Flow Control in Wireless Networks

Minghua Chen, *Student Member, IEEE,* Avideh Zakhor *Fellow, IEEE*

***Abstract*— Flow control, including congestion control for data transmission, and rate control for multimedia streaming, is an important issue in information transmission in both wired and wireless networks. Widely accepted flow control methods in wired networks are TCP [1] for data, and TCP Friendly Rate Control (TFRC) [2] for multimedia. Kelly [3] [4] has laid down theoretical framework for TCP in wired networks demonstrating its optimality, fairness, and stability. However, TCP and TFRC both assume that packet loss in wired networks is primarily due to congestion, and as such, are not applicable to wireless networks in which the bulk of packet loss is due to errors at the physical layer. In this paper, we first show flow control in the wireless networks can be formulated as the same concave optimization problem Kelly defined in the wired networks. In doing so, we show that TCP and TFRC in the wireless networks pursue the optimal solution using inaccurate feedback. All existing approaches to TCP or TFRC over wireless problem correct this inaccurate feedback by casting modifications to existing protocols such as TCP, or infrastructure elements such as routers, thereby making them hard to deploy in practice. In this paper, we formulate the problem as another concave optimization problem with a different utility function, and propose a new class of solutions. Our approach is end-to-end, and achieves reasonable performance by adjusting the number of connections within an application according to a properly selected control law. The control law is based on only one bit of information, which can be reliably measured at the application layer. We show that the control system has a unique, stable equilibrium that solves the concave optimization problem, implying scalability and optimality of the solution. We apply our results to design a practical rate control scheme for data transmission over wireless networks, and characterize its performance using NS-2 simulations and actual experiments over Verizon Wireless 1xRTT data network. Analysis and simulation results also indicate our scheme is applicable to both wired and wireless scenarios.**

I. Introduction

TCP has been widely successful on the wired Internet since its first implementation by Jacobson [1] in 1988. TCP Reno, the most popular TCP version today, in its congestion avoidance stage, increases its windows size by one if no packet is lost in the previous round trip time, and halves the windows size otherwise. The *key* assumption TCP relies on is that packet loss is a sign of congestion. In wireless networks however, packet loss can also be caused by physical channel errors. As a result, the congestion assumption breaks down, and TCP seriously underutilizes the wireless bandwidth. Similar observations hold for TCP-friendly schemes, such as TCP-Friendly Rate Control (TFRC) commonly used for streaming applications [2], as they share the same key assumption as TCP. In particular, our experiments over Verizon Wireless 1xRTT wireless data network have shown that TFRC achieves only 56% of the available wireless bandwidth [5]. The need to solve the problem is becoming urgent as wireless data and streaming services are becoming increasingly more popular.

Consequently, There have been a number of efforts to improve the performance of TCP or TFRC over wireless [6]–[11]. These approaches either hide packet loss caused by wireless channel error from hosts, e.g. Snoop [6], or provide end-hosts the ability to distinguish between packet loss caused by congestion, and that caused by wireless channel error, e.g. end-to-end statistics based solutions [8]–[11].

Another alternative is to use non-loss based rate control schemes. For instance, TCP Vegas [12], in its congestion avoidance stage, uses queueing delay as a measure of congestion, and hence could be designed not to be sensitive to any kind of packet loss, including that due to wireless channel error. It is also possible to enable the routers with ECN markings capability to control rate using ECN as the measure of congestion [13]. As packet loss no longer corresponds to congestion, ECN based rate control does not adjust sending rate upon observing a packet loss.

Our recent work, MULTFRC [5], opens an appropriate number of connections to fully utilize the wireless bandwidth. This approach improves TFRC performance in wireless networks by modifying only the application layer, and as such, is also applicable to TCP. We have also proposed an improved scheme named AIO-TFRC [14], which eliminates the overhead and performance degradation associated with operating multiple connections.

While the existing approaches provide practical insight on how to improve the performance of TCP and TFRC over wireless, it is unclear whether their performance can easily scale to a network as large as the Internet, and whether they are the only possible way to achieve optimal performance. Hence a general framework for flow control over wireless is needed to address the issues of optimality and stability, and to provide guidelines and performance prediction prior to any implementation. Specifically, the following questions need to be answered:

- How does one define the problem of flow control over wireless networks? What is the analytical framework?
- Under the framework, is there any *optimal* and *stable* application layer scheme that solves the problem?

In the past decade, there has been a great deal of research activity on decentralized end-to-end flow control algorithms on wired networks. A widely recognized setting, introduced by Kelly et. al. [3] and refined for TCP in [4], views flow control schemes as algorithms to compute the optimal solution to a utility maximization problem. Similar work has focused on relating different versions of TCP to different utility functions,

This work was supported by AFOSR contract F49620-00-1-0327.

Both authors are with Department of Electrical Engineering and Computer Science at University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: {minghua, avz}@eecs.berkeley.edu).

evaluating network stability with and without delay or noise, and designing new TCP and queue management algorithms [15]–[19]. In primal algorithms proposed by Kelly et. al. [3], users adapt source rates dynamically based on the prices along the path, and the routers select a static law to determine their prices directly from the arrival rates at the link. Kelly has showed that TCP is in fact a primal like algorithm with packet loss rate as associated price function [4]. The stability for the system with and without delay, with and without disturbance, are developed in [4].

In this paper, we assume a wireless link is associated with a fixed bandwidth and a fixed packet loss rate caused by the physical channel errors. We first show that in the presence of the packet loss caused by physical channel error, flow control in the wireless network can be formulated as the same concave optimization problem defined by Kelly in the wired network [4]. In doing so, we have shown that TCP and TFRC in the wireless networks pursue the optimal solution using inaccurate feedback[1]. All existing approaches to this problem correct this inaccurate feedback by casting modifications to existing protocols such as TCP, or infrastructure elements such as routers, thereby making them hard to deploy in practice.

In this paper, we formulate the problem as another concave optimization problem with a different utility function, followed by a new class of solutions. Our approach is end-to-end, and achieves reasonable performance by adjusting the number of application's connections according to a properly selected control law. The control law is based on only one bit of information, which can be reliably measured at the application layer. We show that the resulting control system has a unique stable equilibrium that solves the concave optimization problem, implying scalability and optimality of the solution. We then apply our results to design a practical rate control scheme for data transmission over wireless networks, and characterize its performance using NS-2 simulations, and actual experiments over Verizon Wireless 1xRTT data network. Analysis and simulation results indicate our scheme is applicable to both wired and wireless scenarios.

This paper is organized as follows. Section II includes problem formulation. A new approach addressing the problem is proposed in Section III, together with analysis for its optimality and stability. Section IV shows the design of a practical scheme, and analyzes its utilization over a single user, single wireless link scenario. NS-2 simulations and actual experiments over 1xRTT wireless data networks are included in Section V. Section VI concludes the paper with discussions and future work.

II. Problem formulation

A. Overview of the flow control framework and TCP modeling over wireless

Consider a network with a set J of *resources*, i.e. links, and let C_j be the finite capacity of resource j, for $j \in J$. Let R be the set of routes, where a *route* r is a non-empty subset of J associated with a positive round trip delay T_r. Set $a_{jr} = 1$ if $j \in r$, and set $a_{jr} = 0$ otherwise. This defines a 0-1 routing matrix $A = (a_{jr}, j \in J, r \in R)$, indicating the connectivity of the network.

Associate a route r with a user, i.e. a pair of sender and receiver, and assume users behave independently; furthermore, endow a user with a sending rate $x_r \geq 0$ and a utility function $U_r(x_r)$, which is assumed to be increasing, strictly concave, and continuously differentiable. For convenience, we also define x to be a vector of users' sending rates, i.e. $x = [x_1, x_2, \ldots]^T$.

Assume utilities are additive, so that the aggregate utility of the entire system is $\sum_{r\in R} U_r(x_r)$. The flow control problem under a deterministic fluid model, first introduced by Kelly et. al. [3] and later refined in [4], is a concave optimization problem maximizing the net utility[2]:

$$\max_{x\geq 0} \sum_{r\in R} U_r(x_r) - \sum_{j\in J} \int_0^{\sum_{s:j\in s} x_s} p_j(z)\, dz, \qquad (1)$$

where $\int_0^{\sum_{s:j\in s} x_s} p_j(z)\, dz$ can be considered as the cost incurred at link j; $p_j(z)$ is called the price function and is required to be non-negative, continuous, increasing, and not identically zero. With these assumptions on $p_j(z)$, the objective function in (1), the sum of users' utility minus the costs associated with utilizing the links, is strictly concave. One common price function used in practice is the packet loss rate, which is zero if there is no congestion, and concavely increases otherwise:

$$p_j(z) = \frac{(z - C_j)^+}{z} = \begin{cases} 0, & z \leq C_j; \\ 1 - \frac{C_j}{z}, & z > C_j. \end{cases}, \qquad (2)$$

where C_j is the capacity of link j. For ease of use in the remainder of the paper, define the aggregate rate arriving at link j as follows:

$$y_j(t) = \sum_{s:j\in s} x_s(t), \quad j \in J.$$

In the rest of this paper, we assume $p_j(y_j(t))$ is small enough such that the end-to-end packet loss rate for user r, i.e. $1 - \prod_{j\in r}(1 - p_j(y_j(t)))$, is approximately $\sum_{j\in r} p_j(y_j(t))$.

Under these settings, Kelly [4] has shown TCP Reno [3] to be a primal-like algorithm, with $x_r(t)$ satisfying:

$$\dot{x}_r(t) = \frac{1}{2S}\left(\frac{2S^2}{T_r^2} - x_r^2(t)\sum_{j\in r} p_j(y_j(t))\right), \quad r \in R, \quad (3)$$

and solving the optimization problem in (1) with $U_r(x_r) = -\frac{2S^2}{T_r^2 x_r}$ where S is the TCP packet size, T_r is the end-to-end

[1] In [20], [21], we have also shown the similar formulation using the framework in [3].

[2] It is easy to show that this is nothing but a penalty relaxation solving the sum utility maximization with capacity constrains, namely:

$$\begin{aligned} \max_{x\geq 0} \quad & \sum_{r\in R} U_r(x_r) \\ \text{s.t.} \quad & Ax \leq C, \text{ (i.e. } \textstyle\sum_{s:j\in s} x_s \leq C_j, j \in J) \end{aligned}$$

[3] In the rest of the paper, we use this version of TCP.

round trip time, and $p_j(y)$ takes the form in (2). Rewriting (1) with these quantities, the net utility function becomes:

$$\max_x \left\{ -\sum_{r\in R} \frac{2S^2}{T_r^2 x_r} - \sum_{j\in J} \int_0^{\sum_{s:j\in s} x_s} \frac{(z-C_j)^+}{z}\, dz \right\}, \quad (4)$$

Thus, TCP can be viewed as a discrete version of the steepest gradient descent algorithm which solves the optimization problem in (4).

Equation (3) describes the time evolution of sending rate $x_r(t)$, whereby the user exploits only the aggregate packet loss information along its path. We assume the same flow $x_r(t)$ is presented to all links $j \in r$, even though the flow in downstream links slightly shrinks due to losses at upstream links. This is a direct implication of our previous assumption that the packet loss rate on link j, i.e. $p_j(\sum_{s:j\in s} x_s)$, is small.

Kelly [4] showed the system in (3) has a unique equilibrium, to which all trajectories converge, as follows:

$$x_r^o = \frac{\sqrt{2}S}{T_r\sqrt{\sum_{j\in r} p_j\left(y_j^o\right)}}, \quad r \in R; \quad (5)$$

where $y_j^o = \sum_{s:j\in s} x_s^o$. Equation (5) is similar to the well known TCP steady state throughput equation as described by Mahdavi and Floyd in [22]. This equilibrium is also the finite solution for the optimization problem in (1), associated with the following desirable properties: firstly, all routes are fully utilized, yet there is no congestion collapse, i.e. $\forall r \in R, \exists j \in r$, s.t. $C_j \le y_j^o < \infty$; secondly, there is roughly α-fairness [23] among users with $\alpha = 2$, implying the users are allocated rates that roughly maximize a sum of utility of the form $x_r^{-\alpha} = x_r^{-2}$.

B. TCP and TFRC over wireless

For wireless networks, we assume the links $j \in J$ are associated with not only a fixed capacity but also a packet loss rate caused by the physical channel errors, necessarily nonnegative, denoted by $\epsilon_j \ge 0, j \in J$. Nevertheless, TCP over wireless is still associated with the same concave optimization problem as with the wired networks, shown in (4). This is because neither the utility function of users in (4), i.e. $U_r(x_r)$, nor the cost associated with using network resources, i.e. $\int_0^{y_j} p_j(z)\, dz$, is a function of $\epsilon_j \ge 0, j \in J$. In effect, ϵ_j results in the price functions fed back to users to be inaccurate, since it now includes losses both due to congestion and physical channel errors. Hence TCP algorithms still aim to address the same optimization problem shown in (4), but with inaccurate prices fed back from network.

This inaccurate price function, denoted by $q_j(y_j(t))$, is the sum of ϵ_j and $p_j(y_j(t))$, under the assumption that ϵ_j is small:

$$q_j(y_j(t)) = p_j\left(y_j(t)\right) + \epsilon_j \ge \epsilon_j, \quad j \in J. \quad (6)$$

When the link is not congested, $q_j(y_j(t)) = \epsilon_j$ since all packet losses are caused by channel error; $q_j(y_j(t))$ gradually increases otherwise. With this inaccurate price, TCP now adjusts the sending rates as:

$$\dot{x}_r(t) = \frac{1}{2}\left(\frac{2S^2}{T_r^2} - x_r^2(t)\sum_{j\in r} q_j(y_j(t))\right), \quad r \in R. \quad (7)$$

Following a similar analysis in [4], one can show the system (7)-(6) has a new unique equilibrium, to which all trajectories converge, as follows:

$$\bar{x}_r = \frac{\sqrt{2}S}{T_r\sqrt{\sum_{j\in r} q_j\left(\bar{y}_j\right)}} \le \frac{\sqrt{2}S}{T_r\sqrt{\sum_{j\in r}\epsilon_j}}, \quad r \in R, \quad (8)$$

where $\bar{y}_j = \sum_{s:j\in s} \bar{x}_s$. Although it can be shown that there is roughly α-fairness among users with $\alpha = 2$, $\bar{x} \triangleq [\bar{x}_r, r \in R]$ is a suboptimal solution as it is different from the unique optimal one x^o. Furthermore, user r could suffer underutilization if $\sum_{j\in r}\epsilon_j$ is sufficiently large. For instance, in the one user one bottleneck network, underutilization happens if and only if $\frac{\sqrt{2}S}{T_r\sqrt{\epsilon}} < C$, where C is the bottleneck bandwidth and ϵ represents the aggregate packet loss rate caused by wireless channel error experienced by the user [5]. Hence the main problem with TCP over wireless is underutilization of the wireless channel; in fact, similar analysis shows that it is also the main problem with any flow control method that uses packet loss rate as price function, such as TFRC.

One straightforward solution is to provide user r with the accurate price $\sum_{j\in r} p_j(y_j(t))$, and apply it to the control law of $x_r(t)$; this could be done by either end-to-end estimation with or without cross layer information, or by hiding the wireless loss from users via local retransmissions. In fact, most existing approaches belong to this class of solutions, thus requiring modifications either to the transport protocols or to the network infrastructure, making them hard to deploy in practice.

In essence, the main challenge of TCP optimization problem shown in (4) for the wireless network setting is to accurately estimate price $\sum_{j\in r} p_j(y_j(t))$ from the noisy measurements $\sum_{j\in r} q_j(y_j(t))$. One way to overcome this problem is to specifically choose a slightly different utility maximization problem, resulting in a new solution that requires measurements that are easy to obtain in a practical networking setup. In the next section, we will show that by selecting a different utility to maximize, there exists a new, stable and optimal solution, which requires only one bit of end-to-end measurement.

III. Proposed solution

A. A new class of solutions

Motivated by our previous approach MULTFRC [5], we now propose a new approach to flow control based on gradually adjusting the number of connections for user r, denoted by $n_r(t)$. This is an end-to-end application layer based scheme, and requires no modification to the network infrastructure or the transport protocol stack. The goals of our approach are three fold. First, to stably pursue full utilizations of the bottleneck links; second, to optimally solve a net utilities maximization problem; third, to achieve fairness among users.

In MULTFRC [5], the sending rate of individual connections is controlled by TFRC itself; the number of connections $n_r(t)$ is inversely increased if there is no congestion, and is additively decreased otherwise, commonly referred as an IIAD control. The NS-2 simulations and actual experiments in [5] show that the performance of MULTFRC is reasonably

good even though there is no theoretical framework justifying its stability and scalability. Motivated by MULTFRC [5], our proposed approach here is to dynamically adjust both $x_r(t)$ and $n_r(t)$ as follows:

$$\dot{x}_r(t) = \frac{1}{2Sn_r(t)} \left(\frac{2S^2 n_r^2(t)}{T_r^2} - x_r^2(t) \sum_{j \in r} q_j(y_j(t)) \right), r \in R \tag{9}$$

$$\dot{n}_r(t) = c_r \left(\frac{1}{n_r(t)} - n_r(t) I_r[\sum_{j \in r} p_j(y_j(t))] \right), \quad r \in R \tag{10}$$

where $c_r, r \in R$ are nonnegative constants, and $I_r(\sum_{j \in r} p_j(y_j(t)))$ is an indicator function implying the congestion status of route r:

$$I_r[\sum_{j \in r} p_j(y_j(t))] = I\left(\sum_{j \in r} \frac{(y_j(t) - C_j)^+}{y_j(t)} \right) \tag{11}$$

$$= \begin{cases} 1, & \text{if route } r \text{ is congested at time } t, \\ & \text{i.e. } \sum_{j \in r} \frac{(y_j(t) - C_j)^+}{y_j(t)} > 0; \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to see that the control law of $n_r(t)$ in (10) is Inverse Increase and Multiplicative Decrease (IIMD), which is different from MULTFRC [5]. As we will see later, this modification leads to several desirable properties for the system.

As shown in (9), the control law of the aggregate rates for user r, i.e. $x_r(t)$, can be understood as the sum of rates from $n_r(t)$ individual connections, with each connection being controlled using the standard TCP Reno algorithm. Therefore, our approach does not require any modifications to the TCP protocol. On the other hand, as seen in (10), the system tries to achieve full utilization by adjusting the number of connections $n_r(t)$ accordingly. In particular,

- If a route r is underutilized, then $I_r[\sum_{j \in r} p_j(y_j(t))] = 0$; this implies that the number of connections $n_r(t)$ will increase in order to boost the user's rate $x_r(t)$, pursuing full utilization on any route r;
- If the route r is fully utilized, i.e. one of its links is congested, then $I_r[\sum_{j \in r} p_j(y_j(t))] = 1$, lowering $n_r(t)$, and hence $x_r(t)$, to prevent the system from further congestion.

The intuition behind our approach is as follows: when loss rate caused by channel error increases, individual connection's sending rate is lowered, thus users need to open more connections to increase the aggregate throughput. The $I_r(\cdot)$ is the one bit of information required from the end-to-end measurements. In practice, we can estimate $I_r(\cdot)$ using end-to-end round trip time measurements. In particular, we estimate the queuing delay by comparing current round trip time with the propagation delay, and set $I_r(\cdot) = 1$ if the queuing delay is positive, and $I_r(\cdot) = 0$ otherwise.

The system in Eqns. (9) and (10) is a coupled, discontinuous, nonlinear system. In order to verify that this system actually meets our design goals, we need to analyze the existence of a unique equilibria, its stability and its optimality in the sense of solving a utility maximization problem.

B. Discontinuity approximation and the two time scale decomposition

The discontinuities of functions $I_r[\sum_{j \in r} p_j(y_j(t))]$ and $p_j(y_j(t))$ hinder the analysis of the equilibria. To carry out the analysis, we first approximate these discontinuous functions using continuous functions, in order to generate an approximate continuous system to the original discontinuous system[4]: $\forall j \in J, r \in R$,

$$p_j(y_j(t)) \approx \frac{1}{\beta} \ln \left(1 + e^{\beta \frac{y_j(t) - C_j}{y_j(t)}} \right) \triangleq g_j(y_j(t)), \tag{12}$$

$$I_r[\sum_{j \in r} p_j(y_j(t))] \approx \frac{e^{\beta \sum_{j \in r} g_j(y_j(t))} - 1}{e^{\beta \sum_{j \in r} g_j(y_j(t))} + 1} \triangleq f(\sum_{j \in r} g_j(y_j(t))), \tag{13}$$

where β is a nonnegative constant. It should be clear that $f(\sum_{j \in r} g_j(y_j(t)) \to I_r(\sum_{j \in r} g_j(y_j(t))$ and $g_j(y_j(t)) \to p_j(y_j(t))$ as $\beta \to \infty$.

Thus, an approximate continuous version of the original system in (9) and (10) is given by: $\forall r \in R$,

$$\begin{cases} \dot{x}_r(t) = \frac{1}{2Sn_r(t)} \left(\frac{2S^2 n_r^2(t)}{T_r^2} - x_r^2(t) \sum_{j \in r} [\epsilon_j + g_j(y_j(t))] \right), \\ \dot{n}_r(t) = c_r \left(\frac{1}{n_r(t)} - n_r(t) f(\sum_{j \in r} g_j(y_j(t))) \right). \end{cases} \tag{14}$$

Since the approximate system in (14) is continuous, we can analyze its equilibrium and stability for arbitrary values of β. As $\beta \to \infty$, the system in (14) approaches the original system in (9) and (10). Therefore, the equilibria and the stability results for the approximate system also correspond to the results for the original system in the Filippov sense [24], [25].

The approximate system in (14), though continuous, is difficult to analyze in general. It is a nonlinear, coupled, multivariable system, and the two equations are not exactly symmetric even though they might appear to be so. Hence, we introduce a two time scale assumption to analyze the approximate system: *The number of connections, $n_r(t)$, changes in a timescale much slower than the source rate, $x_r(t)$.* This assumption is the key to carrying out the equilibria and stability analysis. It is also reasonable in practice, where the sending rates are expected to change on the order of tens of milliseconds, while number of connections is expected to change at a much slower rate, e.g. tens of seconds.

Under the above assumption, system (14) fits into the classical singular-perturbation framework [24] [25], and therefore can be decoupled into a short-scale and a long-scale system. The short-scale system is described by (9) with the corresponding $n_r(t), r \in R$ being constant, namely boundary system: $\forall r \in R$,

$$\begin{cases} \dot{x}_r(t) = \frac{1}{2Sn_r(t)} \left(\frac{2S^2 n_r^2(t)}{T_r^2} - x_r^2(t) \sum_{j \in r} [\epsilon_j + g_j(y_j(t))] \right), \\ n_r(t) = \text{constant}. \end{cases} \tag{15}$$

The long-scale system is described by (10) along the manifold defined by the stationary solution of (9), namely reduced

[4]We will discuss the relationship between the approximate system and the original system, and performance of the actual implementation later.

system: $\forall r \in R$,

$$\begin{cases} x_r(t) = \frac{n_r(t)\sqrt{2}S}{T_r\sqrt{\sum_{j\in r}[\epsilon_j + g_j(y_j(t))]}}, \\ \dot{n}_r(t) = c_r\left(\frac{1}{n_r(t)} - n_r(t) f(\sum_{j\in r} g_j(y_j(t)))\right). \end{cases} \tag{16}$$

Under the two times scale setting, the behavior of the system can be described as follows. On the fast timescale, $n(t)$ can be thought of as being constant since its dynamics happens at a slow timescale. The entire system can then be expressed as a boundary system shown in (15). As the boundary system is similar to Kelly et. al.'s control system on wired networks [4] except for the constant $n(t)$, and the price function $p_j(y_j(t))$ being replaced by $\epsilon_j + g_j(y(t)$, the behavior of the boundary system can be easily inferred from the known results for the system in (3). Specifically, at the fast timescale, $x(t) \triangleq [x_r(t), r \in R]$ globally exponentially converges to the equilibrium manifold defined as follows [4], [17]:

$$x_r(t) = \frac{n_r(t)\sqrt{2}S}{T_r\sqrt{\sum_{j\in r}\left[g_j\left(y_j(t)\right) + \epsilon_j\right]}}, \quad r \in R. \tag{17}$$

This can be easily shown be to an one-to-one mapping between $x(t)$ and $n(t) \triangleq [n_r(t), r \in R]$. On the slow timescale, $x(t)$ has already converged to the above equilibrium manifold, and now the system collapses into the reduced system described in (16), whose behavior determines how the approximate system evolves at the slow timescale. Therefore, together with boundary system, it fully characterizes behavior of the system for all possible timescales. For illustration purposes, we have handdrawn the time dynamics of singular perturbation system comprising of a single user with sending rate $x_1(t)$, and $n_1(t)$ connections in Figure 1. As shown, given any initial condition, the system first converges rapidly onto the equilibrium manifold, representing the fast timescale indicated by its boundary system. On the manifold, the system's behavior follows its reduced system; if the reduced system has the equilibrium point as the globally asymptotically stable equilibrium, then the trajectories, along the manifold, will converge to the equilibrium as time goes to infinity.

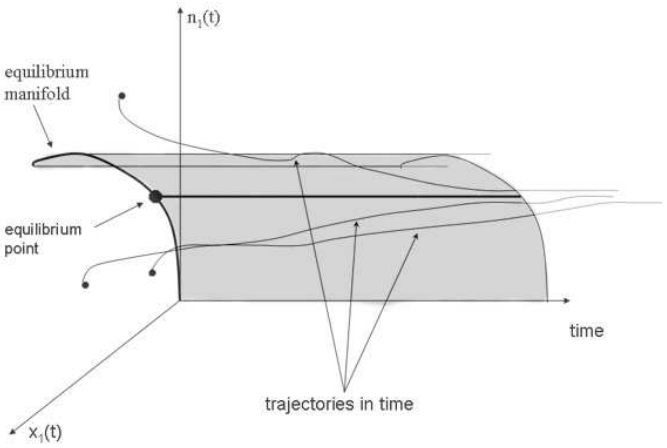

Fig. 1. Time dynamics of a singular perturbation system.

In the next subsection, we present the results on the existence of a unique optimal equilibrium of the system and its stability.

C. The existence of a unique optimal equilibrium and its stability

Given the system in (14), the first question to answer is whether it has any equilibria, and if so how many? The second question is the local and global stability of these equilibria. These two questions are important in the sense that they describe the behavior of the system as time evolves, predicting the system's performance in actual implementations in practice. For instance, if the system in (14) has no equilibrium, the users' sending rates would not converge, making the system undesirable to implement. The consequence would be even worse if the system has one globally stable equilibrium at infinity, implying the users' rates become arbitrarily large, resulting in serious congestion collapse.

We now show that the system in (14) has only one unique equilibrium, which is locally exponentially stable, and semi globally exponentially stable, i.e. it is exponentially stable as long as $n(t)$ and $x(t)$ are constrained to lie in a compact set[5]. Further, the unique equilibrium solves a concave optimization problem. We show this through the following theorem.

Theorem 1: For arbitrary $\beta > 0$, the approximate system in (14) has a unique equilibrium, denoted by (x^*, n^*) as

$$\begin{aligned} n_r^* &= \frac{1}{f(\sum_{j\in r} g_j(y_j^*))}, \quad r \in R; \\ x_r^* &= \frac{\sqrt{2}S}{f(\sum_{j\in r} g_j(y_j^*))T_r\sqrt{\sum_{j\in r}\left[g_j\left(y_j^*\right)+\epsilon_j\right]}}, \quad r \in R. \end{aligned} \tag{18}$$

Further, this unique equilibrium solves the following concave optimization problem

$$\max_{x\geq 0} \sum_{r\in R} U_r(x_r) - \sum_{j\in J}\int_0^{y_j} g_j(z)\, dz, \tag{19}$$

with U_r being concave function:

$$U_r(x_r) = \int_0^{x_r} h_r^{-1}\left(\frac{2S^2}{T_r^2\nu^2}\right) d\nu, \quad r \in R,$$

where h_r^{-1} is the inverse of an monotonically increasing function h_r:

$$h_r(z) \triangleq \left(\sum_{j\in r}\epsilon_j + z\right) f(z) = \left(\sum_{j\in r}\epsilon_j + z\right)\frac{e^{\beta z}-1}{e^{\beta z}+1}, \quad r \in R.$$

Proof: Refer to Appendix A. ■

One observation to be made from Theorem 1 is that the unique equilibrium for the system in (14) in wireless scenario solves a concave optimization problem similar to the one TCP Reno solves in the wired network. They have the same form as (1) but with different users' utility functions $U_r(x_r)$. The only difference is that the $U_r(x_r)$ in the wired case is only a function of x_r, while in wireless scenario it is also a function of $\sum_{j\in r}\epsilon_j$, i.e. the packet loss rate associated with the route r. In fact, if we let $\beta \to \infty$ and $\epsilon_j = 0, \forall j \in J$, i.e. in the wired network scenario, we have $h_r(z) = z$, and the optimization

[5]Note $n(t)$ and $x(t)$ are constrained in practice anyway

problem in (19) becomes identical to the TCP optimization problem in (4). In this case, the equilibrium (x^*, n^*) is exactly the same as x^o, implying TCP optimization problem in the wired network is merely a special case of that in (19).

Given the system in (14) has a unique optimal equilibrium, an important question to answer is that whether it is stable, i.e. will the users' rates converge to it asymptotically. The following two theorems explore the answer to this question.

Theorem 2: For arbitrary $\beta > 0$, under the two timescale assumption, the unique equilibrium of the reduced system in (16) is locally exponentially stable. Combining the known fact that the boundary system is globally exponentially stable, with the same arguments used in [24] and [25] for the stability of singular disturbance non-linear system, the unique equilibrium of the approximate system in (14), (x^*, n^*), is locally exponentially stable.

Proof: Refer to Appendix B. ■

Theorem 2 implies that if the number of connections $n(t)$ is initially in a small ball around the equilibrium n^*, then the entire system will converge exponentially fast to the equilibrium. As no convergence can be faster than exponential, this is the best result one can expect in a local region around the equilibrium.

How about when $n(t)$ starts far away from the equilibrium n^*? To answer this question, we explore the global stability of the equilibrium. The following theorem states that if n and x are constrained to a compact set, the system is exponentially stable. In practice, as $n(t)$ and $x(t)$ take only finite values, we conclude this theorem is sufficient to claim that $(n(t), x(t))$ converges to the optimal equilibrium given arbitrary initial conditions.

Theorem 3: For arbitrary $\beta > 0$, under the two timescale assumption, the unique equilibrium of the reduced system in (16) is semi globally exponentially stable; hence the unique equilibrium of the approximate system in (14), (x^*, n^*), is semi globally exponentially stable.

Proof: Refer to Appendix C. ■

The intuition behind both the local and global convergence of the entire system is as follows: $x_r(t)$ first converges in the fast timescale to the equilibrium of the boundary system (15), defined by the equilibrium manifold as in (17); then in a slow timescale , $n_r(t)$ and $x_r(t)$ follow the control laws of the reduced system in (16) to converge to the optimal equilibrium along the manifold. An important consequence of this convergence argument is that, a combination of control law (10) on $n_r(t)$ and any flow control method on $x_r(t)$ resulting in the same equilibrium manifold as TCP Reno, will retain the convergence behavior shown in Theorems 2 and 3. Therefore, it is possible to extend all these results to TFRC since it has been shown in [2] that TFRC has the same stationary behavior as TCP Reno.

Remarks. Theorems 1, 2 and 3 state the existence of a unique optimal equilibrium, and ensure its stability for the continuous approximate system in (14), for arbitrary values of β. In the limit as $\beta \to \infty$, the approximate system approaches the original discontinuous system in (9) and (10). Therefore, for extremely large β, we expect the approximate system to behave quite similarly to the original system, except at the discontinuities $y_j(t) = C_j$.

As seen in the next section, in actual implementation of the proposed system in (9) and (10), it is necessary to discretize continuous quantities. For instance, controlling $n_r(t)$ is implemented by adjusting the number of connections, which has to be an integer number; controlling $x_r(t)$ is implemented by TCP to adjust the finite number of packets to be sent out in a time interval. Therefore, it is highly unlikely for the system to operate at discontinuous points. From this point of view, the analysis based on the approximate system is accurate enough to predict and interpret the performance of the actual implementation of the algorithm in practice.

Note that the c_r can be chosen arbitrarily in system (14), as long as the two timescale decomposition holds. Practically, this implies that each user can adjust $n_r(t)$ according to a different rate. Specifically, a global setting among all the users, as for instance used in our simulations, is not necessary. Furthermore, allowing some of the c_r to be equal to zero represents a scenario according to which the proposed scheme coexists with TCP. In this situation, all theorems still hold, except for a modification to Theorem 1. More precisely, the utility for users applying TCP should be $U_r(x_r) = 2S^2/(T_r^2 x_r^2)$, rather than the one defined in the theorem.

Our results imply that in a network where our schemes coexist with TCP, all users' rates will again converge, semi-globally exponentially, to a unique optimal equilibrium. These observations on stability and scalability encourage incremental deployment of our scheme in the current Internet where TCP is dominant, and directly addresses a major concern. The simulations in Section V partially support this observation.

In summary, the equilibrium x^* of the approximate system in (14), not only solves the optimization problem in (19), but also enjoys several desirable properties: first, all network bottlenecks are fully utilized, yet the network is free of congestion collapse, for *arbitrary* topology, *arbitrary* initial sending rates, and *arbitrary* number of users applying either proposed solution or TCP; second, users' rates globally exponentially converge to a unique and optimal equilibrium, resulting in no rate fluctuation in stationary state. Hence all of our design objectives are met.

Since all of the above analysis and results hold regardless of the values of $\epsilon_j, j \in J$, it is possible to design only one practical scheme, following (9)-(10), for both wireless and wired networks, with the latter corresponding to the case where $\epsilon_j = 0, j \in J$.

IV. Enhanced MULTCP (E-MULTCP)

In this section, we design a practical scheme for TCP-based data transmission over wireless networks, based on our analysis and control law in (10), by adjusting the number of TCP connections. Although the analysis in previous section is based on TCP model, it can be extended to TFRC, since it has been shown TFRC has the same stationary behavior as TCP [2].

The framework of our proposed E-MULTCP is shown in Figure 2. As seen, there are two components in the system: RTT Measurer Sub-system (RMS), and Connections Controller Sub-system (CCS).

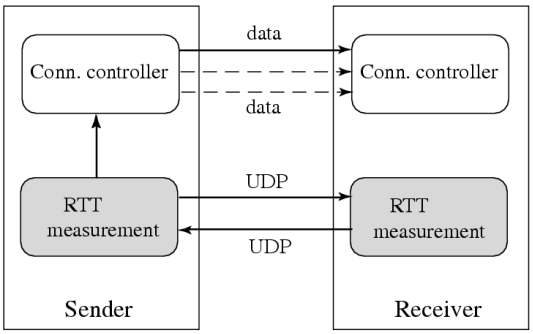

Fig. 2. E-MULTCP system framework.

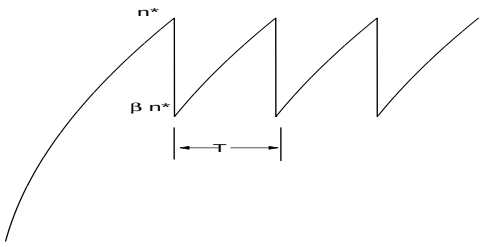

Fig. 3. Demonstration of the change on the number of connections n, controlled by single E-MULTCP over single wireless link.

A. RTT Measurement Sub-system (RMS)

The gray blocks in Fig. 2 represent RMS. They measure round trip time samples between sender and receiver, denoted by rtt_{sample}, by computing difference between the time sender emits a packet, and the time it receives the ACK from receiver. These packets are sent through the forward and backward UDP connections between sender and receiver.

To reduce overhead and to prevent congestion collapse, the rate at which RTT-measurement packets are sent, is set to be the same as that of data packets, which is estimated on sender side by measuring the number of data packets sent in the previous round trip time. The RTT measurement packets contain only an IP header and a timestamp, a total of 28 bytes. In the case when TCP uses typical packet size of 1500 bytes, the RTT measurement overhead is 2% of the total throughput.

After waiting for a fixed interval, denoted by τ, RMS computes the running average, ave_rtt, of these rtt_{sample}s, and reports it to the CCS. Clearly, $1/\tau$ is the frequency of adjusting number of connections; τ has to be large enough to ensure the frequency is much lower than that of changing the source rate, which is typically of the order of round trip time.

B. Connection Controller Sub-system (CCS)

The CCS is shown as the white blocks in Figure 2. Its basic functionality is to properly control the number of connections n, following the control law shown in (10). As seen in (10), when route r is underutilized, the indicator function $I_r(\cdot) = 0$, and n_r increases proportional to $1/n_r$. When route r is congested, $I_r(\cdot) = 1$, and n_r will decrease roughly proportional to itself. This indicates CCS at the sender should roughly Inversely Increase and Multiplicatively Decrease (IIMD(α, β)) the number of connections n, based on route congestion status. Specifically, ave_rtt is reported to CCS by RMS, CCS sets the rtt_min as the minimum over all ave_rtt seen so far, and then adapts the number of connections n as follows:

$$n = \begin{cases} \beta n + \alpha/n, & \text{if } ave\_rtt - rtt\_min > \gamma rtt\_min; \\ n + \alpha/n, & \text{otherwise.} \end{cases} \tag{20}$$

where $\alpha = 1 - \beta < 1$ and γ is a preset parameter. This is nothing but a discrete implementation of the control law in (10), with $c_r = \alpha/\tau$. The value of the indicator function $I_r(\cdot)$ is estimated by comparing the measured queuing delay $ave\_rtt - rtt\_min$ with a dynamic threshold γrtt_min. For a given route, the rtt_min is a constant representing the minimum observed round trip time for that route, approximating physical propagation delay. As such, $ave\_rtt - rtt\_min$ corresponds to current queuing delay, and γrtt_min is a threshold on the queuing delay that E-MULTCP can tolerate before it starts to decrease the number of connections. Therefore, under ideal conditions, E-MULTCP keeps increasing the number of connections to make ave_rtt as close as possible to $(1+\gamma)rtt\_min$ without exceeding it.

C. Discussion

In the increasing stage, i.e. $I(ave\_rtt - rtt\_min > \gamma rtt\_min) = 0$, E-MULTCP has an increasing rate of $\dot{n}(t) = \alpha/(\tau n(t))$. Therefore, to increase n from N_2 to N_1, assuming $N_1 > N_2$, it roughly takes E-MULTCP $\frac{\tau}{2\alpha}(N_1^2 - N_2^2)$ seconds. On the other hand, the decreasing rate for the decreasing stage is roughly $\dot{n}(t) = -\frac{\alpha n(t)}{\tau}$, and hence it takes E-MULTCP $\frac{1}{\alpha}\tau \ln(N_1/N_2)$ seconds to decrease n from N_1 to N_2. Thus, E-MULTCP is conservative in adding connections, but aggressive in closing them.

In a simple topology with one E-MULTCP system over one wireless link, we can use the above results to compute bandwidth utilization ratio. At steady state, E-MULTCP periodically increases the number of connections n to the optimal n^* that fully utilizes the wireless bandwidth, and proportionally decreases n upon reaching n^*. The approximated continuous version of this process is demonstrated in Fig. 3. In the plot, T is the time for $n(t)$ to increases from βn^* to n^*, and hence is $(n^*)^2(1-\beta^2)\tau/2\alpha$. The number of connections $n(t) = \sqrt{(t-kT)2\alpha/\tau + (\beta n^*)^2}$ for $kT \le t \le (k+1)T$ and $k \in Z^+$.

The utilization ratio, at steady state, is then the ratio between the average number of connections and n^*, as follows:

$$\frac{1}{Tn^*}\int_0^T n(\delta)d\delta = \frac{2}{3}\frac{1+\beta+\beta^2}{1+\beta}. \tag{21}$$

This ratio is at least 2/3, only depends on β, and is independent of wireless packet loss rate, τ, and n^*. As $\beta \to 1$, the utilization ratio in (21) tends to 1. Hence, large values of β improve the utilization ratio in this scenario. However, since $\alpha = 1 - \beta$, large values of β will also result in slow convergence rate, as indicated by the above results using NS-2 simulations. We have empirically found $\beta = 0.75$ to provide reasonable compromise between utilization and convergence. For $\beta = 0.75$, the utilization ratio is about 0.88.

When there is a route change either due to change in the wireless base station, or due to route change within the wired networks, the value of rtt_min changes significantly, affecting the performance of E-MULTCP. Under these conditions, it is conceivable to use route change detection tools such as traceroute [26] at the sender to detect the route change, in order to reset rtt_min to a new value. Furthermore, it can be argued that the overall throughput of E-MULTCP will not go to zero, resulting in starvation; this is because E-MULTCP always keeps at least one connection open.

Since the data stream in E-MULTCP is transmitted using multiple connections, the receiver could potentially receive out of order packets. If the receiver uses a buffer to reorder the arriving packets, the out of order arrived packets can be reordered according to their sequence numbers.

In summary, in the E-MULTCP system, the number of connections is controlled according to a discrete version of (10) at the sender; the sending rate of each TCP connection is adjusted automatically by itself. The rate of change of number of connections is expected to be much slower than that of sending rate satisfying the two time scale assumption in the previous section. Thus, the optimality and stability analysis in the previous section for the dynamic system in (9) and (10) applies to E-MULTCP, indicating that E-MULTCP results in a stable, yet fully utilized network as shown in (19).

Notice that E-MULTCP is similar to our previously proposed scheme MULTFRC [5] in its design. The differences are two fold. First, E-MULTCP is based on TCP, while MULTFRC is based on TFRC. Second, E-MULTCP applies IIMD as the control law for number of connections n, while MULTFRC applies IIAD. It is of course conceivable to design an Enhanced MULTFRC (E-MULTFRC) based on TFRC by using IIMD to control the number of connections [27]. A fundamental and inherent advantage of E-MULTCP and E-MULTFRC over MULTFRC are their provable convergence and stability properties as shown in the previous section.

V. NS-2 SIMULATIONS AND 1XRTT WIRELESS EXPERIMENTS

In this section, we carry out NS-2 [28] simulations and actual experiments over Verizon Wireless 1xRTT CDMA data network to evaluate the performance of E-MULTCP. We use TCP NewReno implementation for TCP protocol in all simulations. We have empirically found the performance of E-MULTFRC to be comparable to that of E-MULTCP [27].

A. Setup

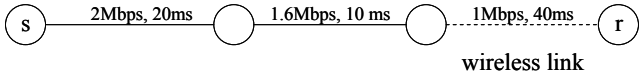

Fig. 4. Simulation topology.

The topology used in simulations is shown in Figure 4. The sender denoted by s, and the receiver denoted by r, both run E-MULTCP at the application layer. The number of connections, as mentioned in the previous section, is controlled by the sender. For all simulations, the wireless bandwidth B_w is set be 1 Mbps, and is assumed to be the bottleneck. The wireless link is modeled by an exponential error model, and p_w varies from 0.0 to 0.08 in increments of 0.02. DropTail type queue is used for each node. In order to evaluate E-MULTCP's performance in the presence of wireless channel errors, we examine three issues; first, how E-MULTCP performs in terms of average throughput, average round trip time, and packet loss rate, as a function of p_w. Second, whether the number of connections is stable. Third, whether or not a E-MULTCP application can fairly share with an application using one TFRC or one TCP connection. In all the simulations, throughput is measured every second, packet loss rate is measured every 30 seconds, the average round trip time is measured every 100 packets, and the number of connections is sampled whenever there is a change.

For the actual experiments over 1xRTT, we stream from a desktop connected to Internet via 100 Mbps Ethernet in eecs.berkeley.edu domain, to a notebook connected to Internet via Verizon Wireless 1xRTT CDMA data network. Thus it is quite likely that the last 1xRTT CDMA link is the bottleneck for the connection. The connection is live for 30 minutes. As we cannot control p_w in actual experiments, we measure the average throughput, average number of connections, and packet loss rate.

B. Performance Characterization of E-MULTCP

We have empirically found the following parameters to result in reasonable performance: $\alpha = 0.25, \beta = 0.75$, $\gamma = 0.2$, and $\tau = 20$ seconds. Intuitively, larger τ results in more reliable estimates of round trip time, but at the expense of a lower sampling frequency, resulting in a less responsive system. Larger γ results in a system that is more robust to round trip time estimates, but at the expense of longer queues in routers, and hence a longer queueing delay. Larger α results in a faster convergence rate to full utilization, but at the expense of a slower response to congestion. We have determined these values empirically through simulations and experiments.

We simulate the E-MULTCP system to send data for 9000 seconds, compute the average throughput and packet loss rate for p_w =0.0, 0.02, 0.04, 0.06 and 0.08[6], and compare them to the optimal, i.e. $B_w(1 - p_w)$ for each p_w. The results for $B_w = 1\ Mbps$ and $RTT_{min} = 168\ ms$ are shown in Figure 5. As seen, the throughput is within 25% of the optimal, and is reasonably close to the predicted one, i.e. the product of the optimal and the utilization ratio computed using (21). The round trip time is within 20% of RTT_{min}, and the packet loss rate is almost identical to the optimal, i.e. a line of slope one as a function of wireless channel error rate. As expected, the average number of connections increases with wireless channel error rate, p_w. [7]

[6]It is well known that TCP's degraded performance is dominated by timeout if wireless packet loss rate p_w is higher than 0.1. As our model does not capture the timeout effect, we are more interested in cases where $p_w < 0.1$.

[7]Note the round trip time for $p_w = 0$ is not shown in Figure 5 because it represents the case when the channel is error free. In this case, E-MULTCP reduces to one TCP connection.

Considering the throughput plot in Figure 5, for some values of p_w, there is a significant difference between the actual and optimal throughput. This is due to the quantization effect in situations where the number of connections is small, i.e. 2 to 4. In these situations, a small oscillation around the optimal number of connections results in large variation in observed throughput. One way to alleviate this problem is to increase γ in order to tolerate larger queuing delay and hence absorb throughput fluctuations, at the expense of being less responsive. Another alternative is to use smaller packet size in order to reduce the "quantization effect" at the expense of (a) larger overhead and hence lower transmission efficiency, and (b) the slower rate of convergence to the optimal number of connections. In the case of TFRC, we have shown that it is possible to combine multiple TFRC connections into one connection, in order to mitigate the quantization effect [14]. It is also possible to combine multiple TCP connections into one UDP connection whose rate emulates the aggregate rate of multiple TCP connections.

One might also notice that the difference between E-MULTCP's throughput and the optimal becomes larger as p_w increases. This is due to increased timeout events as p_w increases. Upon timeout, TCP slow starts again, generating bursty traffic that results in transient queuing delay. Consequently, detecting congestion based on RTT estimate becomes less reliable, decreasing the average throughput. This effect is more pronounced for large p_w and number of connections.

In order to examine E-MULTCP's performance as a function of p_w, as well as the dynamics of E-MULTCP, we use E-MULTCP with p_w initially set at 0.02. Then at 3000^{th} second, p_w is switched to 0.06, and at 6000^{th} second switched back to 0.02. Here, we artificially change p_w to see how E-MULTCP adapts to the change in p_w. The throughput, packet loss rate, round trip time and the number of opened connections are shown in Figure 6. As seen, the number of connections varies from around 2-3 to around 5 as p_w switches from 0.02 to 0.06. The ramp up and ramp down times from 2 to 5, and 5 to 2 connections are about 660 and 80 seconds, respectively. These are very close to the theoretical values obtained from Section IV-C, i.e. 650 and 73 seconds respectively.

Similar experiments are carried out on Verizon Wireless 1xRTT CDMA data network. The 1xRTT CDMA data network is advertised to operate at data speeds of up to 144 kbps for one user. As we explore the available bandwidth for one user using UDP flooding, we find the highest average available bandwidth averaged over 30 minutes to be between 80 kbps to 97 kbps. In our experiments, we stream for 30 minutes from a desktop on wired network in eecs.berkeley.edu domain to a laptop connected via 1xRTT CDMA modem using E-MULTCP, E-MULTFRC and TCP. The results are shown in Table I for packet size of 1460 bytes. As seen, on average E-MULTCP opens up 1.7 connections, and results in 58% higher throughput, at the expense of a larger round trip time, and higher packet loss rate. Similar observations can be made for E-MULTFRC.

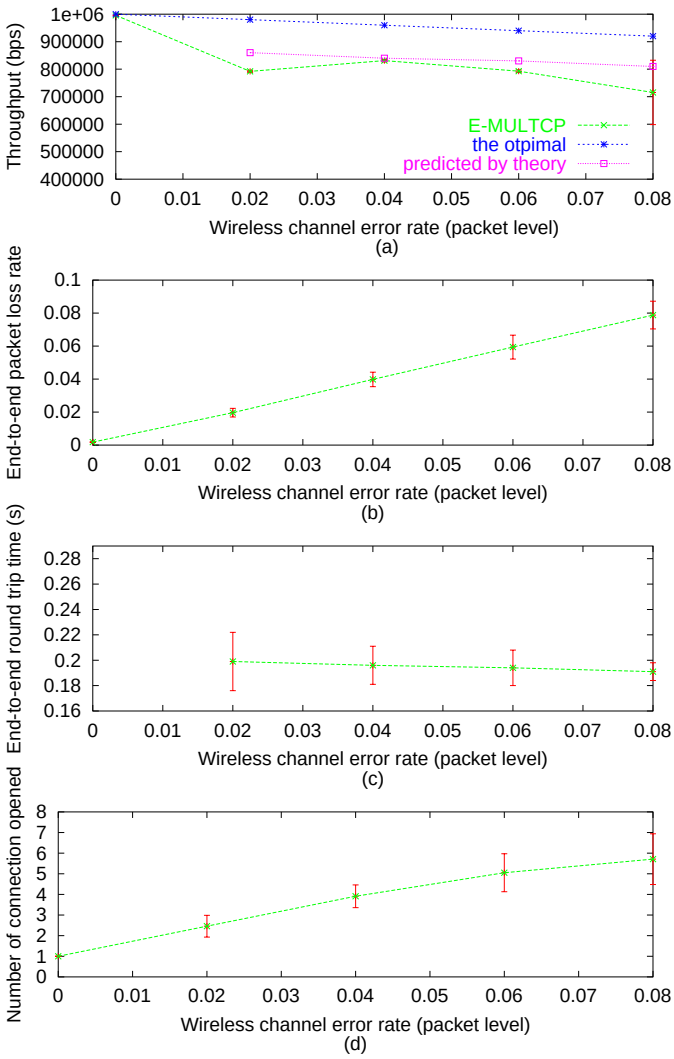

Fig. 5. NS-2 simulations for $B_w = 1\ Mbps$ and $RTT_{min} = 168\ ms$; (a) throughput, (b) end-to-end packet loss rate, (c) end-to-end round trip time, (d) number of connections, all as a function of packet level wireless channel error rate.

C. Fairness between E-MULTCP and TCP

To investigate the fairness of E-MULTCP, we carry out NS-2 simulations based on the "dumbbell" topology shown in Fig. 7. Senders are denoted by $si, i = 1, \ldots, 16$, and receivers are denoted by $di, i = 1 \ldots, 16$. We investigate two types of fairness: the inter-protocol fairness between E-MULTCP and TCP, and the intra-protocol fairness within E-MULTCP.

The intra-protocol fairness is defined as the fairness between E-MULTCP flows. In our simulations, we run E-MULTCP on all 16 sender-receiver pairs shown in Fig. 7 for 5000 seconds, and compare their throughput. E-MULTCP is said to be intra-protocol fair if all receivers achieve the same throughput. The fairness ratios for $p_w = 0.01$ and $p_w = 0.04$ are shown in Table II. The fairness ratio is defined as receivers' throughput divided by the average throughput; the closer to one, the more fair the E-MULTCP system is. As seen, the fairness ratio is fairly close to one, indicating E-MULTCP flows are fair to

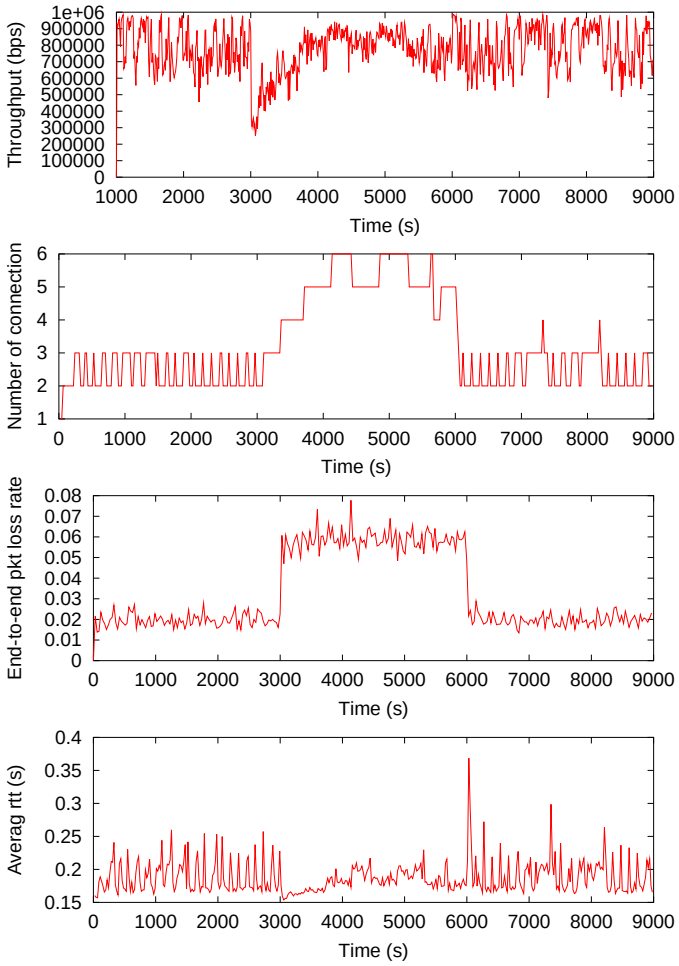

Fig. 6. NS-2 simulation results as p_w changes from 0.02 to 0.06 and back again; (a) end-to-end round trip time, (b) throughput, (c) number of connections, (d) end-to-end packet loss rate, all as a function of time.

TABLE I

ACTUAL EXPERIMENTAL RESULTS FOR E-MULTCP AND E-MULTFRC SYSTEMS OVER 1XRTT CDMA.

scheme	**throughput (kbps)**	**rtt (ms)**	**ave. # of conn.**
one TCP	59	1954	N/A
E-MULTCP	93	2447	1.7
E-MULTFRC	89	2767	1.9

each other, at least in this simulation setting. The bandwidth utilization ratios are 96% for $p_w = 0.01$ and 98% for $p_w = 0.04$.

The inter-protocol fairness is defined as the fairness between E-MULTCP and TCP. In our simulations, we run E-MULTCP on the first 8 sender-receiver pairs, i.e. $(si, di), i = 1, \ldots, 8$, and TCP on the remaining 8 sender-receiver pairs shown in Fig. 7; each session lasts 5000 seconds, and we compare their throughput for $p_w = 0.01$ and $p_w = 0.02$. Under the simulation settings, each E-MULTCP consumes more bandwidth than one TCP under full utilization. This is because the wireless channel error rate is large enough to make the number of virtual connections of each E-MULTCP to be larger than

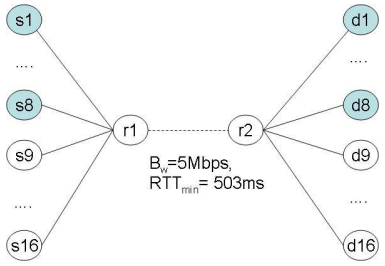

Fig. 7. The simulation topology for E-MULTCP's fairness evaluation.

TABLE II

SIMULATION RESULTS FOR INTRA-PROTOCOL FAIRNESS OF E-MULTCP.

receiver	fairness ratio p_w=0.01	fairness ratio p_w=0.04	receiver	fairness ratio p_w=0.01	fairness ratio p_w=0.04
d1	1.06	1.03	d9	1.07	0.98
d2	1.00	1.01	d10	0.97	0.98
d3	1.06	1.03	d11	1.02	1.02
d4	1.02	1.08	d12	1.02	0.99
d5	0.90	0.98	d13	0.98	0.97
d6	0.93	0.98	d14	0.89	1.00
d7	1.02	1.04	d15	1.01	1.01
d8	0.99	1.01	d16	0.98	0.94

one. Hence, it is meaningless to define the fairness between E-MULTCP and TCP as having the same throughput.[8] As such, in our simulations, we define E-MULTCP to be fair to TCP if it does not result in a significant decrease in TCP's throughput as compared to TCP throughput in the absence of E-MULTCP. Specifically for our simulations, it implies TCP retains the same throughput whether or not it coexists with E-MULTCP under the same network setting. The throughput of E-MULTCP and TCP, as well as the total bandwidth utilization ratios for the setup shown in Fig. 7, are shown in Table III for two scenarios: (a) 8 E-MULTCP coexisting with 8 TCP connections, (b) 16 TCP connections. Figs. 8 and 9 also show the dynamics of throughput, packet loss rate, RTT, and the number of virtual connections n for the case where 8 TCP connections coexist with 8 E-MULTCP connections. Comparing E-MULTCP+TCP with TCP-alone in Table III, we see the former achieves a much higher utilization of the wireless bandwidth at the expense of slightly lower TCP throughput. A careful examination of Fig. 8 reveals that this throughput drop is caused by the higher RTT experienced by TCP connections in the E-MULTCP+TCP scenario as compared with TCP-alone scenario. For example, for $p_w = 0.01$ and $\gamma = 0.2$, The RTT for E-MULTCP+TCP is around 0.56 seconds, while for TCP-alone is 0.5 seconds, i.e. the propagation delay[9]. As TCP's throughput is known to be inversely proportional to RTT, the 12% increase in the RTT of TCP connections roughly explains the 11% decrease in the TCP's throughput shown in first row in Table III.

[8]Obviously, there are situations in which E-MULTCP ends up with performing similar to one TCP. An example would be E-MULTCP competing for bandwidth with TCP on wired networks. In that case, however, the fairness between E-MULTCP and TCP is reduced to the fairness between TCP connections themselves, and has been well explored in literature.

[9]In this NS simulation, TCP and E-MULTCP share the same route and hence both have the same round trip time.

TABLE III

SIMULATION RESULTS FOR FAIRNESS BETWEEN E-MULTCP AND TCP.

settings	8 E-MULTCP + 8 TCP			16 TCP	
	ave. thput. (E-MULTCP) (kbps)	ave. thput. (TCP) (kbps)	utilization (%)	ave. thput. (TCP) (kbps)	utilization (%)
p_w=0.01 γ=0.2	432.1	160.3	96	179.8	58
p_w=0.01 γ=0.1	402.3	170.0	93	179.8	58
p_w=0.02 γ=0.2	468.8	107.8	94	120.4	40
p_w=0.02 γ=0.1	446.1	114.3	92	120.4	40

This increase in the RTT experienced by TCP is, by design, a consequence of E-MULTCP controlling n according to (20). As n is only decreased after the queuing delay exceeds the threshold γrtt_min, round trip time is increased when E-MULTCP increases n to achieve full utilization. One way to address this problem is to use a smaller value for γ, in order to reduce the increase in the RTT, and hence minimize the TCP's throughput drop. However, smaller values of γ also result in lower bandwidth utilization due to increased sensitivity of E-MULTCP to RTT measurements. As shown in Table III, decreasing γ from 0.2 to 0.1 results in both a smaller drop in the TCP's throughput and lower utilization. Regardless, the stability of the network with mixed TCP and E-MULTCP is guaranteed by Theorem 3.

VI. DISCUSSION AND FUTURE WORK

In this paper, we have formulated flow control problem in wireless networks as a general concave optimization problem, of which Kelly's optimization problem can be shown to be a special case. This reformulation results in a new class of end-to-end based solutions, in which an appropriate number of connections are opened at the application layer by the sender. The solutions require only one bit of information on whether or not the route is congested, making it easy to estimate accurately at the application layer. Hence no modification to either existing protocols, e.g. TCP, or infrastructure, e.g. routers, is expected. We have shown that our proposed scheme has a unique stable equilibrium that solves the concave optimization problem, implying the scalability and optimality of the solution. Specifically, the unique optimal equilibrium is semi-globally exponentially stable. Stability, optimality and scalability of proposed solution coexisting with TCP can also be inferred in a straightforward fashion.

In practice, these results guarantee all network bottlenecks are fully utilized, yet the network is free of congestion collapse, for *arbitrary* topology, *arbitrary* initial sending rates, and *arbitrary* number of users applying either proposed solution or TCP. Furthermore, users' rates globally exponentially converge to a unique and optimal equilibrium, resulting in no rate fluctuation in stationary state.

To demonstrate the generality of our proposed solution, we have developed a practical scheme called E-MULTCP for data transmission over the wireless network. Its efficient

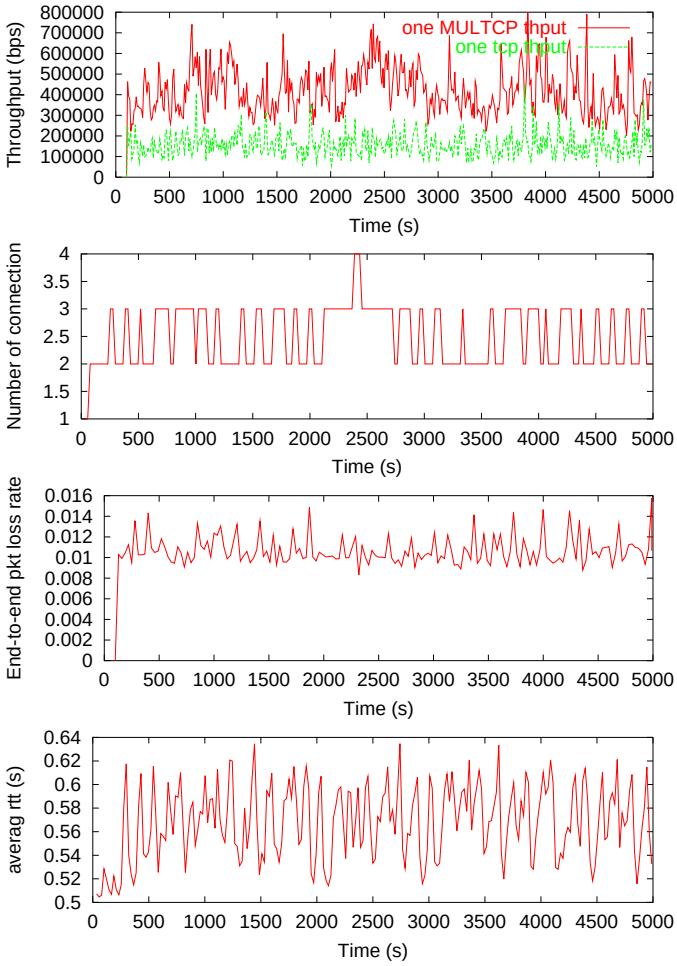

Fig. 8. NS-2 simulation results for the case $p_w = 0.01, \gamma = 0.2$ for E-MULTCP+TCP scenario: the dynamics of (a) throughput, (b) number of connections , (c) end-to-end packet loss rate, (d) end-to-end RTT, all as a function of time.

performance, and fairness to both itself and TCP are characterized and evaluated using both NS-2 simulations and 1xRTT wireless experiments. Analysis and simulation results also indicate E-MULTCP in fact works in both wired and wireless scenarios. The same way as we have extended TCP to E-MULTCP, it is possible to extend TFRC to E-MULTFRC by using IIMD control law in order to control the number of TFRC connections [27]. In the E-MULTFRC case, we have shown that it is also possible to overcome the undesirable consequence of quantization of the number of connections by combining them into one [14].

E-MULTCP is different from existing schemes such as [8] that use delay or round trip time variation to infer congestions in several ways: first E-MULTCP measures the round trip time variations over a large time window, while others measure the round trip time variations instantaneously, resulting in more noisy measurements; second, E-MULTCP uses the measured variation to adapt the number of opened connections in order to adapt the rate, while the existing schemes use it to differentiate between congestion loss and wireless loss.

Even though C_j and ϵ_j are assumed to be constant in

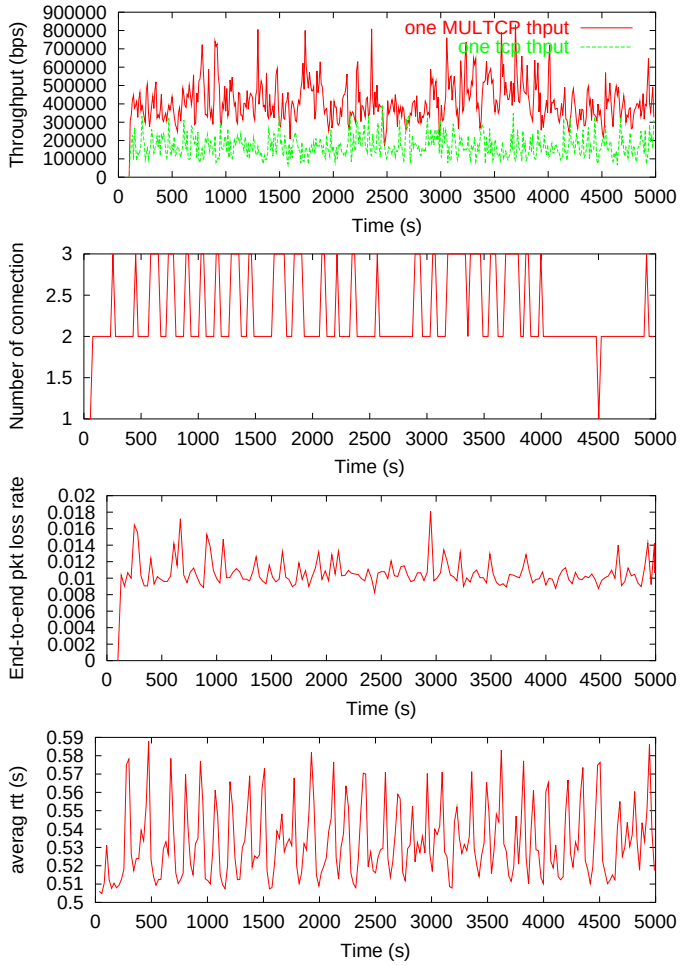

Fig. 9. NS-2 simulation results for the case $p_w = 0.01, \gamma = 0.1$ for E-MULTCP+TCP scenario: (a) throughput, (b) number of connections , (c) end-to-end packet loss rate, (d) end-to-end RTT, all as a function of time.

our analysis, in some networks such as wireless Local Area Networks (WLAN) and CDMA networks, C_j and ϵ_j might be time varying or even change in a correlated fashion. E-MULTCP adjusts the number of connections in a timescale much slower than that of each connection's sending rate, in order to satisfy the two timescale assumption, as well as to reduce the complexity and overhead of opening/closing connections. As a result, in a scenario where C_j and ϵ_j are time varying, by design, E-MULTCP could not react instantaneously. Hence, it is interesting to quantify how fast E-MULTCP can adapt to the changes in C_j and ϵ_j. In practice, our experimental results in this paper have verified that our proposed scheme works in an actual CDMA network, where the C_j and ϵ_j are typically not constant.

Currently, E-MULTCP relies on accurate detection of queuing delay along the route, which in turn requires knowledge of the propagation delay. In situation where propagation delay is highly volatile, such as in 3G wireless networks that deploy opportunistic scheduling, E-MULTCP's performance could potentially degrade due to inaccurate estimation of queuing delay. As discussed in the beginning of Section V-B, even though it is possible to choose a large γ to tolerate inaccuracy in estimating queuing delay, it would be desirable to develop a more reliable way to estimate the congestion status of route, i.e. the one bit of information needed by E-MULTCP.

There are many other interesting and important directions for future research. Stability in the presence of delay and noisy disturbances are of great interest, from both control and the networking points of view. The fairness properties of the equilibrium rates are also of interest, and might explain the reason the equilibrium rates are acceptable in terms of fairness in various scenarios combining wired and wireless networks.

Our framework captures the impact of wireless channel packet loss on TCP and TFRC performance over wireless. However, it does not model the timeout effect of TCP and TFRC over wireless possibly caused by link layer local retransmissions or heavy wireless loss, potentially degrading the performance. It is desirable to extend our framework in order to take into account the effect of timeout in wireless networks.

Last but not the least, it is interesting to examine whether it is possible to use a different utility maximization problem that leads to another fundamentally different solution for the TCP and TFRC over wireless network problem.

ACKNOWLEDGEMENT

The authors would like to express their thanks to Alessandro Abate and Shankar Sastry for helpful discussions and help in preparing the paper.

APPENDIX

A. *Proof of Theorem 1*

First note (x^*, n^*) is the equilibrium of the system in (14), and x^* is also the solution for the optimization problem in (19), which can be seen by setting the derivative to zero. By the definition of $h_r(z)$ and $g_j(z)$, it is not difficult to see the objective function in (19) is a concave function. Note the constraint set of the optimization problem shown in (19) is convex, and that the optimization problem is in fact a concave optimization problem. Hence it has a unique solution, which is in fact x^*. Since the equilibrium must lie on the equilibrium manifold in (17), we know the unique x^* leads to a unique n^*. Therefore, the equilibrium (x^*, n^*) exists and is unique.

B. *Proof of theorem 2*

Proof: Let $G(x) = diag(\frac{(x_r)^2 T_r^2}{2S^2})$ and

$$D(x) = diag\{\frac{1}{x_r}\sum_{j\in r}[\epsilon_j + g_j(y_j)]\} + \frac{1}{2}A^T diag\{g_j'(y_j)\}A,$$

where A is the connectivity matrix defined in Section II-A. We can have the relation between $\dot{x}$ and $\dot{n}$ on the equilibrium manifold as:

$$G(x)D(x)\dot{x} = diag\{n_r\}\dot{n},$$

Combining the dynamics of n in the reduced system, we can rewrite the above equation as:

$$\dot{x} = cD^{-1}(x)G^{-1}(x)\{1 - \frac{T_r^2}{2S^2}x_r^2\sum_{j\in r}[\epsilon_j + g_j(y_j)]f(\sum_{j\in r} g_j(y_j))\}.$$

Around the equilibrium of the reduced system, let $x_r(t) = x_r^* + z_r(t)$, denote $D(x^*)$ as $\tilde{D}$ and $G(x^*)$ as $\tilde{G}$; after linearization, we have that, $\forall r \in R$,

$$\begin{aligned}\dot{z}(t) &= -c\tilde{D}^{-1}\tilde{G}^{-1}\left[2diag\quad f(\sum_{j\in r} g_j(y_j^*))\right)\tilde{G}\tilde{D}+ \\ &\quad \tilde{G}\cdot diag\ \sum_{j\in r}\ \epsilon_j + g_j(y_j^*)\quad \cdot \\ &\quad diag\quad f'(\sum_{j\in r} g_j(y_j^*))\right)A^T diag(g_j'(y_j^*))A\Big]\, z(t) \qquad (22)\end{aligned}$$

where $g_j'(y_j^*) = \frac{C_j}{(y_j^*)^2}\frac{e^{\beta\frac{y_j^*-C_j}{y_j^*}}}{1+e^{\beta\frac{y_j^*-C_j}{y_j^*}}} > 0, \quad j \in J.$

Denote $E = 2diag\left(f(\sum_{j\in r} g_j(y_j^*))\right)\tilde{G}\tilde{D} + \tilde{G}\cdot diag\big(\sum_{j\in r}\left(\epsilon_j + g_j(y_j^*)\right)\big)diag\left(f'(g_j(y_j^*))\right)A^T diag(g_j'(y_j^*))A$. Then by simple linear algebra arguments, the system in (22) is stable if and only if $\tilde{D}^{-1}\tilde{G}^{-1}E$ has all positive eigenvalues. We now show that this requirement is verified.

First note that this is equivalent to showing $E\tilde{D}^{-1}\tilde{G}^{-1}$ has all eigenvalues be positive since $E\tilde{D}^{-1}\tilde{G}^{-1}$ is similar to $\tilde{D}^{-1}\tilde{G}^{-1}E$. But

$$\begin{aligned}E\tilde{D}^{-1}\tilde{G}^{-1} &= 2\tilde{G}\cdot diag\quad f'(\sum_{j\in r} g_j(y_j^*))\frac{[\sum_{j\in r}(\epsilon_j + g_j(y_j^*))]^2}{x_r^*}\Big) \\ &\cdot\Big\{\cdot diag\quad \frac{x_r * f(\sum_{j\in r} g_j(y_j^*))}{f'(\sum_{j\in r} g_j(y_j^*))[\sum_{j\in r}(\epsilon_j + g_j(y_j^*))]^2}\Big) \\ &+ \frac{1}{2}diag\ \frac{x_r^*}{\sum_{j\in r}[\epsilon_j + g_j(y_j^*)]}\ A^T diag(g_j'(y_j^*))A\tilde{D}^{-1}\Big\}\tilde{G}^{-1},\end{aligned}$$

Define the terms inside the curly brackets as B for later usage. $A^T diag(g_j'(y_j^*))A\tilde{D}^{-1}$ is a product of a positive definite matrix and a (semi)-positive definite matrix, having all nonnegative eigenvalues. On the other hand, $A^T diag(g_j'(y_j^*))A = 2(\tilde{D} - diag\{\frac{\sum_{j\in r}[\epsilon_j + g_j(y_j^*)]}{x_r^*}\})$; hence

$$\begin{aligned}A^T diag(g_j'(y_j^*))A\tilde{D}^{-1} &= 2\cdot diag\ \frac{\sum_{j\in r}[\epsilon_j + g_j(y_j^*)]}{x_r^*}\quad \cdot \\ &\quad \left[diag\ \frac{x_r^*}{\sum_{j\in r}[\epsilon_j + g_j(y_j^*)]}\quad - \tilde{D}^{-1}\right].\end{aligned}$$

Now $diag\{\frac{x_r^*}{\sum_{j\in r}[\epsilon_j + g_j(y_j^*)]}\} - \tilde{D}^{-1} \succeq 0$. This can be shown by contradiction. Suppose the contrary; left multiple the above equation by $\left(diag\{\frac{\sum_{j\in r}[\epsilon_j + g_j(y_j^*)]}{x_r^*}\}\right)^{-1/2}$, and right multiple the above equation by $\left(diag\{\frac{\sum_{j\in r}[\epsilon_j + g_j(y_j^*)]}{x_r^*}\}\right)^{1/2}$; then the left hand side has all nonnegative eigenvalues, while the right hand side is a non-semi-positive definite matrix with at least one eigenvalue being negative, resulting in contradiction.

We claim $E\tilde{D}^{-1}\tilde{G}^{-1}$ has all positive eigenvalues, due to the following three facts:

- $B \succ 0$ since it is a sum of two positive definite matrices.
- $diag\left(f'(\sum_{j\in r} g_j(y_j^*))\frac{[\sum_{j\in r}(\epsilon_j + g_j(y_j^*)]^2}{x_r^*}\right)B$ has all positive eigenvalues, because it is the product of two positive definite matrices;
- $E\tilde{D}^{-1}\tilde{G}^{-1}$ has all positive eigenvalues, because it is similar to $diag\left(f'(\sum_{j\in r} g_j(y_j^*))\frac{[\sum_{j\in r}(\epsilon_j + g_j(y_j^*)]^2}{x_r^*}\right)B$.

Eventually, $\tilde{D}^{-1}\tilde{G}^{-1}E$ has all positive eigenvalues and hence the reduced system in (22) is locally exponentially stable for arbitrary $\beta > 0$.

Hence combining the fact that the boundary system is locally exponentially stable, we conclude the entire system is locally exponentially stable, according to Theorem 11.4 in [25]. ■

C. Proof of theorem 3

First, we show the reduced system is semi-globally exponentially stable, i.e. the system is exponentially stable whenever it is constrained to lie in a large enough compact set. Second, combining the fact that the boundary system is semi-globally exponentially stable, we conclude the whole system is semi-globally exponentially stable, following the same argument and proof in [24], [25].

Define $z_r = \frac{4S^2 n_r^2}{x_r^2 T_r^2}$, and $\rho_r = \frac{4S^2}{T_r^2}$, for $r \in R$. We have

$$\begin{aligned}\dot{Z} &= diag(\frac{2\rho_r}{x_r^2})diag(n_r)\dot{n} - diag(\frac{2\rho_r n_r^2}{x_r^3})\cdot \\ &\quad D^{-1}diag(\frac{2\rho_r}{x_r^2})diag(n_r)\dot{(}n) \\ &= diag(n_r\sqrt{z_r})diag(\frac{2\rho_r}{x_r^2})\Lambda diag(\frac{2\rho_r}{x_r^2})diag(n_r)\dot{n} \qquad (23)\end{aligned}$$

where $\Lambda \triangleq diag(x_r^3/2n_r^2\rho_r) - D^{-1}$ has been shown in Appendix B to be a semi-positive definite matrix.

Define nondecreasing functions $\phi_r(y)$ and $\varphi_r(y), r \in R$ as follows:

$$\phi_r(y) = \int_{\epsilon_r}^{f^{-1}(y)+\epsilon_r} \frac{1}{\sqrt{y}}dy, \quad r \in R \qquad (24)$$

$$\varphi_r(y) = \int_{\epsilon_r}^{f^{-1}(y)+\epsilon_r} \frac{f(y-\epsilon_r)}{\sqrt{y}}dy, \quad r \in R \qquad (25)$$

The following is a La Salle function for the reduced system:

$$\begin{aligned}V(n,z) &= -\sum_{r\in R} c_r\ \int_{\epsilon_r}^{z_r}\frac{1}{\sqrt{y}n_r}dy - \int_{\epsilon_r}^{z_r}\frac{n_r}{\sqrt{y}}f(y-\epsilon_r)dy \\ &\quad + \int_0^{n_r}\frac{1}{y^2}\phi_r(\frac{1}{y^2})dy + \int_0^{n_r}\varphi_r(\frac{1}{y^2})dy\quad .\end{aligned}$$

Its Lee derivative is

$$\begin{aligned}\dot{V}(n,z) &= -\sum_{r\in R} c_r\quad \frac{1}{\sqrt{z_r}n_r} - \frac{n_r^2}{\sqrt{z_r}n_r}f(z_r-\epsilon_r)\quad \dot{z}_r - \\ &\quad \sum_{r\in R} c_r\frac{\dot{n}_r}{n_r^2}\quad \phi_r(\frac{1}{n_r^2}) - \phi_r(f(z_r-\epsilon_r))\quad - \\ &\quad \sum_{r\in R} c_r\dot{n}_r\quad \varphi_r(\frac{1}{n_r^2}) - \varphi_r(f(z_r-\epsilon_r)) \\ &= -\dot{n}diag(n_r)diag(\frac{2\rho_r}{x_r^2})\Lambda\, diag(\frac{2\rho_r}{x_r^2})diag(n_r)\dot{n} - \\ &\quad \sum_{r\in R} c_r\frac{\dot{n}_r}{n_r^2}\quad \phi_r(\frac{1}{n_r^2}) - \phi_r(f(z_r-\epsilon_r))\quad - \\ &\quad \sum_{r\in R} c_r\dot{n}_r\quad \varphi_r(\frac{1}{n_r^2}) - \varphi_r(f(z_r-\epsilon_r)) \qquad (26) \\ &\leq 0.\end{aligned}$$

The equality is taken only when $\dot{n} = 0$, i.e. at the equilibrium. Therefore, the system is globally asymptotically stable. Furthermore, following the proof of Lemma 2 in [29], it can be shown that the reduced system is actually semi globally exponentially stable, as long as $n_r(t)$ is constrained to a large enough compact set defined in Lemma 1 in [29]. It has also been argued in the Lemma that this "large enough" requirement is trivial to meet in practice.

Eventually, following the proof of Theorem 3 in [29] and according to Theorem 11.4 in [25], we conclude that the entire system is semi globally exponentially stable.

REFERENCES

[1] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1988, pp. 314–329.

[2] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Aug. 2000, pp. 43–56.

[3] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness, and stability," *Journal of the Operationl Research Society*, vol. 49, pp. 237–252, 1998.

[4] F. P. Kelly, "Fairness and stability of end-to-end congestion control," *European Journal of Control*, vol. 9, pp. 159–176, 2003.

[5] M. Chen and A. Zakhor, "Rate control for streaming video over wireless," in *Proc. IEEE INFOCOM*, Hongkong, China, Mar. 2004.

[6] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving tcp performance over wireless links," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 756–769, 1997.

[7] H. Balakrishnan and R. Katz, "Explicit loss notification and wireless web performance," in *Proc. of IEEE Globecom Internet Mini-Conference*, Nov. 1998.

[8] N. Samaraweera, "Non-congestion packet loss detection for tcp error recovery using wireless links," *IEE Proceedings of Communications*, vol. 146, no. 4, p. 222C230, Aug. 1999.

[9] S. Cen, P. Cosman, and G. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 703–717, 2003.

[10] G. Yang, M. Gerla, and M. Y. Sanadidi, "Adaptive video streaming in presence of wireless errors," in *Proc. ACM MMNS*, San Diego, USA, Jan. 2004.

[11] F. Yang, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end TCP-Friendly streaming protocol and bit allocation for scalable video over mobile wireless internet," in *Proc. IEEE INFOCOM*, Hongkong, China, Mar. 2004.

[12] L. S. Brakmo and L. L. Peterson, "TCP Vegas: end-to-end congestion avoidance on a global internet," *IEEE J. Select. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.

[13] S. Floyd, "TCP and explicit congestion notification," *ACM Computer Communication Review*, vol. 24, pp. 10–23, Oct. 1994.

[14] M. Chen and A. Zakhor, "AIO-TFRC: A light-weighted rate control scheme for streaming over wireless," in *Proc. of IEEE WirelessCom Symposium on Multimedia over Wireless 2005*, June 2005.

[15] S. H. Low and D. E. Lapsley, "Optimization flow control, i: Basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 861–875, Dec. 1999.

[16] F. Paganini, Z. Wang, J. Doyle, and S. Low, "A new TCP/AQM for stable operation in fast networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 2003.

[17] S. Kunniyur and R. Srikant, "A time scale decomposition approach to adaptive ecn marking," *IEEE Trans. Automat. Contr.*, vol. 47, no. 6, pp. 882–894, June 2002.

[18] G. Vinnicombe, "On the stability of end-to-end congestion control for the internet," University of Cambridge, Cambridge, UK, Tech. Rep. CUED/F-INFENG/TR.398, 2001.

[19] F. Paganini, J. Doyle, and S. Low, "Scalable laws for stable network congestion control," in *Proc. IEEE CDC*, Dec. 2001.

[20] M. Chen, A. Abate, and S. Sastry, "New congestion control schemes over wireless networks: Stability analysis," in *Proceedings of the* 16^{th} *IFAC World Congress*, Prague, 2005.

[21] A. Abate, M. Chen, and S. Sastry, "New congestion control schemes over wireless networks: Delay sensitivity analysis and simulations," in *Proceedings of the* 16^{th} *IFAC World Congress*, Prague, 2005.

[22] J. Mahdavi and S. Floyd. (1997, Jan.) TCP-Friendly unicast rate-based flow control. Technical note sent to end2end-interest mailing list. [Online]. Available: http://www.psc.edu/networking/papers/tcp_friendly.html

[23] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Networking*, vol. 8, no. 5, pp. 556 – 567, Oct. 2001.

[24] S. Sastry, *Nonlinear Systems, Analysis, Stability and Control*. New York, NY: Springer Verlag, 1999.

[25] H. Khalil, *Nonlinear Systems (3rd edition)*. Prentice Hall, 2001.

[26] Traceroute. [Online]. Available: http://www.traceroute.org/

[27] M. Chen and A. Zakhor. (2005, June) Enhanced MULTFRC (E-MULTFRC). Technical report of EECS Department, University of California at Berkeley. [Online]. Available: http://www-video.eecs.berkeley.edu/∼minghua/papers/emultfrc.pdf

[28] Network simulation version 2. [Online]. Available: http://www.isi.edu/nsnam/ns/

[29] M. Chen and A. Zakhor, "Application layer based flow control in wireless networks: Semi-global exponential stability of the equilibrium," in *Proceeding of 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006, submitted. [Online]. Available: http://www-video.eecs.berkeley.edu/∼minghua/papers/cdc.06.pdf

PLACE PHOTO HERE

Minghua Chen received a B.Eng. and M.S. degrees in Electronic Engineering from Tsinghua University, and is expecting a Ph.D. degree in Electrical Engineering and Computer Sciences from University of California at Berkeley, in 1999, 2001 and 2006 respectively. He received a Pao Family fellowship in 2001, a Management of Technology in China Fellowship in 2004, all from U.C. Berkeley. He is also a receipt of Outstanding Oversea Student Scholarship from Education Department of China, in 2005. His work on rate control for video over wireless won a runner-up for the Best Paper Award of IEEE WirelessCom 2005 (top 3 out of 250+ accepted papers). He is co-author of the book IPv6 Principle and Practice (People's Posts and Telecommunication Publishing House, 2000). His research interests are in general area of information transmission over wireless networks, including source coding, flow control, network routing and coding, and wireless communications. His current work focuses on flow control in wireless network, routing in wireless sensor networks, digital signal processing, and wireless communications.

PLACE PHOTO HERE

Avideh Zakhor received a B. S. degree from California Institute of Technology, Pasadena, and S. M. and Ph. D. degrees from Massachusetts Institute of Technology, Cambridge, all in electrical engineering, in 1983, 1985, and 1987 respectively. In 1988, she joined the Faculty at U. C. Berkeley where she is currently Professor in the Department of Electrical Engineering and Computer Sciences. Her research interests are in the general area of image and video processing, multimedia communication, and 3D modeling. Together with her students, She has won a number of best paper awards, including the IEEE Signal Processing Society in 1997, IEEE Circuits and Systems Society in 1997 and 1999, international conference on image processing in 1999, and Packet Video Workshop in 2002. She holds 5 U.S. patents, and is the co-author of the book, "Oversampled A/D Converters" with Soren Hein.

Prof. Zakhor was a General Motors scholar from 1982 to 1983, was a Hertz fellow from 1984 to 1988, received the Presidential Young Investigators (PYI) award, and Office of Naval Research (ONR) young investigator award in 1992. From 1998 to 2001, she was an elected member of IEEE Signal Processing Borad of Governers. In 2001, she was elected as IEEE fellow. She received the Okawa Prize in 2004.

She co-founded OPC technology in 1996, which was later by Mentor Graphics (Nasdaq: MENT) in 1998, Truvideo in 2000, and Urban Scan in 2005.