Flow Control over Wireless

Minghua Chen and Avideh Zakhor
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley, CA 94720
{minghua, avz}@eecs.berkeley.edu

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 network. Two widely accepted flow control methods in wired network is TCP [1] (for data) and TFRC [2] (for multimedia). Kelly [22] [23] has laid down solid theoretical framework for these flow control methods over wired network, under which the optimality, fairness, and stability issues can be investigated and evaluated. However, these two methods both assume that packet loss in wired network is primarily due to congestion, and as such not applicable to wireless networks in which the bulk of packet loss is due to error at the physical layer.

In this project we extend Kelly's flow control framework to the wireless scenario. In this framework, the problem of flow control over wireless can be formulated as a concave optimization problem with inaccurate feedback. All current existing approaches belong to one class of solution for the problem, which requires modifications to existing protocols, e.g. TCP, or even infrastructure, e.g. router, making them hard to deploy.

We then propose a new class of solution to the problem, which are end-to-end based approaches achieving optimal performance by only modifying the application layer. The global stability, delay sensitivity, rate of convergence, and robustness to stochastic perturbation of the schemes are investigated. We then apply our results to design a practical rate control scheme for streaming video over wireless network, whose performance is characterized and evaluated using NS-2 simulations and actual experiments over Verizon Wireless 1xRTT data network. Analysis, simulation, and actual experiment results also show the scheme in fact works in both wired and wireless scenario. Similar scheme can also be designed for data transmission over wireless network, which could be part of the future work.

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; thus the congestion assumption breaks down, resulting in TCP seriously underutilizing the wireless bandwidth. Similar observations hold for TCP-friendly schemes, e.g. TCP-Friendly Rate Control (TFRC) [2], [3], [21], as they share the same key assumption as TCP. The need to solve the problem is becoming urgent today as wireless data and streaming services are increasingly popular.

Consequently, there have been a number of efforts to improve the performance of TCP or TFRC over wireless [4]–[20]. All these methods either hide end-hosts from packet loss caused by wireless channel error, or provide end-hosts the ability to distinguish between packet loss caused by congestion, and that caused by wireless channel error. For example, Snoop is a TCP-AWARE link layer approach which suppresses acknowledgement packets (ACK) from the TCP receiver, and does local retransmissions when a packet is corrupted by wireless channel errors [4]. End-to-end statistics can be used to detect congestion when a packet is lost [13]–[20]. For example, by examining trends in the one-way delay variation, one could interpret loss as a sign of congestion if one-way delay is increasing, and a sign of wireless channel error otherwise. The disadvantage of these schemes is that they need modifications to network infrastructure or protocols, making them hard to deploy.

Our recent work, MULTFRC [21], opening appropriate number of connections to fully utilize the wireless bandwidth when needed, indicates a new class of approach to improve TFRC performance in wireless networks by modifying only the application layer, and as such is also applicable to TCP.

This is more like an in-progress report than a proposal, as not only the motivations but also up-to-date results are presented. For more details, please refer to our publication in preparation.

While the existing approaches provide practical insight on how to improve TCP/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 ways to achieve optimal or good performance. Hence a general framework for flow control over wireless is needed to address stability issue, and provide guidelines prior to any implementation. Specifically, answers to the following questions are expected:

- How to define the problem of flow control over wireless network? What is the framework for the analysis?
- Under the framework, are there any *optimal* and *robust* application layer schemes that solve the problem?

Recently, there has been a great deal of research activities on decentralized end-to-end flow control algorithms on wired network. A widely recognized setting, introduced by Kelly et. al. [22] and refined for TCP in [23], views flow control schemes as algorithms to compute the optimal solution to a utility maximization problem.[1]

In one kind of the algorithms Kelly proposed, namely primal algorithms, the 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 showed TCP is in fact a primal like algorithm with packet loss rate as the associated price function [23]. The stability for the system with and without delay, with and without disturbance, are reviewed and developed in [23].

In this project, we extend Kelly's framework in [23][2] to the wireless network scenario, where a wireless link is associated with a fixed bandwidth and a fixed packet loss rate caused by the physical channel errors. Specifically, we analyze the performance of the primal algorithm in the wireless case with packet loss rate as the price. Hence in the presence of packet loss caused by physical channel error, the problem of flow control over wireless can be formulated as a concave optimization problem with inaccurate feedback. A new class of optimal approach is then proposed. The global stability, delay sensitivity, convergence rate, and robustness to stochastic perturbation are investigated. We then apply our results to design a practical rate control scheme for streaming video over wireless network, whose performance is characterized and evaluated using NS-2 simulations and actual experiments over Verizon Wireless 1xRTT data network. Analysis, simulation, and actual experiment results also show the scheme in fact works in both wired and wireless scenario.

This proposal is structured as follows. Section II includes problem formulation. Then one new class of approach addressing the problem is proposed in Section III, together with analysis for global stability etc. issues. Section IV shows a practical scheme with results from NS-2 simulations and actual experiments over 1xRTT wireless data networks. Section V concludes the proposal with discussions and future work.

II. Problem formulation

A. Overview of flow control framework and TCP modeling

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 to be the set of routes, where a *route* r is a non-empty subset of J associating 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.

Assume utilities are additive, so that the aggregate utility of the entire system is $\sum_{r\in R} U_r(x_r)$. Define the cost incurred at link j as $P_j(\cdot)$. The flow control problem under a deterministic fluid model, first introduced by Kelly et. al. [22] and later refined in [23], is a concave optimization problem maximizing the net utility:

$$\max \sum_{r\in R} U_r(x_r) - \sum_{j\in J} \int_0^{\sum_{s:j\in s} x_s} p_j(y)\, dy, \tag{1}$$

[1]There are many other similar works done by Low, Paganini, Basar, Srikant etc., we won't review them here. For a complete review, please refer to our publication in preparation.

[2]We also show the similar extension for the framework in [22], in our recent Infocom submission.

where $p_j(y)$ is called the price function and is required to be non-negative, continuous, increasing and not identically zero. With these assumptions on $p_j(y)$, the objective function in (1) is strictly concave. One common price function used in practice is the packet loss rate, which is zero for no congestion, and concavely increases otherwise:

$$p_j(y) = \frac{(y - C_j)^+}{y}, \tag{2}$$

where C_j is the capacity of link j, and y denotes the aggregate rate passing through link j. In this proposal, we assume $p_j(\sum_{s:j\in s} x_s)$ is small such that the end-to-end packet loss rate for user r, i.e. $1 - \prod_{j\in r} p_j(\sum_{s:j\in s} x_s)$, is approximately $\sum_{j\in r} p_j(\sum_{s:j\in s} x_s)$.

Kelly [23] shows TCP is a primal-like algorithm, as follows:

$$\frac{d}{dt}x_r(t) = \frac{1}{2}\left(\frac{2S^2}{T_r^2} - x_r^2(t)\sum_{j\in r} p_j(\sum_{s:j\in s} x_s(t))\right), \quad r \in R, \tag{3}$$

solving the optimization problem in (1) with $U_r(x_r) = -2S^2/x_r$ where S is the TCP packet size, and $p_j(y)$ takes the form in (2). This can also be viewed as the continuous version of steepest gradient descent algorithm solving the optimization problem in (1).

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

Kelly [23] showed the system (3) has a unique equilibrium, to which all trajectories converge, as follows:

$$x_r^o = \frac{\sqrt{2}S}{T_r\sqrt{\sum_{j\in r} p_j\left(\sum_{s:j\in s} x_s^o\right)}}, \quad r \in R; \tag{4}$$

which should remind us the well known TCP steady state throughput equation [3]. This equilibrium is also the finite solution for the optimization problem in (1), associated with some good properties: firstly, all routes are fully utilized yet no congestion collapse, i.e. $\forall r \in R, \exists j \in r$, s.t. $C_j \le \sum_{s:j\in s} x_s^o < \infty$; secondly, there is roughly α-fairness among users [24] with $\alpha = 2$.

B. Problem of flow control over wireless

For the wireless network, we assume the links are associated with not only a fixed capacity but also a packet loss rate caused by the physical channel errors. Nevertheless, flow control algorithms still aim to address the same optimization problem shown in (1), but with inaccurate prices fed back from network.

Let $\epsilon_j \ge 0, j \in J$ be the packet loss rate due to physical channel errors for link j, then the inaccurate price function, denoted by $q_j(\cdot)$, is a the sum of ϵ_j and $p_j(\cdot)$, under the assumption that ϵ_j is small:

$$q_j(y) = p_j(y) + \epsilon_j \ge \epsilon_j, \quad j \in J, \tag{5}$$

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

$$\frac{d}{dt}x_r(t) = \frac{1}{2}\left(\frac{2S^2}{T_r^2} - x_r^2(t)\sum_{j\in r} q_j(\sum_{s:j\in s} x_s(t))\right), \quad r \in R. \tag{6}$$

Following a similar analysis in [23], one can show the system (6)-(5) has a new unique equilibrium, to which all trajectories converge, as follows:

$$x_r^* = \frac{\sqrt{2}S}{T_r\sqrt{\sum_{j\in r} q_j\left(\sum_{s:j\in s} x_s^*\right)}} = \frac{\sqrt{2}S}{T_r\sqrt{\sum_{j\in r}\left[p_j\left(\sum_{s:j\in s} x_s^*\right) + \epsilon_j\right]}} \le \frac{\sqrt{2}S}{T_r\sqrt{\sum_{j\in r}\epsilon_j}}, \quad r \in R. \tag{7}$$

Although there is roughly α-fairness among users with $\alpha = 2$, x^* is a suboptimal solution as it is different from the unique optimal one x^o. Furthermore, any route r could suffer underutilization if $\sum_{j\in r}\epsilon_j$ is sufficiently large. For instance, in the one user one bottleneck network, the underutilization happens if and only if $\sqrt{2}S/(T_r\sqrt{\epsilon}) < C$, which is also observed in [21]. Hence underutilization is the main problem in TCP over wireless; in fact, similar analysis shows that it is also the main problem in any flow control methods that use packet loss rate as price function, e.g. TFRC.

One possible solution is to provide user r the accurate price $\sum_{j\in r} p_j(\cdot)$ and apply it in the control law of $x_r(t)$, this could be done by either end-to-end estimation with or without cross layer information, or hide the wireless loss from users via local retransmission. All the existing approaches, except MULTFRC, belong to this class of solution, by either modifying the protocols or even the infrastructures, making them hard to deploy.

III. Proposed solution

A. A new class of approach

Motivated by our previous approach MULTFRC [21], we propose a new class of approach to flow control based on gradually adjusting the number of connections for user r, denoted as $n_r(t)$, hence it is an end-to-end application layer based schemes and require modification to neither the network infrastructure nor protocols.

In our approach, both $x_r(t)$ and $n_r(t)$ is dynamically adjusted as follows: $\forall r \in R$,

$$\frac{d}{dt}x_r(t) = \frac{1}{2}\left(\frac{2S^2 n_r^2(t)}{T_r^2} - x_r^2(t)\sum_{j\in r} q_j\Big(\sum_{s:j\in s} x_s(t)\Big)\right), \qquad r \in R \tag{8}$$

$$\frac{d}{dt}n_r(t) = c\left(\frac{1}{n_r(t)} - n_r(t)\frac{\sum_{j\in r} p_j(\sum_{s:j\in s} x_s(t))}{\sum_{j\in r} q_j(\sum_{s:j\in s} x_s(t))}\right), \qquad r \in R \tag{9}$$

where c is a constant.

It is easy to check the above system has x^o as an unique equilibrium for x that solve the optimization problem in (1). The intuition behind the approach is that when loss rate caused by channel error is large hence individual connection sending rate drops, users open more connections to bring up the aggregate rate. Further analysis shows that the proposed approach is a gradient descent method, not a steepest one however, solving the optimization problem in (1), under the following two time scales assumption that reflects the physical reality: *The number of connection, $n_r(t)$, changes in a timescale much larger than the source rate, $x_r(t)$.*

Under the above assumption, system (8)-(9) also fits into the classical small-perturbation framework [25] [26], and therefore can be decoupled into a fast-scale system described by (8) with $w_r(t), r \in R$ being constant, namely boundary system, and a slow-scale system by (9) along the manifold defined by the stationary solution of (8), namely reduced system. Several results on stability etc. can be obtained for this two timescale nonlinear system.

B. Stability, delay sensitivity and convergence rate

Here our main results on the global stability, delay sensitivity, convergence rate, and robustness to stochastic perturbation are stated[3], as follows:

Theorem 1: Under the two time scales assumption, the system (8)-(9) has x^o in (4) as an unique equilibrium solving the optimization problem in (1), to which all trajectories converge.

Intuitively, $x_r(t)$ first converges in a fast time scale to the stationary solution of (8), defining a manifold $n_r^2(t) = x_r^2(t)T_r^2\sum_{j\in r} q_j(\sum_{s:j\in s} x_s(t))/(2S^2), r \in R$; then along this manifold, $n_r(t)$ and $x_r(t)$ converge in a slow time scale to the global optimal equilibrium. Consequently, combination of control law (9)on $n_r(t)$ and any flow control method that has the same asymptotical stationary $x_r(t)$ as TCP will retain the global asymptotical convergence shown in Theorem 1.

[3] For the consideration on the pages length, we skip the proofs here, which can be provided upon request.

Theorem 2: Under the two time scales assumption, the delayed version of system (8)-(9), is

$$\begin{aligned} \frac{d}{dt}x_r(t) &= \frac{1}{2}\left(\frac{2S^2 n_r^2(t-T_r)}{T_r^2} - x_r^2(t-T_r)\sum_{j\in r} q_j(\sum_{s:j\in s} x_s(t-d_1(s,j)-d_2(j,r)))\right), \\ \frac{d}{dt}n_r(t) &= c\left(\frac{1}{n_r(t-T_r)} - n_r(t-T_r)\frac{\sum_{j\in r} p_j(\sum_{s:j\in s} x_s(t-d_1(s,j)-d_2(j,r)))}{\sum_{j\in r} q_j(\sum_{s:j\in s} x_s(t-d_1(s,j)-d_2(j,r)))}\right), \end{aligned}$$

where $d_1(r,j)+d_2(j,r)=T_r, \forall r\in R$, $d_1(r,j)$ is the forward delay from sender of route r to link j, and $d_2(j,r)$ is the return delay from the link j to the sender of route r. The above delay system is locally stable around the equilibrium x^o if $\forall r\in R$,

$$\begin{aligned} \frac{x_r^o}{2}\left(2\sum_{j\in r} q_j + \sum_{j\in r} q_j' \sum_{s:j\in s} x_s^o\right) &< \frac{\pi}{2T_r}, \\ \frac{c}{\min_{r\in R}\sum_{j\in r} q_j}\left(2\sum_{j\in r} p_j + \sum_{j\in r} p_j' \sum_{s:j\in s} x_s^o\right) &< \frac{\pi}{2T_r}, \end{aligned}$$

Theorem 3: Under the two time scales assumption, the system (8)-(9) has a local convergence rate determined by the smallest eigenvalue, necessarily positive, of matrix $diag\{T_r^2, r\in R\}DB^{-1}diag\{T_r^{-2}\}$ where

$$\begin{aligned} B &= diag\{2x_r^o\sum_{j\in r} q_j, r\in R\} + diag\{x_r^o, r\in R\}A^T diag\{q_j', j\in J\}A\cdot diag\{x_r^o, r\in R\}, \\ D &= diag\{2x_r^o\sum_{j\in r} p_j, r\in R\} + diag\{x_r^o, r\in R\}A^T diag\{p_j', j\in J\}A\cdot diag\{x_r^o, r\in R\}. \end{aligned}$$

Theorem 4: Under the two time scales assumption, the system (8)-(9) with a Gaussian perturbation on price, has a solution centered on the equilibrium x^o with a multivariate normal distribution $N(x^o,\Sigma)$, whose covariance matrix Σ is determined explicitly in terms of the parameters of the network. [4]

In summary, for system (8)-(9), not only its equilibrium x^o solves the optimization problem in (1), but also several good properties are associated with the equilibrium: first, all routes are fully utilized yet no congestion collapse; second, there is roughly α-fairness among users with $\alpha=2$; third, the system is locally robust in the presence of heterogenous delays of users, under some mild conditions; fourth, the system is locally robust in the presence of stochastic perturbation.

Therefore, it is possible to achieve the good performance in wireless network, as good as if there is no wireless link error, by adjusting the number of connections at application layer. Furthermore, notice all the analysis and results hold regardless of the values of $\epsilon_j, j\in J$, then it is possible to design only one practical scheme, following (8)-(9), for both wireless network and wired network, which corresponds to $\epsilon_j=0, j\in J$.

C. Relaxing the ratios $\sum_{j\in r} p_j(\cdot)/\sum_{j\in r} q_j(\cdot)$

The system (8)-(9), if implemented, depends on the measurement of the ratios $\sum_{j\in r} p_j(\cdot)/\sum_{j\in r} q_j(\cdot), r\in R$, which could be hard to be accurate in reality. Here we show if we can relax our goals to full utilization and no congestion collapse regardless of the network scale, we can relaxing the ratio to be only one bit information from the congestion measurement.[5]

We first understand intuitively how the ratios affect the system performance, from (9):

- when route r is underutilized, the ratio is zero, resulting in an increase in the number of connections $n_r(t)$ and hence an increase in source rate $x_r(t)$, which make the system pursue full utilization on any route r;
- after route r is fully utilized, the ratio takes values between zero and one, finely adjusting $n_r(t)$ and hence $x_r(t)$ to make the system pursue the maximum net utility and $\alpha=2$ fairness among users;

[4] Details are included in publication in preparation.

[5] In fact, we find we expect the new system to be a projected gradient descent method solving a constrained concave optimization problem with an unique optima, and some kind of fairness can be guaranteed. The investigation is carrying out currently.

- in case route r is heavily congested, the ratio is close to one, strongly pushing down $n_r(t)$[6], hence preventing $x_r(t)$ go unbounded. This way, the system avoids congestion collapse.

One should note the above observations are true even when some routes in the network are congested and some not, since any underutlized route r is affected by $\epsilon_j, j \in r$ only, which does not depend on other users' rate, and hence is essentially decoupled from the congested routes.

Therefore if we only want to achieve full utilization and no congestion collapse on all the routes regardless of the network scale, we can replace the ratio by a congestion indicator function for each user, denoted as $I_r(t)$, taking value one if route r is congested, and zero otherwise.[7] The number of connections, $n_r(t)$ now is adjusted as bellows:

$$\frac{d}{dt}n_r(t) = c\left(\frac{1}{n_r(t)} - n_r(t)I_r(t)\right), \tag{10}$$

and the entire system now become (8)-(10). In reality, this one bit information can be accurately gathered by measuring the increases on one-way delays or round trip time, as we explore in designing the practical scheme in next section.

IV. ENHANCED MULTFRC (E-MULTFRC)

Although the analysis in previous Section is based on TCP model, they can be extended to TFRC, since it is both practically and theoretically shown TFRC has the same stationary behavior as TCP [2] (and our publication in preparation). As we are more interesting in doing the rate control for streaming video over wireless, we design a practical scheme for that based on our analysis and proposed solution by adjusting the number of TFRC connections for the streaming application. The design goals are first to achieve full utilization on wireless link; second avoid congestion collapse in any size of network; third, works efficiently on both wireless and wired network.

The framework of E-MULTFRC is shown in Figure 1. As seen, there are two components in the system: rtt measurement sub-system (RMS), and connections controller sub-system (CCS).

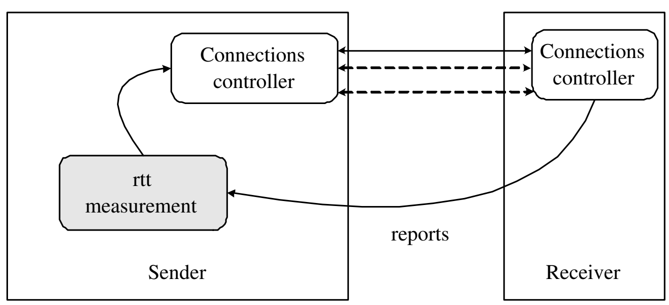

Fig. 1. E-MULTFRC system framework.

A. rtt Measurement Sub-system (RMS)

The gray blocks in Fig. 1 represents RMS that resides at the sender; it receives reports from receiver every round trip time, containing the an average rtt_{sample} measured in the past round trip time window. After waiting for a fixed interval, denoted as τ, RMS then further computes a running average of these rtt_{sample}s, denoted by ave_rtt, and reports it to the CCS. Here, τ defines the frequency of adjusting number of connections, and it has to be large enough to ensure the frequency is much slow than the one of source rate, which is typically in the order of round trip time. In current version, we choose τ to be 20 seconds.

[6]Unless $n_r(t)$ has take the minimum value 1.

[7]The validity of this claim are included in publication in preparation.

B. Connection Controller Sub-system (CCS)

The CCS is shown as the white blocks in Figure 1. Its basic functionality is to Inversely Increase and Multiplicative Decrease (IIMD(α, β)) the number of connections n, based on the input from RMS. Specifically, it first sets the rtt_min as the minimum ave_rtt seen so far, and then adapts the number of connection n as follows:

$$n = \begin{cases} \beta n, & \text{if } ave\_rtt - rtt\_min > \gamma rtt\_min; \\ n + \alpha/n, & \text{otherwise.} \end{cases} \quad (11)$$

where $\alpha = 1 - \beta < 1$ is suggested by differential equation (9); γ is a preset parameter; and $I[ave\_rtt - rtt\_min > \gamma rtt\_min]$ is the congestion indicator function. We then empirically chooses $\alpha = 0.25$, $\beta = 0.75$, and $\gamma = 0.25$.

For a given route, the rtt_min is a constant representing the minimum round trip time for that route, i.e. 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-MULTFRC can tolerate before it starts to decrease the number of connections. As a result, under ideal conditions, E-MULTFRC keeps increasing the number of connections to make ave_rtt as close as possible to $(1 + \gamma)rtt\_min$ without exceeding it.

C. NS-2 simulations and 1xRTT wireless experiments

To show E-MULTFRC performance, we carry out both NS-2 [27] simulations and actual experiments over Verizon Wireless 1xRTT CDMA data network. The topology for NS-2 simulations is shown in Figure 2 with the following settings: $C_1 = 1\ Mbps, C_2 = 5\ Mbps$, $S = 760$ bytes, user 1, 2, 3 has a round trip time 691 ms, 651 ms, and 69 ms respectively. We try two sets of (p_w^1, p_w^2). The first set is $(0, 0)$, representing the wired scenario; the second set is $(0.02, 0.01)$, representing the wireless scenario. The wireless link is modeled as a wired link with an exponential random packet loss model.

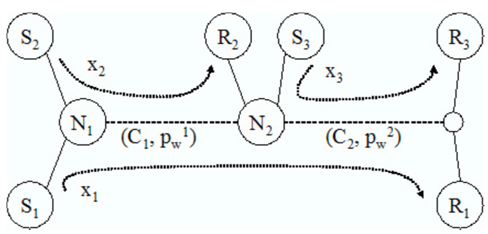

Fig. 2. Simulation topology.

In simulations, we stream 9000 seconds video from s_i to r_i for user i, $i = 1, 2, 3$, using E-MULTFRC scheme, or only one TFRC. The simulation results are shown in Fig. 3, including throughput measured every 10 seconds, number of connections, end-to-end packet loss rate measured every 30 seconds and average rtt measured every 20 seconds. Several observations can be drawn from the results. First, as predicted in the analysis in Section III, E-MULTFRC achieves the same good performance as one TFRC in wired network[8], as E-MULTFRC opens only one connection anyway seen in Fig. 3.c. Furthermore, we have checked that the optimality and fairness for the stationary sending rates stated in Section II-A are achieved. Second, as seen from Fig. 3.b, E-MULTFRC can achieve almost the same stationary sending rates in wireless network, as compared to in wired network. Moreover, the end-to-end packet loss rates for each users are kept around the minimum values, and so does the round trip time, as a result of E-MULTFRC pursuing the boundary between full utilization and underutilization. Fourth, E-MULTFRC superiorly outperforman one TFRC scheme, at the expense of minor modification to the application layer. This confirms our analysis in Section III, and shows E-MULTFRC can achieve good performance in wireless scenario.

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 department at U.C. Berkeley to a laptop connected via 1xRTT CDMA modem using E-MULTFRC and

[8]Assuming TFRC has the same asymptotical behavior as TCP.

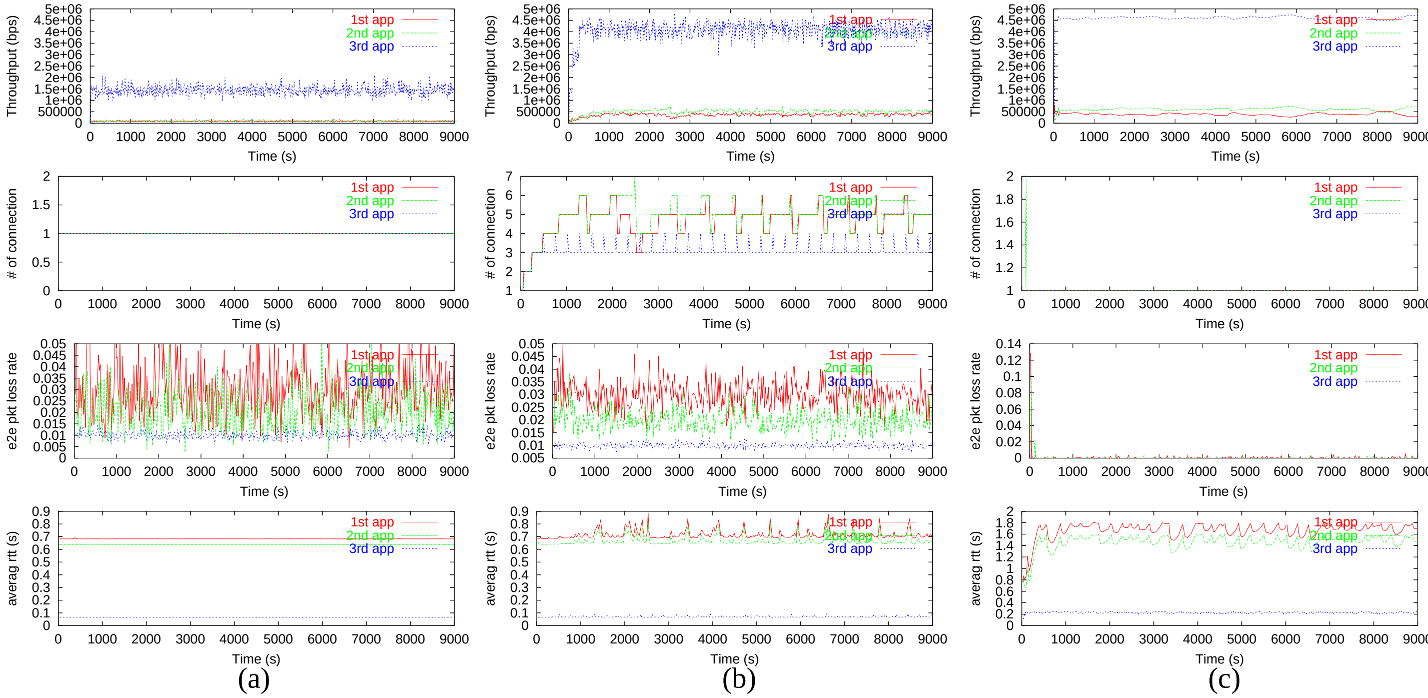

Fig. 3. Throughput, number of connections, end-to-end packet loss rate and average rtt for (a) one TFRC with $p_w^1 = 0.02, p_w^2 = 0.01$, (b) E-MULTFRC with $p_w^1 = 0.02, p_w^2 = 0.01$, and (c) E-MULTFRC with $p_w^1 = 0.00, p_w^2 = 0.00$.

TFRC. We compare the performances in Table I with packet size of 760 bytes. As seen, E-MULTFRC on average opens up 1.9 connections, and results in 60% higher throughput at the expense of a larger round trip time, and higher packet loss rate. This experiment demonstrates the efficiency of E-MULTFRC in real world, at least in some part of it.

TABLE I

ACTUAL EXPERIMENTAL RESULTS FOR A E-MULTFRC SYSTEM OVER 1XRTT CDMA.

scheme	throughput (kbps)	rtt (ms)	packet loss rate	ave. # of conn.
one TFRC	54	1624	0.031	N/A
E-MULTFRC	89	2767	0.041	1.9

V. DISCUSSION AND FUTURE WORK

In this project, we have extended Kelly's framework for flow control into wireless scenario. The fundamental problem of wireless flow control is the prices flow control methods rely on becomes inaccurate. We then proposed a new class of end-to-end based solution to achieve the optimal performance, by open appropriate number of connections on application layer. Hence the solution requires no modification to either existing protocols, e.g. TCP, or infrastructure, e.g. router. A practical scheme called E-MULTFRC is then proposed and its efficient performance is characterized and evaluated using both NS-2 simulations and 1xRTT wireless experiments.

There are some on-going works on this project. Most importantly, we are trying to fully characterize the performance of system (8)-(10) based on the congestion indicator function. As mentioned in Section III-C and shown in Section IV-C, we have shown the system guarantees full utilizations on all routes and prevents congestion collapse. We expected more can be said about the convergence and fairness issues.

Future work includes designing a practical scheme, following the similar idea in E-MULTFRC, to improve data transmission over wireless based on TCP, and evaluate its performance.

REFERENCES

[1] Jacobson V, ”Congestion avoidance and control”, *Proceedings of ACM SIGCOMM*, 1998:314-329.

[2] Sally Floyd, Mark Handley, Jitendra Padhye and Joerg Widmer, ”Equation-Based Congestion Control for Unicast Applications”, *Proc. ACM SIGCOMM 2000*, Aug. 2000, pp. 43 - 56

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

[4] H. Balakrishnan, V. Padmanabhan, S. Seshan, R. Katz, ”A comparison of mechanisms for improving TCP performance over wireless links”, *ACM SIGCOMM'96*, pp.256 - 269

[5] Wenqing Ding, Jamalipour A, ”A new explicit loss notification with acknowledgment for wireless TCP”, *12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Part vol.1, 2001, pp.B-65-9 vol.1. Piscataway, NJ, USA.

[6] Ratnam K, Matta I, ” WTCP: an efficient mechanism for improving wireless access to TCP services”, *International Journal of Communication Systems*, vol.16, no.1, Feb. 2003, pp.47-62. Publisher: Wiley, UK.

[7] Rendon J, Casadevall F, Carrasco J, ”Wireless TCP proposals with proxy servers in the GPRS network”, *13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Part vol.3, 2002, pp.1156-60 vol.3. Piscataway, NJ, USA.

[8] H. Balakrishnan and R. Katz, ”Explicit Loss Notification and Wireless Web Performance”, *IEEE Globecom Internet Mini-Conference*, Nov. 1998

[9] Yaling Yang, Honghai Zhang, Kravets R, ”Channel quality based adaptation of TCP with loss discrimination”, *IEEE Global Telecommunications Conference*, Part vol.2, 2002, pp.2026-30 vol.2. Piscataway, NJ, USA.

[10] Chiasserini CF, Meo M, ”Improving TCP over wireless through adaptive link layer setting”, *IEEE Global Telecommunications Conference*, Part vol.3, 2001, pp.1766-70 vol.3. Piscataway, NJ, USA.

[11] Cobb JA, Agrawal P, ”Congestion or corruption? A strategy for efficient wireless TCP sessions”, *Proceedings IEEE Symposium on Computers and Communications*, 1995, pp.262-8. Los Alamitos, CA, USA.

[12] Jin-Hee Choi, See-Hwan Yoo, Chuck Yoo, ”A flow control scheme based on buffer state for wireless TCP”, *4th International Workshop on Mobile and Wireless Communications Network* , 2002, pp.592-6. Piscataway, NJ, USA.

[13] S. Biaz, N. H. Vaidya, ”Distinguishing congestion losses from wireless transmission losses: a negative result”, *Computer Communications and Networks*, 1998

[14] S. Biaz, N. H. Vaidya, ”Discriminating congestion loss from wireless losses using inter-arrival times at the receiver”, *Proc. of IEEE Symposium on Application-specific System and Software Engr. and Techn.*, pp. 10-17, Richardson, TX, Mar 1999

[15] N. Samaraweera, ”Non-congestion packet loss detection for TCP error recovery using wireless links”, in *IEE Proceedings of Communications*, 146(4), pp. 222C230, Aug 1999.

[16] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda, ”Achieving moderate fairness for UDP flows by pathstatus classification,” in *Proc. 25th Annual IEEE Conf. on Local Computer Networks (LCN 2000)*, pp. 252C61, Tampa, FL, Nov 2000.

[17] S. Cen, P.C. Cosman, and G.M. Voelker, ” End-to-end differentiation of congestion and wireless losses”, *Proc. Multimedia Computing and Networking (MMCN) conf. 2002*, pp. 1-15, San Jose, CA, Jan 23-25, 2002

[18] Sinha P, Venkitaraman N, Nandagopal T, Sivakumar R, Bharghavan V, ”A wireless transmission control protocol for CDPD”, *IEEE Wireless Communications and Networking Conference*, Part vol.2, 1999, pp.953-7 vol.2. Piscataway, NJ, USA.

[19] Sinha P, Nandagopal T, Venkitaraman N, Sivakumar R, Bharghavan V, ”WTCP: a reliable transport protocol for wireless wide-area networks”, *Wireless Networks*, vol.8, no.2-3, 2002, pp.301-16. Netherlands.

[20] Jae-Joon Lee, Fang Liu, Kuo C-CJ, ”End-to-end wireless TCP with non-congestion packet loss detection and handling”, *Proceedings of Spie - the International Society for Optical Engineering*, vol.5100, 2003, pp.104-13. USA.

[21] M. Chen and A. Zakhor, ”Rate Control for Streaming Video over Wireless”, *Proceeding of Infocom 2004*, Hongkong, China, March, 2004.

[22] F. P. Kelly, A. Maulloo, 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

[23] F. Kelly, ”Fairness and stability of end-to-end congestion control”, *European Journal of Control*, 9 pp. 159-176, 2003

[24] J. Mo, J. Walrand, ”Fair end-to-end window-based congestion control”, *IEEE/ACM Trans. Networking 8, 5, Pages 556 - 567*, Oct. 2000

[25] Shankar Sastry, ”Non Linear Systems, Analysis, Stability and Control”, *Springer Verlag*, 1999

[26] Hassan Khalil, ”Nonlinear Systems (3rd edition)”, *Prentice Hall*, 2001

[27] Network Simulation version 2, *http://www.isi.edu/nsnam/ns/*