source has to transmit a prespecified amount N of packets to a destination through the network at a desired sending time T_d, prescribed by the user. The transmition rate (x), which is the frequency the corresponding switch places packets on the source output buffer, is controlled by an appropriately designed rate based protocol. Assuming for a moment that a single packet is transmitted every round trip time (RT), we may calculate the desired round trip time (RT_d) as $RT_d = \frac{T_d}{N}$. To each source/destination pair there is an associated virtual circuit which has a fixed path over which all packets of a given connection travel. Upon arrival of a packet, the destination issues an acknowledgment, which is received by the source. For a given path between a source and its destination, there is a propagation delay (d_p), the time taken between sending a packet and receiving its acknowledgement by the source, when all the buffers in the routers along the path are empty. When a packet arrives to be forwarded on a router, it is marked with probability depending on its buffer level, thus altering the packet-marking bit (ECN bit) from 0 to 1. Each buffer has a finite capacity. When the buffer level exceeds its capacity, the extra packets are lost. Round trip time (RT) consists of the transmition time T_t (defined as $T_t = 1/x,\ x \in (0\ \overline{x}]$, with $\overline{x}$ to denote the maximum transmition rate, known by construction), the propagation delay d_p and some additional buffering delay db formed in every non empty router. Hence, $RT = T_t + d_p + d_b$. Typically, upon receiving an acknowledgement, the source calculates RT and estimates the buffering delay (d_b) via the recursive formula

$$\hat{d}_b(k) = a\hat{d}_b(k-1) + (a-1)m(k) \tag{2}$$

$$m(k) = \begin{cases} 1, & \text{if the k-th paket is marked} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Where in (2) $\hat{d}_b(k)$ is the buffering delay estimate after receiving the k-th packet acknowledgement and $m(k)$ is the marking bit of the k-th packet. With $\hat{d}_b(0)$ we denote the initial condition on (2). The design constant a controls the significance of old estimates in deriving the new one. From (2), (3) becomes apparent that $\hat{d}_b \in [0,\ 1]$, with $\hat{d}_b = 0$ when all buffers are empty and $\hat{d}_b = 1$ when at least one buffers in the path has reached its capacity. In the ideal case, where all buffers in the routers along the path are empty, buffering delays and $\hat{d}_b$ are zero. Hence, RT reduces to $RT = T_t + d_p = 1/x + d_p$ making it a decreasing function

of the trasmition rate x. Thus, with no loss of generality, RT can be described by

$$\dot{RT} = f(x) \tag{4}$$

where $f(x) < 0,\ \forall x \in (0,\ \overline{x}]$ is an unknown, smooth, strictly decreasing and bounded function. However, in reality, some buffers may not be empty, making $\hat{d}_b \neq 0$. As $\hat{d}_b$ admits higher values (i.e., increased congestion levels), the dependency of RT on x becomes less important and $\hat{d}_b$ actually controls its behavior. To emphasize, in the special case of $\hat{d}_b = 1$ RT has to increase without bound. Obviously, such a behavior is not included in (4). To incorporate reality, we introduce the following model for RT:

$$\dot{RT} = f(x)f_1(\hat{d}_b) + \Delta(\hat{d}_b) \tag{5}$$

where $f(x)$ is as in (4), $f_1(\hat{d}_b) \geq 0,\ \forall \hat{d}_b \in [0,\ 1]$ is an unknown, smooth, strictly decreasing function of $\hat{d}_b$ having the property $f_1(1) = 0$ and $\Delta(\hat{d}_b) > 0,\ \forall \hat{d}_b \in [0,\ 1]$ is an unknown, smooth, strictly increasing and bounded function of $\hat{d}_b$. Apparently as $\hat{d}_b \to 1$ the term $\Delta(\hat{d}_b)$ dominates $f(x)f_1(\hat{d}_b)$, forcing (5) to exhibit the appropriate behavior. Aggregating (5) yields

$$\dot{RT} = g(x, \hat{d}_b) + \Delta(\hat{d}_b),\ \ g(x, \hat{d}_b) = f(x)f_1(\hat{d}_b) \tag{6}$$

Since both $g(\cdot)$ and $\Delta(\cdot)$ are unknown functions of their arguments, neural networks of the form described in Section II.A are employed to obtain an accurate model for RT. Owing to the approximations capabilities of the linear in the weights neural networks (Property 1), we can assume, with no loss of generality, that there exists weight values W_g^*, W_Δ^* and appropriately defined regressors S_g, S_Δ such that (6) can be completely described by

$$\dot{RT} = W_g^{*T} S_g(x, \hat{d}_b) + W_\Delta^{*T} S_\Delta(\hat{d}_b) + \omega(x, \hat{d}_b) \tag{7}$$

For the modeling error term $\omega(x, \hat{d}_b)$ we pose the following assumption that is common in the literature and is a direct consequence of Property 1.

Assumption 1. *There exist an arbitrarily small positive constant δ such that $|\omega(x, \hat{d}_b)| \leq \delta,\ \forall x \in (0,\ \overline{x}]$ and $\forall \hat{d}_b \in [0,\ 1]$.*

Remark 1. Since $g(x, \hat{d}_b) \leq 0$ and $\Delta(\hat{d}_b) \geq 0$ with $g(0, 1) = 1$ and $\Delta(0) = 0$, we construct $S_g(x, \hat{d}_b)$, $S_\Delta(\hat{d}_b)$ to satisfy $S_{gi}(x, \hat{d}_b) \geq 0,\ S_{gi}(0, \hat{d}_b) = S_{gi}(x, 1) = 0,\ i = 1, 2, ..., L_g$ and $S_{\Delta i}(\hat{d}_b) \geq 0,\ S_{\Delta i}(0) = 0,\ i = 1, 2, ..., L_\Delta$. Accordingly, we may assume, without harming generality, that $W_{gi}^* < 0,\ \ i = 1, 2, ..., L_g$ and $W_{\Delta i}^* > 0,\ \ i = 1, 2, ..., L_\Delta$ Furthermore, $S_{gi}(x, \hat{d}_b) = 0 \Leftrightarrow x = 0\ and/or\ \hat{d}_b = 1,\ \ i = 1, 2, ..., L_g$ and $S_{\Delta i}(\hat{d}_b) = 0 \Leftrightarrow \hat{d}_b = 0,\ i = 1, 2, ..., L_\Delta$

For the network we make the assumptions:

Assumption 2. The propagation delays are negligible (i.e., $d_p = 0$).

Assumption 3. No uncontrollable traffic flows through the network.

We aim at solving the problem of designing a decentralized adaptive rate controller, capable of guaranteeing the user selected desired sending time T_d, in the absence of congestion. In case of congestion, the control algorithm has to reduce its source rate forcing the network to eventually escape the congestion region. Owing to the aforementioned analysis, the first objective is equivalent to selecting x in (7) to guarantee a uniform ultimate boundedness property for the error $e = RT - RT_d$ with respect to an arbitrarily small set, whenever $W_g^{*T} S_g(x, \hat{d}_b$ is the dominant term in (7), which is true, provided $\hat{d}_b$ is not close to unity (no congestion). In the proposed rate control architecture, the source sends one packet every RT and remains idle after transmition. In reality, the number of packets per source is high, which unavoidably leads to a significant increase in sending time, thus deteriorating performance. Apparently, improvement can be achieved by reducing the aforementioned idle time. One way to achieve such a behavior is by introducing the notion of communication channels. Each source creates a number (η) of communication channels that operate in parallel as if each one is a separate source (each has its own rate controller). The general architecture is presented in Fig. 1. For proper monitoring and control during the process of transmission, each source divides the total number of source packets into G groups with M packets per group and the channels undertake the parallel transmition of each packet in the group. However, now the desired round trip time (RT_d) per channel is provided by

$$RT_d = \frac{y_d \eta}{M} \tag{8}$$

where y_d denotes the desired sending time per group. After the transmition of each group, the number of channels is calculated and the transmition process is continued.

Let us assume that we have already transmitted k groups and we are about to transmit the $k + 1$ group. Let $y(k + 1)$ and $y(k)$ be the sending times of $k + 1$ and k groups respectively. Adopting similar reasoning as in the case of RT modeling, we argue that the variation $\Delta y(k + 1) = y(k+1) - y(k)$ depends on the number of channels and the buffering delay. More precisely, in the ideal case, where all buffers in the routers along the path are empty, buffering delays and $\hat{d}_b(k)$ are zero. Hence,

$$\Delta y(k) = F(\eta(k)),\ \ y(0) = 0 \tag{9}$$

where $F(\eta) < 0,\ \ \forall \eta \in [\eta_{min}, \eta_{max}]$ is an unknown, smooth, strictly decreasing and bounded function. In reality though, some buffers may not be empty, making $\hat{d}_b(k) \neq 0$. As $\hat{d}_b(k)$ admits higher values, the dependency of $\Delta y(k)$ on $\eta(k)$ becomes insignificant and $\hat{d}_b(k)$ actually controls its behavior. To emphasize, in the special case of $\hat{d}_b(k) = 1$, $\Delta y(k)$ has to increase without bound. Obviously, such a behavior is not included in (9). To incorporate reality, we introduce the following model

$$\Delta y(k) = F(\eta(k))\overline{F}(\hat{d}_b(k)) + F_1(\hat{d}_b(k)),\ \ y(0) = 0 \tag{10}$$

where $F(\eta(k))$ is as in (9), $\overline{F}(\hat{d}_b(k)) \geq 0,\ \forall \hat{d}_b(k) \in [0,1]$ is an unknown, smooth, strictly decreasing function of $\hat{d}_b(k)$ having the property $\overline{F}(1) = 0$ and $F_1(\hat{d}_b(k)) \geq 0,\ \forall \hat{d}_b \in [0,1]$ is an unknown, smooth, strictly increasing and bounded function of $\hat{d}_b(k)$. Apparently as $\hat{d}_b(k) \to 1$ the term $F_1(\hat{d}_b(k))$ dominates $F(\eta(k))\overline{F}(\hat{d}_b(k))$ forcing (10) to exhibit the appropriate behavior. Aggregating,

$$y(k+1) = y(k) + F_o(\eta(k), \hat{d}_b(k)) + F_1(\hat{d}_b(k)) \quad (11)$$

Since both F_o and F_1 are unknown functions of their arguments, neural networks of the form described in Section II.A are employed to obtain an accurate model for $y(k)$. Owing to the Property 1, we can assume, with no loss of generality, that there exists weight values W_o, W_1 and appropriately defined regressors S_o, S_1 such that (11) can be completely described by

$$\begin{aligned} y(k+1) &= y(k) + W_o^T S_o(\eta(k), \hat{d}_b(k)) + W_1^T S_1(\hat{d}_b(k)) \\ &\quad +\omega_1(\eta(k), \hat{d}_b(k)),\ y(0) = 0 \end{aligned} \quad (12)$$

Following the reasoning of Remark 2 we may construct $S_o(\eta(k), \hat{d}_b(k)),\ S_1(\hat{d}_b(k))$ to satisfy $S_{oi}(\eta(k), \hat{d}_b) \geq 0,\ S_{oi}(0, \hat{d}_b(k)) = S_{oi}(\eta(k), 1) = 0,\ i = 1, 2, ..., L_o$ and $S_{1i}(\hat{d}_b(k)) \geq 0,\ S_{1i}(0) = 0,\ i = 1, 2, ..., L_1$ and seek for optimal weights $W_o^*,\ W_1^*$ with the property $W_{oi}^* < 0,\ i = 1, 2, ..., L_o$ and $W_{1i}^* > 0,\ i = 1, 2, ..., L_1$. In this way we guarantee the approximation $W_o^{*T} S_o(\eta(k), \hat{d}_b(k))$, $W_1^{*T} S_1(\hat{d}_b(k))$ of $F_o(\eta(k), \hat{d}_b(k))$ and $F_1(\hat{d}_b(k))$ respectively, are of the correct sign.

Assumption 4. *There exist an arbitrarily small positive constant δ_1 such that $|\omega_1(\eta(k), \hat{d}_b(k))| \leq \delta_1,\ \forall \eta(k) \in [\eta_{min},\ \eta_{max}]$ and $\forall \hat{d}_b(k) \in [0,\ 1]$.*

Let us define the desired group sending time y_d as:

$$y_d(k+1) = \frac{T_d - \sum_{i=1}^{k} y(i)}{G - k},\ \ k = 0, 1, 2, ... \quad (13)$$

In this way y_d is not fixed a priori. The reasoning behind (13) is to equip the transmition process with a correction mechanism that will try to speed up transmition in the absence of congestion, by forcing y_d to admit lower values.

The second objective of this paper is to design a channel selection algorithm operating towards guaranteeing a uniform ultimate boundedness property for the error $\tilde{y}(k) = y(k) - y_d(k)$, thus, providing the means for the automatic determination of the number of channels, which remains constant for each group transmition. Simultaneously, the desired round trip time per channel is calculated via (8) which in turn is downloaded to the channel rate controller, who undertakes the responsibility of group transmition.

Remark 4. The design of the channel rate controller is based on fluid flow model of RT. However, we didn't follow the same approach when developing the channel selection algorithm, where a discrete time model was adopted, since the latter operates on groups of packets, remaining idle during group transmition.

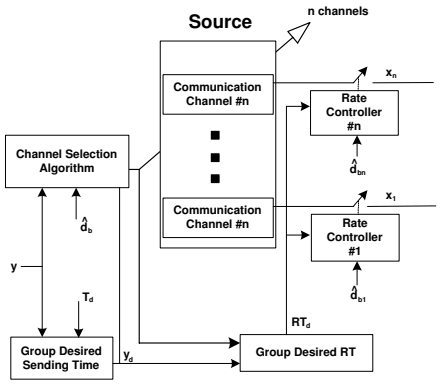

Fig. 1. The General Architecture.

III. Neural Network Rate Control Algorithm

The purpose of this section is to design a rate control algorithm based on the continuous time model of RT whose validity was explained in Section II. The analysis will be performed on a per channel basis, since all communication channels operate in parallel with no coupling among them. Recall that

$$\dot{RT} = W_g^{*T} S_g(x, \hat{d}_b) + W_\Delta^{*T} S_\Delta(\hat{d}_b) + \omega(x, \hat{d}_b) \quad (14)$$

After adding and subtracting the terms $\hat{W}_g^T S_g(x, \hat{d}_b)$ and $\hat{W}_\Delta^T S_\Delta(\hat{d}_b)$, (9) becomes

$$\begin{aligned} \dot{RT} &= -\tilde{W}_g^T S_g(x, \hat{d}_b) - \tilde{W}_\Delta^T S_\Delta(\hat{d}_b) + \hat{W}_g^T S_g(x, \hat{d}_b) \\ &\quad +\hat{W}_\Delta^T S_\Delta(\hat{d}_b) + \omega(x, \hat{d}_b) \end{aligned} \quad (15)$$

where $\tilde{W}_i^T = \hat{W}_i^T - W_i^{*T}\ i = g, \Delta$ and $\hat{W}_g^T,\ \hat{W}_\Delta^T$ are weight estimates of the unknown weight values W_g^{*T}, W_Δ^{*T} respectively.

The control objective is to force the state to follow a given reference value RT_d, calculated via (8) which by definition is a positive constant. Define the control error e as $e = RT - RT_d$. Differentiating e with respect to time we get:

$$\begin{aligned} \dot{e} &= -\tilde{W}_g^T S_g(x, \hat{d}_b) - \tilde{W}_\Delta^T S_\Delta(\hat{d}_b) + \hat{W}_g^T S_g(x, \hat{d}_b) \\ &\quad +\hat{W}_\Delta^T S_\Delta(\hat{d}_b) \end{aligned} \quad (16)$$

To derive stable control and update laws, Lyapunov stability theory is employed. Taking the Lyapunov function candidate $V = \frac{k_1}{2} e^2 + \frac{1}{2}|\tilde{W}_g|^2 + \frac{1}{2}|\tilde{W}_\Delta|^2 + \frac{1}{2}x^2$, with $k_1 > 0$ a design constant and following certain Lyapunov stability arguments we can prove the theorem:

Theorem 2. Consider a network satisfying Assumption 2,3 where all sources operate over TCP and the RT per packet model (14). The control and update laws:

$$\begin{aligned} \dot{x} &= \mathcal{P}_x\{\frac{1}{x}[-k_1 e \hat{W}_g^T S_g(x, \hat{d}_b) - k_1 e \hat{W}_\Delta^T S_\Delta(\hat{d}_b) \\ &\quad -\gamma e^2 + \phi]\} \end{aligned} \quad (17)$$

$$\phi = \begin{cases} k_1 e \hat{W}_\Delta^T S_\Delta(1), & \text{if } \begin{matrix} \hat{d}_b = 1,\ e < 0 \text{ and} \\ -k_1 e \hat{W}_\Delta^T S_\Delta(\hat{d}_b) - \gamma e^2 > 0 \end{matrix} \\ 0 \quad , & \text{otherwise} \end{cases}$$

$$\dot{\hat{W}}_g = \mathcal{P}_g - k_g \hat{W} g^T + k_1 e S_g(x, \hat{d}_b) \tag{18}$$

$$\dot{\hat{W}}_\Delta = \mathcal{P}_\Delta - k_\Delta \hat{W} \Delta^T + k_1 e S_\Delta(\hat{d}_b) \tag{19}$$

where k_g, $k_\Delta > 0$ are design constants and $\mathcal{P}_g$, $\mathcal{P}_\Delta$ denote the projection operators with respect to the convex sets $\mathcal{R}_g = \{\hat{W}_g \in R^{L_g} : \hat{W}_{gi} < 0 \quad i = 1,..,L_g\}$, $\mathcal{R}_\Delta = \{\hat{W}_\Delta \in R^{L_\Delta} : \hat{W}_{\Delta i} > 0 \quad i = 1,..,L_\Delta\}$ guarantee:

- a uniform ultimate boundedness property of e with respect to the arbitrarily small set $\mathcal{E}$, as well as the boundedness of all other signals in the closed loop, provided $\hat{d}_b < 1$.
- $\dot{x} < 0$ whenever $\hat{d}_b = 1$.

Remark 3. Theorem 2 practically states that the proposed scheme achieves the imposed RT_d provided the network is not in congestion (i.e., $\hat{d}_b < 1$). In the opposite, ($\hat{d}_b = 1$), the rate controller forces the network to escape congestion.

IV. Number of Channels Selection Algorithm

The purpose of this section is to develop a channel selection algorithm to automatically determine the required number of channels during transmission.

Define the control and weight estimation errors as $\tilde{y}(k) = y(k) - y_d(k)$, $\tilde{W}_i(k) = W_i - \hat{W}_i(k)$, $i = 0, 1$.

To derive stable control and update laws, Lyapunov stability theory is employed. Taking the Lyapunov function candidate

$$J = \tilde{y}(k)\tilde{y}(k) + \tilde{W}_o^T(k)\tilde{W}_o(k) + \tilde{W}_1^T(k)\tilde{W}_1(k) + \eta(k)$$

and selecting

$$\begin{aligned} \hat{W}_o(k+1) &= \hat{W}_o(k) + \mathcal{P}_o\{[y(k) - y_d(k+1)] \\ &\quad S_o(\eta(k), \hat{d}_b(k))\} \end{aligned} \tag{20}$$

$$\begin{aligned} \hat{W}_1(k+1) &= \hat{W}_1(k) + \mathcal{P}_1\{[y(k) - y_d(k+1)] \\ &\quad S_1(\hat{d}_b(k))\} \end{aligned} \tag{21}$$

$$\begin{aligned} \eta(k+1) &= \eta(k) + \mathcal{P}_\eta\{2y_d(k+1)y_d(k) - y_d^2(k+1) \\ &\quad + [y(k) - y_d(k+1)]^2 |S_1(\hat{d}_b(k))|^2 \\ &\quad 2[y(k) - y_d(k+1)]\hat{W}_o^T(k) S_o(\eta(k), \hat{d}_b(k)) \\ &\quad + [y(k) - y_d(k+1)]^2 |S_o(\eta(k), \hat{d}_b(k))|^2 \\ &\quad + 2[y(k) - y_d(k+1)]\hat{W}_1^T(k) S_1(\hat{d}_b(k)) \\ &\quad - y_d^2(k) - A - \alpha_1 \tilde{y}^2(k)\},\ \alpha_1 > 0 \end{aligned} \tag{22}$$

where α_1 and $\mathcal{P}_o$, $\mathcal{P}_1$ projection operators with respect to the convex sets $\mathcal{W}_o = \{\hat{W}_o \in R^{L_o} : \hat{W}_{oi} < 0 \quad i = 1,..,L_o\}$, $\mathcal{W}_1 = \{\hat{W}_1 \in R^{L_1} : \hat{W}_{1i} > 0 \quad i = 1,..,L_1\}$ used to guarantee that the estimates $\hat{W}_o S_o(\eta(k), \hat{d}_b(k))$, $\hat{W}_1 S_1(\hat{d}_b(k))$ are of the correct sign, provided W_o^*,$W_o(0) \in \mathcal{W}_o$ and W_1^*,$W_1(0) \in \mathcal{W}_1$ with $W_o(0)$, $W_1(0)$ the initial values of $\hat{W}_o$, $\hat{W}_1$ respectively, while $\mathcal{P}_\eta$ is a projection operator used to guarantee that η is actually confined within $[\eta_{min}, \eta_{max}]$ $\forall k$, we can prove the theorem:

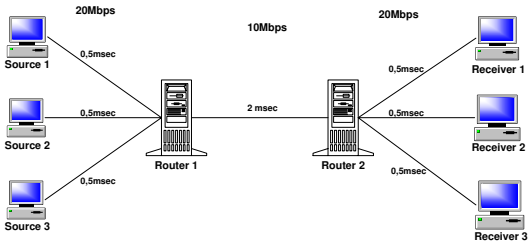

Fig. 2. Network Configuration.

Theorem 3. Consider the system (12) the number of channels selection algorithm (20), (21), (22) guarantees the uniform ultimate boundedness of the group sending time error $\tilde{y}(k) = y(k) - y_d(k)$, where $y_d(k)$ is given by (13), with respect to the arbitrarily small set $\mathcal{E}_1 = \{\tilde{y}(k) : \tilde{y}(k) \geq \frac{\delta_1}{(1+\alpha_1)} + \sqrt{\frac{\delta_1^2}{(1+\alpha_1)^2} + \frac{\overline{\delta}}{(1+\alpha_1)}}\}$ where

$$\begin{aligned} \overline{\delta} &= |W_o^T|^2 |S_o(\eta(k), \hat{d}_b(k))|^2 + |W_1^T|^2 |S_1(\hat{d}_b(k))|^2 \\ &\quad + 2|W_o^T||S_o(\eta(k), \hat{d}_b(k))||W_1^T||S_1(\hat{d}_b(k))| \\ &\quad + 2\delta_1 |W_o^T||S_o(\eta(k), \hat{d}_b(k))| + 2\delta_1^2 + 2\delta_1 y(k) \\ &\quad + 2\delta_1 |W_o^T||S_o(\hat{d}_b(k))| - 2\delta_1 y_d(k+1) \end{aligned}$$

whose size is directly controlled by α_1, as well as the boundedness of all other signals in the closed loop, provided $\hat{d}_b(k) < 1$.

Remark 4. Theorem 3 guarantees certain stability properties provided the group buffering delay estimate per source $\hat{d}_b(k) < 1$ which is satisfied due to the action of the rate control algorithm.

Remark 5. Notice that (22) outputs a real number in $[\eta_{min}, \eta_{max}]$. However, by definition $\eta \in Z$. To guarantee the aforementioned property, $\eta(k)$ is rounded off to the smallest integer preserving the stability properties of the algorithm.

V. Simulations

In this section the design methodology developed herein, is illustrated via simulations performed on a simple network configuration. The network consists of 2 Routers and 3 pairs of TCP sources and receivers as shown in Fig. 2. Obviously the link between the two routers generate a bottleneck in router 1. Each source has to transmit a prespecified amount of constant size packs (in this example 15000 packs of 1000bits), to the corresponding receiver, at a desired sending time (T_d) prescribed by the user. In addition every router has a buffer with maximum capacity of 60 packs. All routers mark receiving packs with probability p_m equal to

$$p_m = \begin{cases} \frac{\text{buffer level}}{40}, & \text{if buffer level} < 40 \\ 1 \quad , & \text{if } 40 \leq \text{buffer level} \leq 60 \end{cases}$$

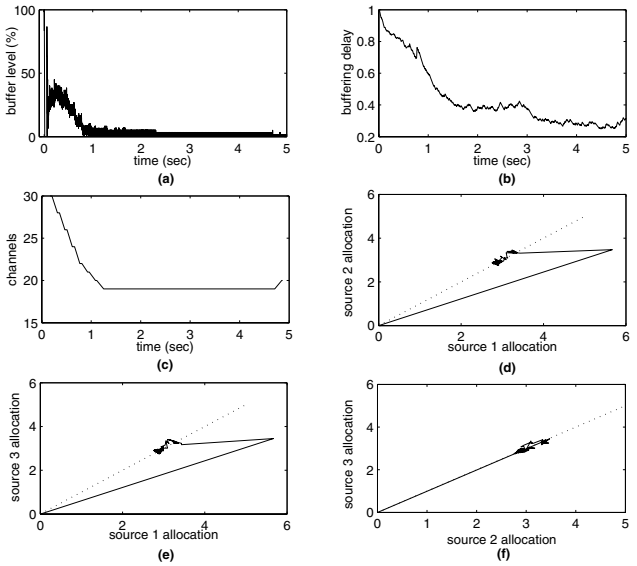

Fig. 3. Performance of the proposed TCP-like protocol ($T_d = 5.0sec$).

To achieve our goal we employ the TCP-like protocol developed in Sections III and IV with $S_g(x, \hat{d}_b) = 0.7 S_{g1}(\hat{d}_b) S_{g2}(x)$, $S_{g1}(\hat{d}_b) = (\frac{1}{1+e^{-11(\hat{d}_b - 0.45)}})$, $S_{g2}(x) = [(\frac{1}{1+e^{-(x-0.1)}})(\frac{1}{1+e^{-(x-0.3)}}), \frac{1}{1+e^{-(x-0.2)}})^2, (\frac{1}{1+e^{-(x-0.3)}})^2, (\frac{1}{1+e^{-(x-0.4)}})^2, (\frac{1}{1+e^{-(x-0.5)}}), (\frac{1}{1+e^{-(x-0.6)}})^2, (\frac{1}{1+e^{-(x-0.7)}}), (\frac{1}{1+e^{-(x-0.8)}})^2, (\frac{1}{1+e^{-(x-0.9)}})^2]^T$, $S_\Delta(x, \hat{d}_b) = (\frac{2}{1+e^{-\hat{d}_b}}) - 1$, $S_o(\eta(k), \hat{d}_b(k)) = S_{o1}(\eta(k)) S_{o2}(\hat{d}_b(k))$, $S_{o1}(\eta(k)) = (\frac{1}{1+e^{-4(\eta(k)-0.25)}})^2$, $S_{o2}(\hat{d}_b(k)) = 1 - (\frac{1}{1+e^{-11(\hat{d}_b(k)-0.65)}})$, $S_1(\hat{d}_b(k)) = (\frac{1}{1+e^{-11(\hat{d}_b(k)-0.75)}})$. The parameters γ, k_1, α_1 that appear in (18), (19), (17), (22) were selected to be 2, 150, 0.6 respectively. The initial condition of (17) was set equal to 2 Mbps, while the neural network weights were initialized as $W_g(0) = [-0.7 - 0.7 - 0.7 - 0.7 - 0.7 - 0.7 - 0.7 - 0.7 - 0.7]$, $W_\Delta(0) = 0.4$, $W_o(0) = -1.5$, and $W_1(0) = 0.8$.

To evaluate the performance of the proposed TCP-like protocol, we considered the case where all sources request identical sending times T_d achieved with relative accuracy less than 1%. All results were derived with respect to the bottleneck router. Furthermore, to better illustrate the congestion escaping property of the proposed scheme we have assumed that initially we were in congestion ($\hat{d}_b = 1$). The values of the desired sending times were selected to be 5.0 sec for each source. The actual sending times achieved by the proposed rate control algorithm, was 5.012, 5.014 and 5.022 (relative accuracy less than 1%) for sources 1,2,3 respectively. Fig. 3 summarizes the achieved results. Clearly, the proposed TCP-like protocol operates toward preventing buffer overflow, escaping congestion ($\hat{d}_b < 1$) and fair bandwidth allocation, while achieving the requested desired sending times with relative accuracy less than 1%.

VI. Conclusions

We have proposed a new approach of controlling congestion in the Internet via regulating the round trip to satisfy a user imposed desired sending time. For this purpose, linear in weights neural networks have been employed to construct an on-line model for the estimated round trip, depending on source rate (control input) and congestion level which is currently directly measured from the network via monitoring the state of the marking bit of each packet acknowledgement. When the network has enough resources to support the user requirement the neural network based rate control algorithm guarantees the uniform ultimate boundedness of the achieved round trip with respect to an arbitrarily small neighborhood of the desired round trip. In the unfortunate case of high congestion, the controller reduces the rate to the point of logical congestion levels. Simulation studies illustrate and highlight the approach.

References

[1] N. E. Cotter, "The Stone-Weierstrass Theorem and its Application to Neural Networks," *IEEE Trans. on Neural Networks*, vol. 1, 1990.

[2] S. Deb and R. Srikant, "Global Stability of congestion controllers for the Internet," *IEEE Trans. on Autom. Contr.*, vol. 48, no. 6, 2003.

[3] L.A. Grieco and S. Mascolo, "Adaptive Rate Control for streaming flows over the Internet," *ACM Mult. Syst. Jour.*, vol. 9, no. 6, 2004.

[4] C. Hollot, V. Misra V, D. Towsley, W. Gong, "A Control Theoretic Analysis of RED," *Proceedings of IEEE Infocom 2001*

[5] S. Jagannathan and Talluri, "Predictive congestion control of ATM networks" *Automatica*, vol 38, no. 5, pp. 815-820, May 2002.

[6] R. Johari, D. Tan, "End-to-end congestion control for the internet: delays and stability," *IEEE/ACM Trans. Net.*, vol. 9, 2001.

[7] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237-252, Mar. 1998.

[8] H. K. Khalil, Nonlinear Systems, Singapore: Maxwell Macmillan International, 1992.

[9] S. H. Lane, D. A. Handelman, J. J. Gelfand, "Theory and Development of High-Order CMAC Neural Networks," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 23-30, 1992.

[10] S. H. Low, "A Duality Model of TCP and Queue Management Algorithms," *IEEE/ACM Trans. on Net.*, vol. 11, no. 4, August 2003.

[11] S. H. Low, L. Peterson, and L. Wang, "Understanding Vegas: A duality model," *J. ACM*, vol. 49, no. 2, pp. 207-235, Mar. 2002.

[12] S.H. Low, D.E. Lapsley, "Optimization flow control, I: basic algorithm and convergence," *IEEE/ACM Trans. Net.*, vol. 7, 1999.

[13] F. Paganini, "On the stability of optimization-based flow control," *in Proc. American Control Conference*, June 2001.

[14] T. Poggio, F. Girosi, Regularization Algorithms for Learning that are Equivalent to Multilayer Networks, *Science*, vol. 247, 1990.

[15] G. A. Rovithakis and M. A. Christodoulou, Adaptive Control with Recurrent High Order Neural Networks, Springer, London, 2000.

[16] D. A. White, D. A. Sofge, (Eds.) Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches, New York, IEEE, 1994.

[17] Y. R. Yang and S. S. Lam, General AIMD congestion control, *in Proc. Int. Conf. Network Protocols (ICNP)*, pp. 187198, Nov. 2000.

[18] R. Zhu, F. Yin, T. Qiu, "Neural Congestion Control Algorithm in ATM Networks with multiple Node," *Proc. Inter. Symp. on Neural Nets*, pp. 299-304, 2004.

[19] G. Robithakis, C. Houmkozlis, "A Neural Network Congestion Control Algorithm for the Internet," *Intern. Symp. on Intel. Control Limassol Cyprous June 27-29, 2005*

[20] W. Lin, A. Wong, T. Dillon, "Application of Soft Computing Techniques to Adaptive User Buffer Overflow Control on the Internet," *IEEE Trans. on Syst., Man, and Cyb.*, pp. 1 - 14, 2005

[21] A. Bivens, B. Szymanski, M. Embrechts, "Network Congestion Arbitration and Source Problem Prediction Using Neural Networks," *Smart Engineering System Design*, vol. 4, pp. 243-252, 2002