

Stability and Delay Consideration for Flow Control over Wireless Networks

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

Abstract—In this paper we develop a general framework for the problem of flow control over wireless networks, evaluate the existing approaches within that framework, and propose new ones. Significant progress has been made on the mathematical modeling of flow control for the wired Internet, among which Kelly’s contribution is widely accepted as a standard framework. We extend Kelly’s flow control framework to the wireless scenario, where the wireless link is assumed to have a fixed link capacity and a packet loss rate caused by the physical channel errors. In this framework, the problem of flow control over wireless can be formulated as a convex optimization problem with noisy feedback. We then propose two new solutions to the problem achieving optimal performance by only modifying the application layer. The global stability and the delay sensitivity of the schemes are investigated, and verified by numerical results. Our work advocates the use of multiple connections for flow, or congestion control, over wireless.

Index Terms—Wireless, flow control, congestion control, primal algorithm, global stability, delay stability

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 underutilizing the wireless bandwidth. Similar observations hold for TCP-friendly schemes, e.g. TCP-Friendly Rate Control (TFRC) [2, 6, 25], as they share the same key assumption as TCP.

Consequently, there have been a number of efforts to improve the performance of TCP or TFRC over wire-

less [8–24]. 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 [8]. End-to-end statistics can be used to detect congestion when a packet is lost [17–24]. 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 on MULTFRC [25], modifying only the application layer, provides a new approach to improve the performance of TFRC in wireless networks, and as such is also applicable to TCP. MULTFRC opens appropriate number of connections given the observed round trip time, so as to fully utilize the wireless bandwidth, while keeping the packet loss rate and round trip time at a minimum. Theoretical analysis on the optimality, NS-2 simulations, and actual experiments evaluating MULTFRC performance are included in [25]. The main issue left to be addressed for MULTFRC is its stability. This is because in MULTFRC, both *the number* of connections and *sending rate* of each connection are dynamically adjusted, potentially resulting in network instability.

While the existing approaches provide practical insight on how to improve TCP over wireless performance, it is unclear whether they can easily scale to a network as large as the Internet. Hence a general framework for flow control over wireless is needed to address both stability and scalability issues prior to any implementation. Specifically, answers to the following questions are expected from this analysis:

- How does one re-define the problem of flow control

This work was supported by NSF grant ANI-9905799, the AFOSR contract F49620-00-1-0327 and the NSF grant CCR-0225610.

over wireless network?

- 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 network flow control algorithms. A widely recognized setting, introduced by Kelly et. al. [26], is to associate a utility function with each flow, a cost function with each resource, and to maximize the aggregate *net* system utility function. Under this framework, flow control schemes can be viewed as algorithms to compute the optimal solution to this maximization problem. Kelly et. al. [26] proposed two complementary flow control algorithms, the *primal* and the *dual*.

In 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. The algorithms proposed by Kunniyur and Srikant [30], Alpcan and Basar [35], Vinnicombe [34] belong to this class. The stability issue of various primal algorithms are investigated in [26, 32, 33, 36].

In dual algorithms on the other hand, the routers adapt the prices dynamically based on the link rates, and the users select a static law to determine the source rates directly from the prices along the path and the source parameters. The algorithms proposed by Low and Lapsley [27], Paganini et. al. [39], Yaiche et. al. [29] belong to this class. The stability issue of various dual algorithms are investigated in [26, 39].

There is also another class of algorithms named primal-dual, where both prices and source rates are changed dynamically by routers and users, respectively. The algorithms proposed by Low and Lapsley [27], Kunniyur and Srikant [31], Pagnini et. al. [39] belong to this class. The stability issue of various primal-dual algorithms are investigated in [27, 31, 37, 39].

These frameworks can be used to understand and design the congestion control algorithms and predict their performance. Low pointed out that different versions of TCP and queue management algorithms such as DropTail and RED can be analyzed under the same duality model with different utility functions and update functions [28]. Kunniyur and Srikant investigated two algorithms that can be used to understand TCP's behavior [30]. Kelly showed TCP is a primal like algorithm with packet loss rate as the associated price function [40]. The stability for the system with and without delay, with and without disturbance, are reviewed and developed in [40]. The paper also discusses the selection of the TCP parameters, in order to achieve a scalable robust congestion control.

In this paper, we extend Kelly's framework in [26] 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. A sufficient and necessary condition for suboptimal system performance is derived. We then analyze existing wireless flow control approaches and propose two efficient optimal approaches. The stability with and without delay are investigated and numerical results are provided to validate our analysis. While we adopt a particular form of price function in this paper, we do not make any assumptions in the stability proofs, other than the basic one by Kelly [26] on the price function. Hence our stability results apply to a more general setting.

Our paper differs from Chiang's work [41] and Chiang and Bell's work [42] in the models applied for wireless network. We model the wireless link with a fixed capacity and a packet loss rate caused by channel error. This assumption is made by taking into account the separation between physical, transport and application layers. In contrast [41] and [42] break down the layering abstraction, and allow end users and middle nodes to change the wireless transmission power in order to vary the capacity and interference of the wireless links. As such [41] and [42] define a more general optimization problem than ours. Another key difference is that [41] and [42] assume the timescale of rate control to be large enough to achieve the wireless channel capacity, and that all packets are successfully protected from channel error. In contrast, we assume the timescale of rate control to be on the order of round trip time so that wireless channel error can still cause packet loss.

This paper is structured as follows. Section II includes problem formulation. Then two approaches addressing the problem together with global stability and delay stability issues are discussed in Sections III and IV. Section V includes discussions and future work.

II. PROBLEM FORMULATION

A. Overview of flow control framework and a primal algorithm

Consider a network with a set J of *resources*, and let C_j be the finite capacity of resource j , for $j \in J$. In practice, the resources in the network are its links. Let a *route* r be a non-empty subset of J , and denote R to be the set of possible routes. Set $a_{jr} = 1$ if $j \in r$, so that resource j lies on route 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, assume a rate x_r allocated to user r results in utility $U_r(x_r)$. $U_r(x_r)$ is assumed to be an increasing, strictly concave, and continuously differentiable function of x_r over the range $x_r \geq 0$, corresponding to the definition of elastic traffic in [43].

Let $U = (U_r(\cdot), r \in R)$, and $x = (x_r, r \in R)$. 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. [26] and later refined in [40], is a concave optimization problem maximizing the net utility:

$$\max \sum_{r \in R} U_r(x_r) - \sum_{j \in J} P_j \left(\sum_{s: j \in s} x_s \right), \quad (1)$$

where the cost function $P_j(\cdot)$ for using resource j , similar to the one introduced in [30], is

$$P_j(y) = \int_0^y p_j(z) dz. \quad (2)$$

$p_j(z)$ is called the price function and is required to be a non-negative, continuous, increasing function and not identically zero. With these assumptions on $p_j(z)$, the function $P_j(y)$ is strictly convex. One common price function used in practice is the packet loss rate¹, which is zero for no congestion, and concavely increases when there is congestion, e.g.

$$p_j(y) = \frac{(y - C_j)^+}{y}, \quad (3)$$

where C_j is the capacity of link j , and y denotes the aggregate rate passing through link j . Fig. 1.(a) shows an example of $p_j(\cdot)$ as packet loss rate on a link with capacity of 1000 bps.

The end-to-end packet loss rate for user r is $1 - \prod_{j \in r} p_j(\sum_{s: j \in s} x_s)$, which is approximately $\sum_{j \in r} p_j(\sum_{s: j \in s} x_s)$ when $p_j(\sum_{s: j \in s} x_s)$ is small. In the rest of this paper, we assume price function to be packet loss rate as shown in (3), and to be small enough so that $\sum_{j \in r} p_j(\sum_{s: j \in s} x_s)$ is the end-to-end packet loss rate for any user r .

For the logarithmic utility function $U_r(x_r) = w_r^o \log x_r$, where w_r^o is a weight and can be understood as pay per unit time, Kelly et. al. in [26] show that both

¹Even though the packet loss rate here is not a strictly increasing function, for simplicity we assume it is. In practice, this assumption makes little difference.

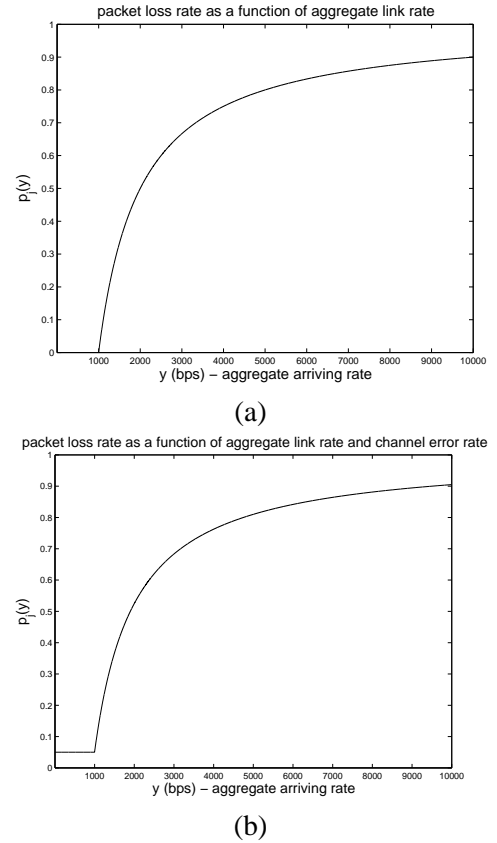


Fig. 1. Examples of the price function as packet loss rate on a link with capacity 1000 bps.

primal and dual algorithms can be used to solve the optimization problem (1). In the rest of this paper, we will focus on the primal algorithm only, as it results in a distributed end-to-end based TCP-like behavior, and closer to what is currently deployed in practice. In particular, we consider the following primal algorithm [26]:

$$\frac{d}{dt} x_r(t) = k_r \left(w_r^o - x_r(t) \sum_{j \in r} \mu_j(t) \right), \quad r \in R \quad (4)$$

with k_r being a positive scale factor or the step size affecting the adaptation rate; and

$$\mu_j(t) = p_j \left(\sum_{s: j \in s} x_s(t) \right). \quad (5)$$

Equation (4) describes the time evolution of x_r , i.e. the source rate of user r . Equation (5) indicates the generation of congestion signal at link j in terms of a congestion price function $p_j(\cdot)$. The congestion signals generated along the links on user r 's route are summed up and then fed back to the user r as the indicators of network congestion. Therefore, user exploits only aggregate information along its path. In practice users obtain feedback

prices by observing packet loss, ECN marking, or the increase in round trip time; the system in (4) corresponds to TCP-like additive increase, multiplicative decrease algorithm. Equation (5) refers to packet dropping or marking algorithm deployed on routers, e.g. RED [45] and ECN [46].

Observe that the source parameter w_r^o is proportional to the number of connections opened by user r . To see that, assume w_r^o is set to w_r^i in (4) to control the one connection's rate when user r opens one connection. User r opening $N - 1$ additional connections can be thought of adding $N - 1$ additional users along the same path as user r , and associating the same source parameter w_r^i to control their rate. Thus, the aggregate rate of these N connections, x_r , is the sum of these N users' rates, and is adjusted in a way shown in (4) but with $w_r^o = N \cdot w_r^i$. Hence the source parameter w_r^o is proportional to the number of connections opened by user r , and changing number of connections is equivalent to adjusting w_r proportionally.

Kelly has also shown the entire system to be globally asymptotically stable using a Lyapunov function, and locally robust to source rate and price disturbances. Kelly et. al. [26], Johari and Tan [32], Vinnicombe [33] study delay sensitivity in order to derive conditions under which the entire system is robust to delay. For a network, global asymptotical stability assures that given any starting rate with associated network congestion situation, the rates will eventually converge to unique finite values, avoiding congestion collapse. The robustness to delay and disturbance implies that delays and disturbances do not affect the stability of the system, i.e. the rates still converge regardless of end users receiving feedbacks from network with heterogeneous delays.

If the end users deploy the algorithm in (4) and the links in networks deploy the algorithm in (5), the unique, globally asymptotically stable rates of the entire network, denoted by $x^o = (x_r^o, r \in R)$, is given by

$$x_r^o = \frac{w_r^o}{\sum_{j \in r} p_j \left(\sum_{s: j \in s} x_s^o \right)}, \quad r \in R; \quad (6)$$

This unique solution is also optimal in the sense that the network bottlenecks are fully utilized, the total net utility is maximized, and the users are proportionally fair to each other [26].

B. Flow control over wireless

For the wireless network, we assume some of the links are associated with not only a fixed capacity but also a packet loss rate caused by the physical channel errors. Thus the end-to-end packet loss rate, i.e. the price sensing

by end-hosts via end-to-end measurement of the packet loss, now becomes noisy. In the presence of noise, flow control algorithms still aim to address the same optimization problem shown in (1), and the optimal solution is still x^o .

Assuming each link has a packet loss rate due to physical channel errors, denoted by $\epsilon_j \geq 0$, the noisy price function, denoted by $\nu_j(\cdot)$, is the total packet loss rate associate with link j as a combination of ϵ_j and $p_j(\cdot)$:

$$\begin{aligned} \nu_j(t) &= q_j \left(\sum_{s: j \in s} x_s(t) \right) \\ &= p_j \left(\sum_{s: j \in s} x_s(t) \right) + \left(1 - p_j \left(\sum_{s: j \in s} x_s(t) \right) \right) \epsilon_j \\ &\geq p_j \left(\sum_{s: j \in s} x_s(t) \right), \end{aligned} \quad (7)$$

Fig. 1(b) shows an example of $q_j(\cdot)$ as packet loss rate on a link with capacity of 1000 bps, $\epsilon = 0.05$ and $p_j(\cdot)$ shown in Fig. 1(a). When the wireless link is not congested, $q(\cdot)$ is ϵ since all the packet loss is caused by channel error; when the link is congested and packets are dropped at the router, $q(\cdot)$ gradually increases, similar to $p(\cdot)$ in Fig. 1(a).

With this noisy price, user r adjusts its rate based on the noisy feedback from the network:

$$\frac{d}{dt} x_r(t) = k_r \left(w_r^o - x_r(t) \sum_{j \in r} \nu_j(t) \right), \quad r \in R. \quad (8)$$

Observe that $\nu_j(t)$ is also a non-negative, continuous, increasing function, and not identically zero; hence following a similar analysis in [26], the system converges to a new stable point $x^* = (x_r^*, r \in R)$:

$$x_r^* = \frac{w_r^o}{\sum_{j \in r} q_j \left(\sum_{s: j \in s} x_s^* \right)}, \quad r \in R. \quad (9)$$

Comparing the stable point x^* in (9) with x^o in (6), which is the optimal solution for the optimization problem shown in Equation (1), we have $x_r^* < x_r^o$ if $\sum_{j \in r} \epsilon_j > 0$. Since the optimal solution for problem in (1) is unique, the equilibrium point x^* is then a suboptimal solution. The analysis shows that if the packet loss rate caused by channel error is not zero, then the system (8)-(7) converges to a suboptimal equilibrium point, with non-maximal net utility, and hence the network bottleneck could be underutilized. This defines the flow control problem over wireless, which we try to address in this paper.

Although the primal algorithms (8) and (4) are not exactly the same as TCP, all their behavior is similar and they face the same problem in wireless networks. All of them treat noise in the packet loss rate as the sign of congestion and reduce rates unnecessarily. Hence, it is likely that the solution and analysis for the problem of primal algorithms over wireless network could give insights for designing solutions for the problem of TCP over wireless network. In fact, the approaches we propose in this paper have strong relation with MULTFRC proposed in [25] in that they advocate the use of multiple connections for flow control over wireless.

C. Existing approaches

Existing approaches [8–24] employ a number of practical schemes to provide user r with the correct price $\sum_{j \in r} \mu_j(t)$, or its estimated value. End users can then modify the source rate control law to react to $\sum_{j \in r} \mu_j(t)$ rather than to the noisy one $\sum_{j \in r} \nu_j(t)$. Hence in most existing approaches the problem of flow control over wireless is reduced to the problem of flow control over wired networks.

Our previous approach [25], on the other hand, adjusts w_r to solve the problem. The intuition is clearly seen from (9), in that if the numerator w_r^o is changed to a larger value then it is possible to preserve $x_r^* = x_r^o$. This is a new approach and can be widely deployed in practice, since changing w_r is proportional to the number of connections in practice, adjusting w_r can be implemented by adjusting number of connections correspondingly, requiring only modifications to the application layer.

Motivated by our previous approach [25], we propose two new approaches to wireless flow control based on adjusting w_r ; and hence both of them are end-to-end application layer based schemes and require no modification to either the network infrastructure or to the network protocols. In the first approach, w_r is instantaneously adjusted using a static function with respect to $\nu_j(t)$ and $\mu_j(t)$; in the second approach, w_r is gradually adjusted by a dynamic update law.

In both approaches, we assume the end user to have access to both the correct price function $\sum_{j \in r} \mu_j(t)$ and the noisy one $\sum_{j \in r} \nu_j(t)$. In practice, users can get an estimate of the end-to-end packet loss rate $\sum_{j \in r} \nu_j(t)$ by carrying out end-to-end measurement; it is also possible to apply schemes from [17–24] to obtain an estimate of the packet loss rate caused by congestion only, i.e. $\sum_{j \in r} \mu_j(t)$. We will revisit this assumption in the discussion in Section V. Our goal is to achieve the optimal total net utility with no modification to network infrastructure, requiring control law for source rate of user r to be (8).

III. STATIC UPDATE SYSTEM

Let the source parameter for user r be a function of time, denoted by $w_r(t)$. In this system $w_r(t)$ is adjusted using a static law with respect to $\nu_j(t)$ and $\mu_j(t)$ as follows:

$$w_r(t) = w_r^o \frac{\sum_{j \in r} \nu_j(t)}{\sum_{j \in r} \mu_j(t)}, \quad (10)$$

and source rate for user r then is given by:

$$\frac{d}{dt}x_r(t) = k_r \left(w_r(t) - x_r(t) \sum_{j \in r} \nu_j(t) \right). \quad (11)$$

The intuition behind this approach is that when noise is significant and $\nu_j(t)$ is large, we increase $w_r(t)$ in (10), so as to compensate for $x_r(t) \sum_{j \in r} \nu_j(t)$ term in (11), and hence to properly control $x_r(t)$. It is easy to see the equilibrium point for the system (11)-(10) is x^o , by setting all the derivatives of $x_r(t)$ to be zero; hence the optimal solution is the equilibrium point. From both a control and a practical point of view, it is of interest to investigate the global stability and delay stability of the system, as we will do in the later subsections.

A. Global stability

Put simply, a globally, asymptotically stable system converges to its equilibrium point regardless of its initial starting point and the size of the system; for our problem, global stability ensures that in an arbitrarily large network with arbitrary number of users, any user rate $x_r, r \in R$ converges to the optimal $x_r^o, r \in R$, and the network is not overwhelmed; this means the rates are appropriately controlled according to the network congestion conditions such that bottlenecks are fully utilized, at the same time as avoiding congestion collapse (Section I in [1]). Motivated by the work in [26], we show here the global stability of system (11)-(10)-(7) through Lyapunov arguments.

Theorem 1: System (11)-(10)-(7) is globally asymptotically stable with the following Lyapunov function:

$$V(x) = \sum_{r \in R} w_r^o \log x_r - \sum_{j \in J} \int_0^{\sum_{s: j \in s} x_s} p_j(y) dy. \quad (12)$$

All trajectories converge to the equilibrium point x^o in (6) that maximizes $V(x)$.

Proof: Refer to Appendix A. ■

B. Delay stability

It is also important to investigate whether our proposed control system is stable, i.e. the rate converges to x_r^o , when

there is delay between the entities that generate the feedback signals, e.g. $\nu_j(t)$ and $\mu_j(t)$, and the controllers, e.g. the senders, that receive the feedback signals. It is difficult to prove global stability in the presence of delay. Instead, we investigate the delay stability locally, i.e. in the region around equilibrium point x^o , enabling us to make use of arguments in the frequency domain. In the presence of delay, the system (11)-(10)-(7) can be re-written as

$$\begin{aligned} \frac{d}{dt}x_r(t) &= k_r w_r^o \frac{\sum_{j \in r} \nu_j(t - d_2(j, r))}{\sum_{j \in r} \mu_j(t - d_2(j, r))} \\ &\quad - k_r x_r(t - T_r) \sum_{j \in r} \nu_j(t - d_2(j, r)) \end{aligned} \quad (13)$$

and

$$\mu_j(t) = p_j \left(\sum_{s: j \in s} x_s(t - d_1(j, s)) \right), \quad (14)$$

$$\nu_j(t) = q_j \left(\sum_{s: j \in s} x_s(t - d_1(j, s)) \right), \quad (15)$$

and

$$d_1(j, r) + d_2(j, r) = T_r \quad \forall r \in R, \quad (16)$$

where $d_1(j, r)$ 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 . T_r hence is the round trip time on route r . Here we assume T_r is fixed, as argued in [26] and [32], at least for the time scales we are interested in. The following theorem provides a sufficient condition upon each end user for the entire network to be robust to heterogeneous delay, i.e. different T_r values for different users r .

Theorem 2: The system (13)-(14) is locally stable if

$$k_r \frac{\sum_{j \in r} q_j}{\sum_{j \in r} p_j} \left(\sum_{j \in r} p_j + \sum_{j \in r} p'_j \sum_{s: j \in s} x_s^o \right) < \frac{\pi}{2T_r} \quad \forall r \in R, \quad (17)$$

where p_j, q_j is the value of $p_j(\cdot)$ and $q_j(\cdot)$ evaluated at the equilibrium point respectively; p'_j is the derivative of $p_j(\cdot)$ evaluated at the equilibrium point.

Proof: Refer to Appendix B. ■

Interpreting k_r as the speed user r adapt its rate x_r , similar to the results in [32] and [33], Theorem 2 implies that user r should adapt its rate in the time interval determined by round trip time T_r . This is because k_r is bounded by a number inversely proportional to T_r . This confirms the general intuition pointed out in [1], [32] and [33] that end users in a network should change their rate at the timescale at least on the order of its round trip time.

It is interesting to compare the delay stability condition for wireless scenario, i.e. Theorem 2 in Equation (17), to the one for wired case where $p_j(\cdot) = q_j(\cdot)$, also shown in [33], namely:

$$k_r \left(\sum_{j \in r} p_j + \sum_{j \in r} p'_j \sum_{s: j \in s} x_s^o \right) < \frac{\pi}{2T_r} \quad \forall r \in R. \quad (18)$$

As seen, the conditions for user r in wired and wireless case only differ by a route-dependent constant $\frac{\sum_{j \in r} q_j}{\sum_{j \in r} p_j}$, which is larger than one when $p_j(\cdot)$ and $q_j(\cdot)$ take the forms in (3) and (5) respectively. Hence upper bound on k_r in the wireless case, given by (17), is smaller than that in the wired case. Intuitively this can be explained as follows: since in the wireless case both $w_r(t)$ and $x(t)$ are adjusted based on feedback prices, the system is more sensitive to obsolete prices caused by delays in the system; hence the system should adjust in a more cautious way, resulting in a smaller upper bound on step size k_r .

C. Numerical results

In this subsection, we perform simulations using MATLAB to verify the results of our analysis on the static update system.

1) Simulations for Global Stability: The topology is shown in Fig. 2; there are three users in the nine node network with two wireless links in it. The wireless link is represented by dashed line and is associated with a pair of parameters: capacity C and the packet loss rate caused by channel error p_w . The setting for two wireless links are $C_1 = 6000 \text{ bps}$, $p_w^1 = 0.01$, and $C_2 = 10000 \text{ bps}$, $p_w^2 = 0.02$. End users apply (11) to control their source rate x_r , and (10) to control the source parameter w_r ; each link measures the aggregate incoming rate as the sum of the user rates, and computes the prices $p(\cdot)$ and $q(\cdot)$ using (3) and (7) respectively; in practice this corresponds to packet dropping at the bottleneck router and on the wireless links. In our simulations, the sum of the prices, i.e. $\sum_{j \in r} p_j(\cdot)$ and $\sum_{j \in r} q_j(\cdot)$, are assumed to be known to user; in practice this corresponds to users carrying out end-to-end measurements to estimate $\sum_{j \in r} q_j(\cdot)$, i.e. the end-to-end packet loss rate, and $\sum_{j \in r} p_j(\cdot)$, i.e. packet loss rate caused only by congestion.

We set w_r to be 500 when the price $\sum_{j \in r} \mu_j(t)$ is zero, in order to prevent $w_r(t)$ computed by (10) from becoming unbounded. In the context of the approach in [25], this can be understood as setting an upper limit on the number of connections one application can open. The parameters $k_r, r = 1, 2, 3$ are set to 1; w_1^o, w_2^o and w_3^o are set to 50, 40 and 30 respectively, in order to facilitate differentiating between the convergence processes of the users.

The convergence of x_r , w_r , packet loss rates caused by both channel error and congestion i.e. q_1 and q_2 , packet loss rates caused by congestion only i.e. p_1 and p_2 , and the total net utility are shown in Fig. 3. As seen, the rates and the net utility converge nicely², and the wireless links get fully utilized. As expected, $w_r(t)$ changes instantly as soon as the wireless links becomes fully utilized. This behavior is not necessarily desirable in practice. Since $w_r(t)$ is proportional to the number of connections in practice, it might cause the number of connections to change dramatically, resulting in a great deal of control overhead between end users; moreover, small stochastic disturbance on the source rate or the feedback prices might result in unnecessary fluctuation on the number of connections.

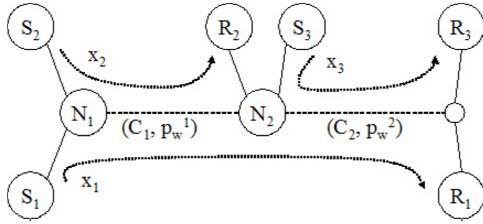


Fig. 2. Simulation topology.

2) *Simulations for Delay Stability:* To show that the system is robust to delay when conditions in Theorem 2 are met, we investigate a simple scenario where two users share a wireless link with capacity of 6000bps and packet loss rate $p_w = 0.03$. Users 1 and 2 have round trip times of $T_1 = 1$ second, and $T_2 = 1.6$ seconds respectively. The parameters are selected as in Table I: k_r is selected to be

TABLE I

PARAMETERS FOR SIMULATION FOR STATIC UPDATE SYSTEM.

para.	value	req. bound on k_r	para.	value
k_1	0.35	0.367	w_1^o	30
k_2	0.2	0.229	w_2^o	60

slightly smaller than the required upper bounds from Theorem 2; the quantities such as p_1 and p'_1 are obtained by first solving the optimization problem in Equation (1) numerically in order to compute the equilibrium point, and then evaluating the functions $p_1(\cdot)$ and $p'_1(\cdot)$ at the equilibrium point. We choose different w_r^o , $r = 1, 2$ to facilitate differentiating between the convergence processes of two users.

Convergence of x_r , w_r , packet loss rate caused by both channel error and congestion i.e. q_1 , packet loss rate

²Here we start with initial condition $x_1 = x_2 = x_3 = 0$; similar convergence observations hold for other random initial conditions.

caused by congestion only i.e. p_1 , and the total net utility are shown in Fig. 4. As seen, x_1 and x_2 converge to the optimal values around 4000 bps and 2000 bps respectively; the net utility also asymptotically converges. p_1 , q_1 and w_r initially oscillate and then converge. It is interesting to investigate the frequency of the oscillation, and the reason behind it, as it could provide insight to the transient behavior of the system, and indicate ways to suppress the oscillation. We plan to investigate this in the future.

IV. DYNAMIC UPDATE SYSTEM

The approach in Section III requires $w_r(t)$ to change instantaneously, resulting in rapid changes in $w_r(t)$ as seen in Fig. 3(b); in practice it is desirable to adjust $w_r(t)$ gradually so that (a) the number of connections is also adjusted slowly, and (b) decrease sensitivity to measurement error in packet loss rate. In this section, we consider the more complex case where $w_r(t)$ is dynamically updated, as follows: $\forall r = 1, 2, \dots, R$,

$$\frac{d}{dt}w_r(t) = c_r \left(w_r^o - w_r(t) \frac{\sum_{j \in r} p_j(\sum_{s: j \in s} x_s(t))}{\sum_{j \in r} q_j(\sum_{s: j \in s} x_s(t))} \right), \quad (19)$$

where $p_j(\cdot)$ and $q_j(\cdot)$ are defined in (3) and (7) respectively with source rate for user r changing as (11). It is easy to see that the equilibrium point for the system (11)-(19) is x^o .

The composite system built upon the equation for the rates x_r in (11) and for w_r in (19), similar to the static update system in Section III, is a nonlinear, coupled, and multivariable. Also, note that two equations are not exactly symmetrical even though they might appear to be so.

A. Global stability with two time-scale approach

In this subsection we will show that the dynamic update system, explicitly defined through (11) and (19), is endowed with some stability properties. To begin with, we make the following key observation: *the dynamics corresponding to (11) and (19) evolve in two different time scales; the first in a faster one, while the second in a slower one.* This assumption is not merely an abstraction; rather, it reflects a physical reality that $w_r(t)$, interpreted as the number of connections, is in general adjusted in a much slower time scale than source rate $x_r(t)$.

After vectorizing the relation (11) and (19), we can then

rewrite the coupled equations, as the following pair³⁴:

$$\varepsilon \dot{\mathbf{x}}(t) = \mathbf{k}\mathbf{w}(t) - \text{diag}(\mathbf{k}\mathbf{x}(t))A^T(\mathbf{p} \circ A\mathbf{x}(t)) = \mathbf{F}(\mathbf{x}); \quad (20)$$

$$\dot{\mathbf{w}}(t) = \mathbf{G}(\mathbf{w}, \mathbf{x}). \quad (21)$$

Here ε is a very small number which we introduce to reframe our current setting with the classical small-perturbation framework, for instance, [26] and [40]; this change can be trivially obtained through rescaling of the constant vectorial parameter \mathbf{k} . Since \mathbf{w} changes much slower than \mathbf{x} , we can think of $\mathbf{k}\mathbf{w}$ to be a fixed parameter within Equation (20). In this case⁵, the following holds:

Theorem 3: For the overall system (20)-(21), featuring price functions that are functions of only the aggregate rates, e.g. (3) and (7), the following is true:

- Equation (20) has a region of equilibrium points $\mathbf{F}(\mathbf{x}) = 0$ that identifies a manifold $\mathbf{w} = \mathbf{h}(\mathbf{x})$;
- Within this manifold, Equation (21), also known as the *reduced system*, has a unique global equilibrium which is the solution of $\mathbf{G}(\mathbf{h}(\mathbf{x}), \mathbf{x}) = 0$;
- The functions \mathbf{F} , \mathbf{G} , \mathbf{h} and their partial derivatives are bounded near the global equilibrium;
- The equilibrium manifold for the boundary-layer system⁶ is exponentially stable;
- The global equilibrium of the reduced system $\dot{\mathbf{w}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \dot{\mathbf{x}} = \mathbf{G}(\mathbf{x}, \mathbf{h}(\mathbf{x}))$ is asymptotically stable;

Hence, by the stability of singular disturbance non-linear system in [47] and [48], there exists an ε^* such that, $\forall \varepsilon \leq \varepsilon^*$, the global equilibrium point of the composite system is asymptotically stable.

Proof: Refer to Appendix C. ■

Similar to the interpretation of Theorem 1 in Section III-A, Theorem 3 implies that the system defined in (11) and (19), under the assumption the source rate $x_r(t)$ is adjusted much faster than the source parameter $w_r(t)$, i.e. $c_r \ll k_r$, converges to its equilibrium point regardless of its initial starting point and the size of the system; for our problem, this ensures that in an arbitrarily large network with arbitrary number of users, any user rate $x_r, r \in R$ will converge to the optimal $x_r^o, r \in R$, and the network is not overwhelmed. That is, the rates are appropriately controlled according to the network congestion condition such that the bottlenecks in the network are fully utilized, yet congestion collapse is avoided.

³⁴From now on, bold letters will denote vectorial quantities.

⁴The vectors $\mathbf{k}\mathbf{w}$ and $\mathbf{k}\mathbf{x}$ are built through the componentwise products of the vectors \mathbf{k} , \mathbf{w} and \mathbf{k} , \mathbf{x} respectively.

⁵For simplicity, we shall consider the simplified case $\mathbf{c} = \mathbf{1}$.

⁶This system is obtained through a rescaling of time, $\tau = t/\varepsilon$ and letting $\varepsilon \rightarrow 0$.

B. Simulations

In this subsection, we perform simulations using MATLAB to verify the results of our analysis on the dynamic update system.

1) *Simulations for Global stability:* The topology, network settings and parameter are exactly the same as those in the simulations for global stability of static update system in Section III-C.1. The only difference is that in this simulation end users apply (19), rather than (10) in Section III-C.1, to control the source parameter w_r . To meet the two timescale assumption made in Theorem 3, we make the update frequency of w_r to be 50 times slower than the one of x_r ; this has the similar effect as making c_r 50 times smaller than k_r . In practice, this is reflected by the requirement that the number of connections needs to change much slower than the source rates.

Convergence of x_r, w_r , packet loss rate caused by both channel error and congestion i.e. q_1 and q_2 , packet loss rate caused by congestion only i.e. p_1 and p_2 , and the total net utility are shown in Fig. 5. As seen, the rates and the net utility converge nicely⁷, and the wireless link get fully utilized; we also notice that the converged rates are exactly the same as the ones in Section III-C.1. We observe that w_r changes slowly, as compared to the simulation results in Section III-C.1 for the static update system. This behavior is desirable in practice, as it implies the number of connections, which is proportional to w_r , varies slowly, and is not sensitive to stochastic disturbances in the network.

V. DISCUSSION AND CONCLUSION

In this paper we have defined the problem of flow control over wireless to correspond to net utility maximization under noisy feedback. We have proposed two schemes to solve the problem. Both of them change the source parameter w_r only, which corresponds to changing the number of connections in practice. The first approach applies a static update law to change w_r , while the second approach applies a dynamic update law. Both systems are shown to be globally stable. The condition for delay stability on the static update system is also provided. Numerical results are provided to verify the analysis. In practice, the evolution of w_r as a function of time can be interpreted as the evolution of the number of connections as a function of time. This is because w_r can be thought of as being proportional to the number of connections. As such, a sender opens a number of connections, whereby the rate for each connection obeys Equation (8) with a w_r chosen

⁷Here we start with initial $x_1 = x_2 = x_3 = 0$; the similar convergence observations hold for other initial conditions.

as a constant corresponding to one connection only, and the aggregate rate of all connections obeys Equation (11) with aggregate $w_r(t)$ evolving as in Equation (10) or (19).

This work provides insight on how to improve the performance of TCP or TCP-friendly schemes over wireless, by only modifying the application layer to open multiple connections. Although we adopt a particular form of price function in the paper, the stability results actually hold more generally, with only the assumption that the price function is non-negative, continuous, increasing function and not identically zero.

One observation to make is that although the two proposed schemes are designed to address the flow control over wireless problem, they can also address the wired one. To see that, note that in wired network we have $\epsilon_j = 0, j \in J$, i.e. no noise in the price function. Hence we can interpret it as a special case for both static and dynamic systems with $q_j(\cdot) = p_j(\cdot)$, as implied from (7). Then all the analysis and conclusions follow, i.e. both the static update and the dynamic update system converge to the equilibrium point x^o globally, and the static update system is locally robust to delay under certain conditions. This implies one can have the same solution for flow control over both wired and wireless networks.

Even though in this paper we assume user r to have access to both $\sum_{j \in r} p_j(\cdot)$ and $\sum_{j \in r} q_j(\cdot)$, only the ratio between them is needed to control $w_r(t)$ in both the static update system in Section III and the dynamic update system in Section IV. The ratio $\sum_{j \in r} p_j(\cdot) / \sum_{j \in r} q_j(\cdot)$ is a kind of nondecreasing route-congestion indicator function in the sense that it is zero when there is no congestion along route r , and takes non-zero value when there is congestion along route r . Hence, in practice it is possible to estimate this ratio by measuring the congestion status of the route, or using any non-decreasing congestion indication function that is zero when there is no congestion and is less than or equal to one when there is congestion. In some sense, this is what MULTFRC does in [25] in order to control the number of connections. Specifically, MULTFRC estimates the queuing delay along the route and infers congestion based on whether the queuing delay is larger than a dynamic threshold, and controls the number of connections based on this inferred congestion status. It is of great interest to investigate how to estimate the ratio, how robust the proposed schemes are to the estimation error, and what performance of a congestion indicator function is as compared to the ratio. This is part of our future work.

Future work also includes the condition for the dynamic update system to be delay stable, and a closer investigation of today's TCP, to provide more precise guidance on

how to design a practical scheme; for example apply our analysis to the TCP model in [40]. It is also interesting to put the practical MULTFRC scheme, as proposed in [25], into the theoretical framework, and to arrive at guidelines for improvements.

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] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", *IEEE/ACM Transactions on Networking*, Aug. 1999
- [4] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation", *ACM SIGCOMM 98*, pp. 303 - 314
- [5] W. Tan and A. Zakhor, "Real-Time Internet Video Using Error Resilient Scalable Compression and TCP-Friendly Transport Protocol", *IEEE Transactions on Multimedia*, June 1999, vol. 1, no. 2, pp. 172-186
- [6] 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
- [7] M. Mathis, J. Semke, J. Mahdavi, T. Ott, "The macroscopic behavior of the TCP congestion avoidance Algorithm", *CCR*, July 1997, vol. 27, No. 3
- [8] 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
- [9] 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.65-9 vol.1. Piscataway, NJ, USA.
- [10] 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.
- [11] 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.
- [12] H. Balakrishnan and R. Katz, "Explicit Loss Notification and Wireless Web Performance", *IEEE Globecom Internet Mini-Conference*, Nov. 1998
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] S. Biaz, N. H. Vaidya, "Distinguishing congestion losses from wireless transmission losses: a negative result", *Computer Communications and Networks*, 1998
- [18] 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
- [19] N. Samaraweera, "Non-congestion packet loss detection for TCP error recovery using wireless links", in *IEEE Proceedings of Communications*, 146(4), pp. 222C230, Aug 1999.
- [20] 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.
- [21] 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
- [22] 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.
- [23] 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.
- [24] 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.
- [25] M. Chen and A. Zakhor, "Rate Control for Streaming Video over Wireless", *Proceeding of Infocom 2004*, Hongkong, China, March, 2004.
- [26] F. P. Kelly, A. Maulloo, D. Tan, "Rate control for communication networks: shadow prices, proportional fairness, and stability", *Journal of the Operational Research Society*, vol. 49, pp. 237-252, 1998
- [27] S. H. Low, D. E. Lapsley, "Optimization flow control I: Basic algorithm and convergence", *IEEE/ACM Trans. on Networking*, pp. 861-875, December 1999
- [28] S. H. Low, "A Duality Model of TCP and Queue Management Algorithms", *IEEE/ACM Trans. on Networking*, August 2003
- [29] H. Yaiche, R. R. Mazumdar and C. Rosenberg, "A game-theoretic framework for bandwidth allocation and pricing in broadband network", *IEEE/ACM Trans. on Networking*, pp. 667-678, October 2000
- [30] S. Kunniyur and R. Srikant, "End-to-end congestion control: utility functions, random losses and ECN marks", *Proceeding of IEEE Infocom*, Tel Aviv, Israel, March 2000
- [31] S. Kunniyur and R. Srikant, "A time scale decomposition approach to adaptive ECN marking", *Proceeding of IEEE Infocom*, pp. 1330-1339, March 2001
- [32] R. Johari, D. Tan, *End-to-End congestion control for the Internet: delays and stability*, IEEE/ACM Trans. on Networking, vol. 9, no.6, 818-832, December 2001
- [33] G. Vinnicombe, "On the stability of end-to-end congestion control for the Internet", *University of Cambridge Technical Report CUED/F-INFENG/TR.398*, <http://www.eng.cam.ac.uk/gv>, 2001
- [34] G. Vinnicombe, "Robust congestion control for the Internet", *University of Cambridge Technical Report*, <http://www.eng.cam.ac.uk/gv>, 2002
- [35] Tansu Alpcan and Tamer Basar, "A game-theoretic framework for congestion control in general topology networks", *Proc. of the 41st IEEE Conference on Decision and Control*, pp. 1218-1224, Las Vegas, NV, December 2002

- [36] Tansu Alpcan and Tamer Basar, "Global Stability Analysis of End-to-End Congestion Control Schemes for General Topology Networks with Delay", *IEEE Conference on Decision and Control*, Dec 9-12, 2003; Maui, Hawaii
- [37] Z. Wang, F. Paganini, "Global Stability with time-delay of a primal-dual congestion control", *to appear in 2003 CDC*, Maui, HI.
- [38] S. Kunniyur and R. Srikant, "End-To-End Congestion Control: Utility Functions, Random Losses and ECN Marks", *To appear in IEEE/ACM Transactions on Networking*, Oct. 2003
- [39] F. Paganini, Z. Wang, J. C. Doyle, S. H. Low, "Congestion control for high performance, stability and fairness in general network", *to appear in ACM/IEEE Trans. on Networking*
- [40] F. Kelly, "Fairness and stability of end-to-end congestion control", *European Journal of Control*, 9 pp. 159-176, 2003
- [41] M. Chiang, "To layer or not to layer: balancing transport and physical layers in wireless multihop networks" *Proceeding of Infocom 2004*, Hongkong, China, March, 2004.
- [42] M. Chiang, J. Bell "Balancing supply and demand of bandwidth in wireless cellular networks: utility maximization over powers and rates" *Proceeding of Infocom 2004*, Hongkong, China, March, 2004.
- [43] S. Shenker, "Fundamental design issues for the future Internet", *IEEE Journal of Selected Areas on Communications*, vol. 13, pp.1176-1188, 1995
- [44] C. A. Desoer, Y. T. Yang, "On the generalized Nyquist stability criterion", *IEEE Trans. on Automatic Control*, 25:187-196, 1980
- [45] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", *IEEE/ACM Trans. on Networking*, 1(4):397-413, August 1993
- [46] S. Floyd, "TCP and Explicit Congestion Notification", *ACM Computer Communication Review*, 24(5), October 1994
- [47] Shankar Sastry, "Non Linear Systems, Analysis, Stability and Control", *Springer Verlag*, 1999
- [48] Hassan Khalil, "Nonlinear Systems (3rd edition)", *Prentice Hall*, 2001

APPENDIX

A. Proof for Theorem 1

Proof: To see $V(x)$ is a Lyapunov function, first it is concave with respect to x_r , then note

$$\frac{\partial V(x)}{\partial x_r} = \frac{w_r^o}{x_r} - \sum_{j \in r} p_j \left(\sum_{s: j \in s} x_s \right),$$

thus finally

$$\begin{aligned} \frac{d}{dt} V(x(t)) &= \sum_{r \in R} \frac{\partial V(x)}{\partial x_r} \cdot \frac{d}{dt} x_r(t) \\ &= \sum_{r \in R} k_r \frac{\sum_{j \in r} q_j \left(\sum_{s: j \in s} x_s(t) \right)}{\sum_{j \in r} p_j \left(\sum_{s: j \in s} x_s(t) \right)} \\ &\quad \left(w_r^o - x_r(t) \sum_{j \in r} p_j \left(\sum_{s: j \in s} x_s(t) \right) \right)^2 \\ &\geq 0. \end{aligned}$$

Hence $V(x)$ is strictly increasing with t , unless $x(t) = x^o$. The function $V(x)$ is thus a Lyapunov function of the system. The unique value x^o maximizing $V(x)$ is a stable point of the system, to which all trajectories converge. ■

B. Proof for Theorem 2

Proof: To analysis the delay stability locally, let $x_r(t) = x_r^o + y_r(t)$, and linearize the system equations around the equilibrium $x_r^o = w_r^o / \sum_{j \in r} p_j$, we have:

$$\begin{aligned} \frac{d}{dt} y_r(t) &= -k_r y_r(t - T_r) \sum_{j \in r} q_j - k_r x_r^o \frac{\sum_{j \in r} q_j}{\sum_{j \in r} p_j} \\ &\quad \sum_{j \in r} \sum_{s: j \in s} y_s(t - d_1(j, s) - d_2(j, r)) p'_j. \end{aligned}$$

Take the Laplacian transform and carry out the simplification using matrix form, we have

$$\begin{aligned} sY(s) &= -\text{diag}\{k_r \frac{\sum_{j \in r} q_j}{\sum_{j \in r} p_j}\} \cdot \text{diag}\{e^{-sT_r}\} \cdot \text{diag}\{x_r^o\} \cdot \\ &\quad \left\{ \text{diag}\{(x_r^o)^{-1}\} \text{diag}\{\sum_{j \in r} p_j\} + M(s) \right\} Y(s), \end{aligned} \quad (22)$$

where $Y(s) = (y_1(s), y_2(s), \dots, y_n(s))^T$ is a $n \times 1$ vector with n being the number of users, and $M(s) = (m_{rq}(s))_{n \times n}$ is a $n \times n$ matrix with:

$$m_{rq} = \sum_{j \in q \cap r} p'_j \cdot e^{-s[d_1(j, q) - d_1(j, r)]}.$$

Comparing above (22) to (6) in [33], we can see they only different with a diagonal matrix $\text{diag}\left\{\frac{\sum_{j \in r} q_j}{\sum_{j \in r} p_j}\right\}$. Thus the local stability investigated here is exactly the same as the one in [33], different by only a constant diagonal matrix. Hence the rest of the proof follows the idea and logic flow in [33], and the results follow. ■

C. Proof for Theorem 3

Proof: First, it is key to realize that a system of the form (11) has a unique equilibrium point, due to the structural assumptions on the functions q_j and because of the positivity of the x_r 's. Kunniyur and Srikant, [31], have shown that a system of this form, and with the same assumption on the structure of its components, is exponentially stable⁸.

Now, given the equilibrium manifold $\mathbf{w} = \mathbf{h}(\mathbf{x})$, or more explicitly $\mathbf{w} = \text{diag}(\mathbf{x}) A^T \mathbf{q}^o$, we focus on the stability properties of the reduced order system $\dot{\mathbf{w}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \dot{\mathbf{x}} = \mathbf{G}(\mathbf{x}, \mathbf{h}(\mathbf{x}))$. In the passing, it is again important to underline that the reduced system, similar to the boundary layer one, in either of its expressions will accept a unique equilibrium.

One observation to make here about the manifold before we go on is that for any \mathbf{x} on the manifold, none of its elements is zero, resulting in $\text{diag}(\mathbf{x})$ to be full rank. This is because any \mathbf{x} on the manifold is nothing but the unique optimal solution for the concave optimization problem in (11) with corresponding $\mathbf{w} = \mathbf{h}(\mathbf{x})$; hence none of their element should be zero by the concavity of the utilities function it maximizes.

First let us calculate the Jacobian of \mathbf{h} . Let us distinguish among two cases:

- $\frac{\partial h_i}{\partial x_k} = x_k \sum_{j \in r_i \cap r_k} \frac{\partial q_j}{\partial x_k} = x_k A_{(\cdot, i)}^T \frac{\partial \mathbf{q}}{\partial x_k} A_{(\cdot, k)}$, if $i \neq k$;
- $\frac{\partial h_i}{\partial x_k} = \sum_{j \in r_k} q_j(\cdot) + x_k \sum_{j \in r_i \cap r_k} \frac{\partial q_j}{\partial x_i} = A_{(\cdot, k)}^T \mathbf{q}(\cdot) + x_k A_{(\cdot, k)}^T \frac{\partial \mathbf{q}}{\partial x_i} A_{(\cdot, i)}$, if $i = k$.

Writing down the Jacobian, we obtain:

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} &= \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_R} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \dots \\ \vdots & \dots & \ddots & \vdots \\ \frac{\partial h_R}{\partial x_1} & \dots & \dots & \frac{\partial h_R}{\partial x_R} \end{bmatrix} \\ &= \text{diag}(A_{(\cdot, i)}^T \mathbf{q}(\cdot)) + \\ &\quad \begin{bmatrix} x_1 A_{(\cdot, 1)}^T \left(\frac{\partial \mathbf{q}}{\partial x_1} A_{(\cdot, 1)} \right) & \dots & x_1 A_{(\cdot, 1)}^T \left(\frac{\partial \mathbf{q}}{\partial x_R} A_{(\cdot, R)} \right) \\ x_2 A_{(\cdot, 2)}^T \left(\frac{\partial \mathbf{q}}{\partial x_1} A_{(\cdot, 1)} \right) & \dots & \dots \\ \vdots & \ddots & \vdots \\ x_R A_{(\cdot, R)}^T \left(\frac{\partial \mathbf{q}}{\partial x_1} A_{(\cdot, 1)} \right) & \dots & x_R A_{(\cdot, R)}^T \left(\frac{\partial \mathbf{q}}{\partial x_R} A_{(\cdot, R)} \right) \end{bmatrix}. \end{aligned}$$

At this point, let us make an observation on the properties of the functions \mathbf{q} ; the scalability requirement for the network requests the

⁸Actually, what the referred paper shows is "semi-global exponential stability", which in our case reduces to exponential stability, due to the constant feature of a parameter which is instead varying in the quoted case.

⁹In the following within this proof, for the sake of clarity, we shall omit the composition operator within the functions \mathbf{p} and \mathbf{q} .

price functions of each resource to be dependent on the “aggregate rates” along that link¹⁰. Therefore, the following always holds:

$$\forall k = 1, \dots, J : \frac{\partial q_k}{\partial x_i} = \frac{\partial q_k(y_k)}{\partial y_k} \frac{\partial y_k}{\partial x_i} = \frac{\partial q_k}{\partial y_k}, \forall i = 1, \dots, R,$$

where $y_k = \sum_{s:k \in S} x_s$ is the aggregate rate passing through link k . Hence we can denote $n_k \equiv \frac{\partial q_k}{\partial y_k}$, $\forall k \in J$.

Now, observing in the passing how the rates are distributed along the rows of the second matrix, it is possible to express the Jacobian as:

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} &= \text{diag}(A_{(\cdot, i)}^T \mathbf{q}(\cdot)) + \text{diag}(\mathbf{x}) A^T \text{diag}(\mathbf{n}) A \\ &= \text{diag}(\mathbf{x}) \left\{ \text{diag}(\mathbf{x}^{-1} A_{(\cdot, i)}^T \mathbf{q}(\cdot)) + A^T \text{diag}(\mathbf{n}) A \right\}. \end{aligned}$$

The expression inside the brackets, denoted by D for later convenience, is the sum of a positive definite matrix, the first term, and a positive semi-definite one. Hence D is a positive definite matrix, and has full rank. It is then multiplied by a diagonal term which is also full rank. This allows us to claim that the Jacobian of \mathbf{h} is always full rank.

After proper simplifications, the reduced system can then be expressed as the following:

$$\dot{\mathbf{x}} = \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right)^{-1} \dot{\mathbf{w}} = \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right)^{-1} (\mathbf{w}^o - \text{diag}(\mathbf{x}) A^T \mathbf{p}), \quad (23)$$

Drawing inspiration from Kelly, [40], a Lyapunov function for the system is:

$$V(\mathbf{x}) = \sum_{r \in R} w_r^o \log x_r - \sum_{j \in J} \int_0^{\sum_{s:j \in S} x_s} p_j(y) dy. \quad (24)$$

The vector of partial derivatives is:

$$\begin{aligned} \left[\frac{\partial V}{\partial \mathbf{x}} \right] &= \left[\left(\frac{w_r^o}{x_r} - \sum_{j \in J} p_j \left(\sum_{s:j \in S} x_s \right) \right)_{r=1, \dots, R} \right]^T \\ &= (\mathbf{w}^o - \text{diag}(\mathbf{x}) A^T \mathbf{p}) \cdot \text{diag}(\mathbf{x}^{-1}); \end{aligned}$$

its componentwise zeros correspond to the equilibria in every direction of the dynamical system. Furthermore,

$$\begin{aligned} \frac{d}{dt} V(x(t)) &= \left[\frac{\partial V}{\partial \mathbf{x}} \right]^T \dot{\mathbf{x}} \\ &= (\mathbf{w}^o - \text{diag}(\mathbf{x}) A^T \mathbf{p}) \cdot \text{diag}(\mathbf{x}^{-1}) D^{-1} \text{diag}(\mathbf{x}^{-1}) \cdot \\ &\quad (\mathbf{w}^o - \text{diag}(\mathbf{x}) A^T \mathbf{p}); \end{aligned}$$

from here, it should be clear that the derivative of the Lyapunov function along trajectories of the system is always a positive number unless $x_r = x_r^o = w_r^o / \sum_{j \in J} p_j$. This is because $\text{diag}(\mathbf{x}^{-1}) D^{-1} \text{diag}(\mathbf{x}^{-1})$ is nothing but a positive definite matrix. The bottom line is that *the reduced order system is asymptotically stable*. Therefore, following the arguments in [47] and [48] for the stability of singular disturbance non-linear system, the composite system shown in (11) and (19) is asymptotically stable. ■

¹⁰This way, no need to compute the single rates at each link is required, but just to tally up the total “flow” through it. That way, we manage to have information over all the network with just knowledge of the links’ situation.

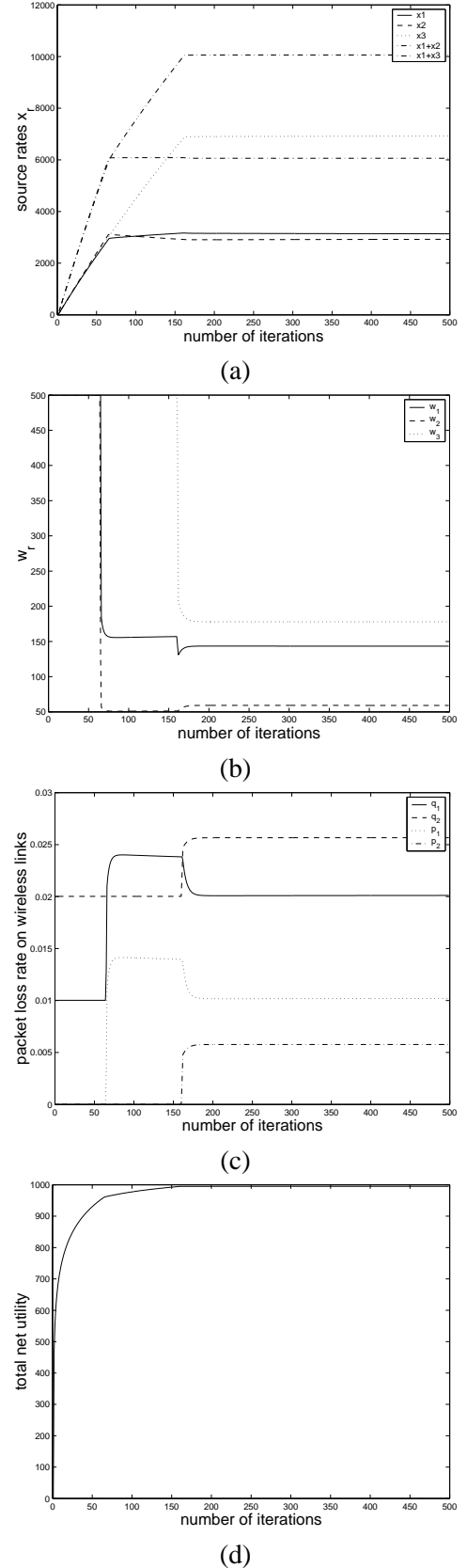
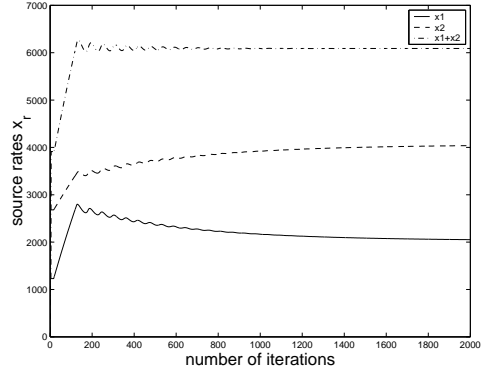
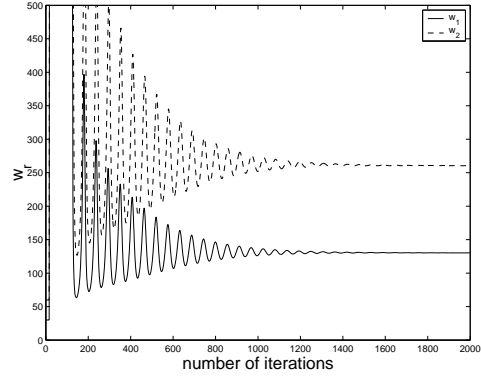


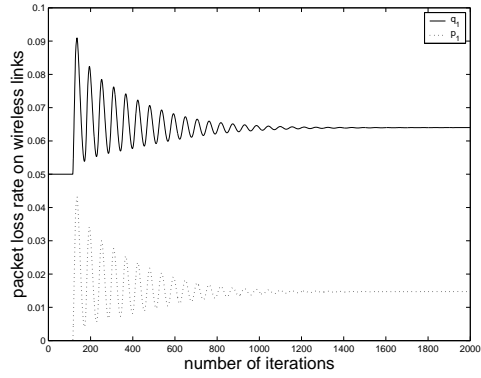
Fig. 3. Static update system: convergence of (a) rates $x_r(t)$, $r = 1, 2, 3$, (b) $w_r(t)$, $r = 1, 2, 3$, (c) packet loss rate $p_j(\cdot)$ and $q_j(\cdot)$, $j = 1, 2$, and (d) net utility, with initial rate set to 0.



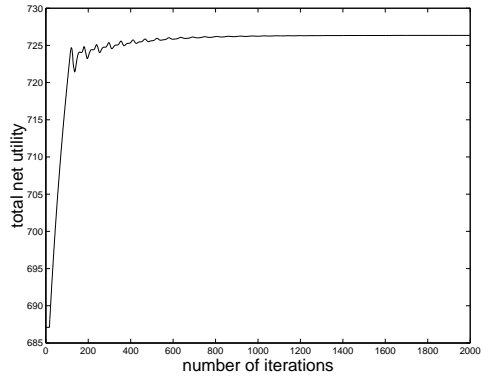
(a)



(b)

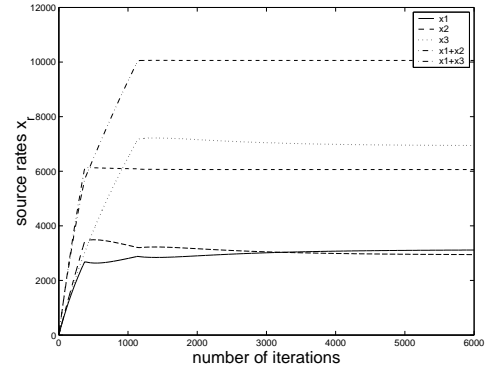


(c)

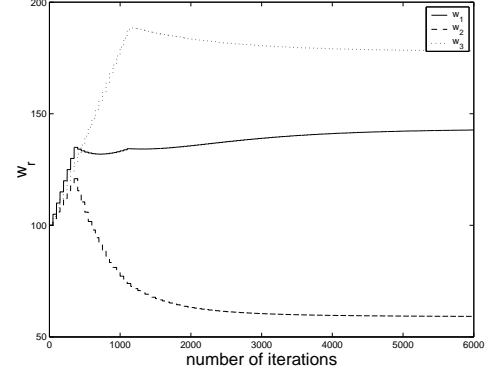


(d)

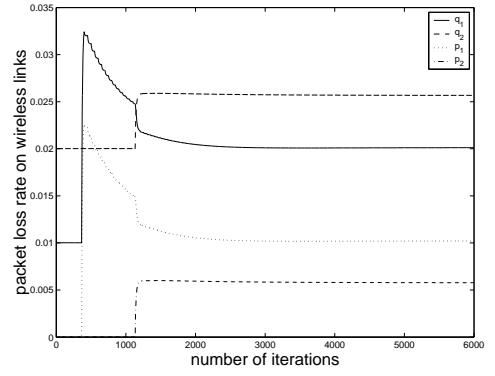
Fig. 4. Static update system with delay: convergence of (a) rates $x_r(t)$, $r = 1, 2$, (b) $w_r(t)$, $r = 1, 2$, (c) packet loss rate $p_1(\cdot)$ and $q_1(\cdot)$, and (d) net utility, with initial rate set to 0.



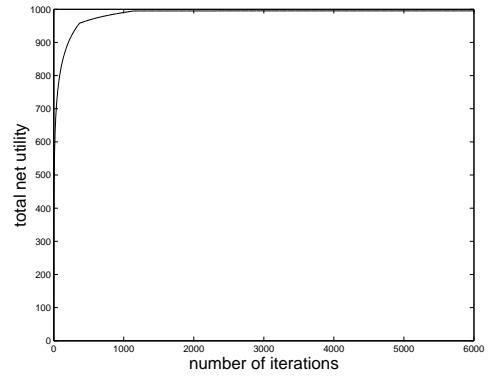
(a)



(b)



(c)



(d)

Fig. 5. Dynamic update system: convergence of (a) rates $x_r(t)$, $r = 1, 2, 3$, (b) $w_r(t)$, $r = 1, 2, 3$, (c) packet loss rate $p_j(\cdot)$ and $q_j(\cdot)$, $j = 1, 2$, and (d) net utility, with initial rate set to 0.