Congestion Control and its Stability in Networks with Delay Sensitive Traffic*

Ying Li Antonis Papachristodoulou Mung Chiang A. Robert Calderbank†

ABSTRACT

We consider congestion control in a network with delay sensitive/insensitive traffic, modelled by adding explicit delay terms to the utility function measuring user's happiness on the Quality of Service (QoS). A new Network Utility Maximization (NUM) problem is formulated and solved in a decentralized way via appropriate algorithms implemented at the users (primal) and/or links (dual). For the dual algorithm, delay-independent and delay-dependent stability conditions are derived when propagation delays are taken into account. A system with voice and data traffic is considered as example and the properties of the congestion control algorithm are assessed.

1. INTRODUCTION

Internet congestion control [20] is a distributed algorithm to allocate available resources to competing sources so as not to exceed link capacities and hence avoid congestion collapse. Congestion signals are generated at the Active Queue Management (AQM) part of the algorithm implemented at the links; the congestion measure is usually based on either delay or packet loss. The source rates are then adapted at the Transmission Control Protocol (TCP) part of the algorithm according to the size of the aggregate price signal that the user sees on the links he uses. The challenge is to use these feedback signals in order to stabilize the system around a 'fair' resource allocation equilibrium for arbitrary networks in a robust way.

What is interesting and extensively researched in the last few years is that the problem of Internet congestion control can be cast as an optimization program [12, 11] which is the framework of network utility maximization (NUM) [6]. Since the publication of the seminal paper [6] in 1998, the framework of NUM has found many applications in network resource allocation algorithms and the design of protocol stacks [4]. Consider a communication network with L logical links, wired or wireless, each with a fixed capacity of c_l bps, and S sources (i.e., end users), each transmitting at a source rate of x_s bps. Each source s emits one flow, using a fixed set $L(s)$ of links in its path, and has a utility function $U_s(x_s)$. Each link l is shared by a set $S(l)$ of sources. NUM, in its basic version, is the following problem of maximizing the aggregate utility, over the source rates $\mathbf{x}$, subject to linear flow constraints for all links:

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & \sum_s R_{ls} x_s \leq c_l, \ \forall l \\ & \mathbf{x} \succeq 0, \end{array} \quad (1)$$

where R is the routing matrix,

$$R_{ls} = \begin{cases} 1 & \text{if source } s \text{ uses link } l, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The utility of each user s is a strictly concave, continuously differentiable non-decreasing function $U_s(x_s)$, measuring the user's 'happiness' when allocated rate x_s. Essentially, it is a measure of the Quality of Service (QoS). For best-effort data traffic, a function of the transmission rate alone is good enough to reflect QoS, but for delay-sensitive traffic, a utility function that is only a function of the transmission rate of user s may not reflect user's perception of QoS. For example, for a Voice over IP (VoIP) application, the R-factor which is a measure of user's satisfactory of the QoS, has a linear term of the end-to-end average delay [1].

When the utility functions are only functions of the transmission rate, the resource allocation problem can be decomposed into a primal and a dual problem by introducing duality-based price signals [12]. In this way, the congestion measures are the 'dual' variables, while the transmission rates are the 'primal' variables. The aim of the designed AQM and TCP algorithms is to drive the congestion signals and the source rates exactly at or approximately close to the optimum of the distributed resource allocation optimization problem. In the simplest case, the structure of the dynamics that are chosen for TCP and AQM are usually based on a sub-gradient descent algorithm on the dual decomposition [6, 4, 11, 2], whose nonlinear dynamics can be shown to be asymptotically stable.

In this work, we study how to accommodate diverse applications in a network, where different types of traffic may have very different requirements for rate and delay. We extend the congestion control in [6] where the users are explicitly sensitive to the rate only, to the case where the users are explicitly sensitive to packet delay, as well as the rate. We focus on a new utility function which incorporates both the requirement on rate and delay. For user s, suppose the end-to-end average packet delay is $\sum_l R_{ls} d_l(y_l)$ where $y_l = \sum_s R_{ls} x_s$ is the link load on link l and $d_l(y_l)$ is the

†Y. Li, M. Chiang and A. R. Calderbank are with the Department of Electrical Engineering, Princeton University, Princeton, NJ, 08544, USA. Email: {yingli, chiangm, calderbk}@princeton.edu. A. Papachristodoulou is with the Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, U.K. Email: antonis@eng.ox.ac.uk.

*This research was in part supported by NSF CCF-0448012, DARPA grant HR0011-06-1-0008, and Oxford-Princeton collaboration scheme. Part of this paper was presented at Globecom 2007 and American Control Conference 2008.

average delay experienced by a packet on link l. The new utility function of user s is

$$U_s = f_s(x_s) - b_s \sum_l R_{ls} d_l(y_l) \quad (3)$$

where f_s is a function of rate x_s and $b_s > 0$ is some constant incorporating the normalization and the relative importance of the delay versus rate of user s.

The new utility function (3) has an explicit term of delay and it couples the rate of the user and the flow on the links. It explicitly reflects how the transmission rate and delay affect users' satisfaction of the service, and provides a richer framework on how to accomplish rate control for delay sensitive traffic.

We formulate a new NUM problem with this new utility function and solve it using a primal algorithm and a dual algorithm to design user and link dynamics. Compared to the basic NUM (1) we find that the dynamics for link price now depends on the delay term added in the utility function. A system with voice and data traffic is considered and a distributed resource allocation algorithm is proposed such that the utility of voice and data is jointly optimized. We find that the proposed algorithm can achieve higher quality of voice traffic and higher throughput of data traffic, sacrificing packet delays in data traffic.

Furthermore, we consider how the presence of communication delays in the network can affect the stability of the dual algorithm. Delays should not be ignored as in general their presence results in degradation of performance or even instabilities. In this paper we derive both delay-independent and delay-dependent stability conditions when propagation delays are taken into account.

The main contributions of this paper are as follows:

1. We analyze the dynamics for the NUM when explicit terms of delay in the utility functions based on the users' application are included. The analysis gives us more insight on how to accomplish optimal congestion control for users who have various applications with different requirements on rate and delay.

2. We apply the analysis to a system with voice and data traffic, and we propose distributed algorithms to allocate the resource such that the utility of voice and data is jointly optimized.

3. For the dual algorithm, delay-independent and delay-dependent stability conditions are derived for arbitrary network topologies, taking into account the communication delays in the dynamics.

The rest of the paper is organized as follows. Related work is introduced in Section 2. In Section 3, we develop algorithms that solve the NUM problem, in which user utilities are functions of the transmission rate and queueing and transmission delay. This results in a richer rate and queueing control algorithms whose dynamics are analyzed ignoring forward and backward communication delays. In Section 4, we illustrate the results on a system with voice and data traffic. In Section 5 we consider the stability of the dual algorithm in the presence of forward/backward communication delays in the dynamics, revealing a delay-independent condition, something unique for dual congestion control algorithms studied so far. Numerical examples are provided in Section 6. We conclude the paper in Section 7.

2. RELATED WORK

The seminal paper [6] focused on a utility function which is only a function of transmission rate and developed dynamics of the rate regulation at every source and the congestion price update at every link. The dynamics converge to the unique stable equilibrium which is the optimal solution of problem (1), as proved in [6]. Further studies on the stability of the dynamics have been undertaken in [12] (see also [20]). One thread of the extensions of the basic NUM framework in [6] is congestion control for delay-sensitive traffic,e.g., [9, 10, 18, 19] (see also the references therein), although this topic is under-explored.

In [19], the congestion control is studied for users with heterogeneous delay sensitivity. The utility function is $\hat{U}_s = U_s(x_s) - x_s h_s \sum_l R_{ls} d_l(y_l)$ where $y_l = \sum_s R_{ls} x_s$ is the total load on link l, $h_s \sum_l R_{ls} d_l(y_l)$ is the delay cost per unit flow, and h_s is a constant. But this utility $\hat{U}_s$ does not always reflect users' requirements for rate and delay based on users' application. For example, for VoIP application, the R-factor which is a measure of user's satisfactory of the QoS, has a linear term of the average delay, not the product of the rate and the average delay [1].

In [18], the authors study joint congestion control and routing. By reverse engineering, when the link cost includes the propagation delay, the optimization problem behind it is to maximize the total utility where the utility function has propagation delay in it. But the utility function does not include explicit average queueing delay which can be a function of users' rates.

A more general utility function is considered in [10], which is a function of rate, reliability and delay, where the delay includes queueing and transmission delay. In [10], the optimal rate-reliability-delay tradeoff is studied for a network with composite links, where each link consists of several sub-links with different rate and reliability.

In this paper, instead of examining the rate-reliability tradeoffs, we focus on the detailed equilibrium and dynamic properties for utilities that are functions of both the transmission rate the end-to-end average delay, as in (3). In particular, we analyze the stability of such dynamics, and propose an algorithm which is simpler than the one adapted from [10] to the single sub-link case.

Another thread of extending the dynamics analysis of the basic NUM in [6], some researchers have included communication delays in the dynamics and developed stability analysis conditions, such as in [21]. The problem of analyzing system behavior at the linear and nonlinear level with heterogeneous delays is difficult [17] but several procedures have been developed for that purpose: for example the methodology developed in [21] analyzes the linearizations of the nonlinear equations and the methods developed in [16] and [22] deal with nonlinear system descriptions.

In this paper we incorporate communication delays in the dynamics developed for the network with delay sensitive traffic, use the linearization of the scheme around the optimum, and derive two conditions, one based on the delay and one that is delay-independent. The basic NUM (1) [6], is thus extended in two directions: the users have utility functions which incorporate the delay and provides a richer profile for the rate adaptation; but also, communication delays in the dynamics are included and stability conditions are derived.

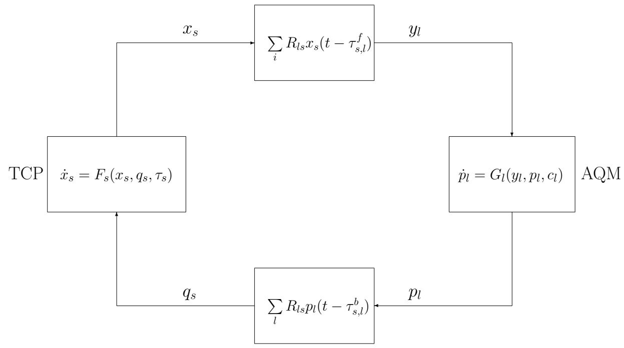

Figure 1: The Internet as an interconnection of sources and links through delays.

3. CONGESTION CONTROL FOR DELAY-SENSITIVE TRAFFIC

3.1 Basic System Review

Consider a network of L communication links shared by S sources, with routing matrix R. Associated with each source s is a transmission rate x_s. All sources whose flow passes through resource l contribute to the *aggregate rate* y_l, the rates being added with some forward time delay $\tau^f_{s,l}$:

$$y_l(t) = \sum_i R_{ls} x_s(t - \tau^f_{s,l}). \quad (4)$$

The resources l react to the aggregate rate y_l by setting a *price* p_l. This is the AQM part of the algorithm. The prices of all the links that source s uses are added to form q_s, the *aggregate price* for source s, again through a delay $\tau^b_{s,l}$:

$$q_s(t) = \sum_l R_{ls} p_l(t - \tau^b_{s,l}). \quad (5)$$

The prices q_s can then be used to set the rate x_s of source s. This is the TCP part of the algorithm, which completes the picture shown in Figure 1. The capacity of link l is denoted by c_l. The forward and backward delays can be combined to yield the Round Trip Time (RTT) for source s, τ_s:

$$\tau_s = \tau^f_{s,l} + \tau^b_{s,l}. \quad (6)$$

This setting is *universal*, and what needs to be specified are two control laws that describe how the sth source reacts to the price signal q_s that it sees

$$\dot{x}_s = F_s(x_s, q_s, \tau_s), \quad (7)$$

and how the lth router reacts to the signal y_l it observes

$$\dot{p}_l = G_l(y_l, p_l, c_l). \quad (8)$$

Here F_s models TCP algorithms (e.g. Reno, Vegas) and G_l models AQM algorithms (e.g. RED, REM).

Note that $\sum_{s \in S(l)} x_s$ is equivalent to $\sum_s R_{ls} x_s$, and similarly $\sum_{l \in L(s)} p_l$ is equivalent to $\sum_l R_{ls} p_l$. Throughout the paper, we use both notations interchangeably.

It is well known that the resource allocation algorithm can be reverse-engineered as the solution of an optimization problem [12, 20]. We associate with each user s a strictly concave, continuously differentiable non-decreasing utility function $U_s(x_s)$ when being allowed to have a transmission rate x_s. Then the optimization of the whole system with delays τ_s's assumed zeros can be cast as problem (1). The uniqueness of solution to the above problem is guaranteed since U_s are strictly concave functions.

The optimization problem (1) can be decomposed into a primal problem that the sources are trying to solve and a dual that the links are trying to solve, regarding the sources x_s as primal variables and the prices set by the links p_l as dual variables. The optimal point of the two sub-problems coincides with the optimal point of the original problem, which is unique. More details can be found in [6][12]. The dynamical system defined by (7-8) with delays ignored aims to drive the system close to or exactly at the optimal point $(\mathbf{x}^*, \mathbf{p}^*)$, using well-known sub-gradient algorithms.

The heterogeneous delays τ^f_s's and τ^b_s's are introduced to the system in order to describe the transmission and propagation time needed for the packets to reach the destination and acknowledgements to be received by the source. The presence of such delays is most of the times destabilizing and may affect greatly the performance of the system.

In this section we study the system dynamics ignoring the communication delays τ_s's, for delay sensitive traffic where the utility function depends on both rate and queueing delay, then in the next section stability analysis is studied for such system with communication delays introduced.

3.2 Delay-Sensitive Utility Maximization

Extending problem (1) to the system with new utility function (3), the optimization problem becomes

$$\begin{array}{ll} \text{maximize} & \sum_s [f_s(x_s) - b_s \sum_l R_{ls} d_l(y_l)] \\ \text{subject to} & \sum_s R_{ls} x_s = y_l, \ \forall l, \\ \text{variables} & \mathbf{x}, \mathbf{y}, \end{array} \quad (9)$$

and with feasible condition that $\mathbf{x} \succeq 0$ and $y_l < c_l$, $\forall l$. Note that the inequality sign is equivalent to the equality sign in the constraint $\sum_{s=1}^S R_{ls} x_s \leq y_l$ because the objective function is monotonically decreasing in y_l.

Assuming differentiability of the functions involved, the following conditions are necessary for $(\mathbf{x}, \mathbf{y})$ to be optimal for problem (9) [2],

$$x_s = \arg\max_{x_s} \left[f_s(x_s) - \left(\sum_l R_{ls} p_l \right) x_s \right] \quad (10)$$

$$p_l = \left(\sum_s R_{ls} b_s \right) d'_l(y_l) \quad (11)$$

$$y_l = \arg\min_{y_l} \left[\left(\sum_s R_{ls} b_s \right) d_l(y_l) - p_l y_l \right]. \quad (12)$$

3.3 The Dynamics Analysis

Here we discuss the dynamics ignoring the communication delays τ_s's. We follow a similar dynamics analysis as in [6]. We will add τ_s's back to the dynamics in Section 5.

We make the following assumptions,

ASSUMPTION 1. *Function $f_s(x_s)$ is increasing and strictly concave in $x_s \geq 0$, $\forall s$.*

ASSUMPTION 2. *Function $d_l(y_l)$ is positive, increasing and strictly convex in $y_l \geq 0$, and $d_l(c_l) = \infty$, $\forall l$.*

Note that Assumption 2 implies that $y_l \leq c_l$. The function $d_l(y_l)$ is a delay function. For instance, if d_l is the average queueing delay of an $M/M/1$ queue [7], then $d_l(y_l) = \frac{a}{c_l - y_l}$ where $a > 0$ is some constant. With this utility function, problem (9) still decentralizes and has similar properties to the one with $b_s = 0$.

3.3.1 A Primal Algorithm

Based on conditions (10-12), the dynamics can be

$$\frac{d}{dt}x_s(t) = \kappa\left(f_s'(x_s(t)) - \sum\nolimits_l R_{ls}p_l(t)\right), \tag{13}$$

where

$$p_l(t) = \left(\sum\nolimits_{s\in S(l)} b_s\right) d_l'(y_l(t)), \tag{14}$$

$$y_l(t) = \sum\nolimits_{s\in S(l)} x_s(t). \tag{15}$$

THEOREM 1. *Let*

$$\mathcal{U}(\mathbf{x}) = \sum_s f_s(x_s) - \sum_s b_s \sum_{l\in L(s)} d_l\left(\sum_{s\in S(l)} x_s\right). \tag{16}$$

Function $\mathcal{U}(\mathbf{x})$ is strictly concave in $\mathbf{x}$. The unique value $\mathbf{x}^*$ maximizing $\mathcal{U}(\mathbf{x})$ is a stable point of the dynamic system (13-15), to which all trajectories converge.

PROOF. The assumptions on f_s and d_l for all s and l ensure that $\mathcal{U}(\mathbf{x})$ is strictly concave in $\mathbf{x} \succeq 0$ with an interior maximum and the maximal $\mathbf{x}^*$ is thus unique. Setting the derivatives $\frac{\partial}{\partial x_s}\mathcal{U}(x_s) = f_s'(x_s) - (\sum_{l\in L(s)}(\sum_{s\in S(l)} b_s)$ $d_l'(\sum_{s\in S(l)} x_s)$ to zero identifies the maximum. Further

$$\begin{aligned}&\tfrac{d}{dt}\mathcal{U}(\mathbf{x}(t)) = \sum\nolimits_s \tfrac{\partial \mathcal{U}}{\partial x_s}\cdot\tfrac{d}{dt}x_s(t)\\ &= \kappa[f_s'(x_s(t)) - \sum\nolimits_l R_{ls}(\sum\nolimits_{\hat{s}} R_{l\hat{s}}b_{\hat{s}})d_l'(\sum\nolimits_s R_{l,s}x_s(t))]^2,\end{aligned}$$

establishing that $\mathcal{U}(\mathbf{x}(t))$ is strictly increasing with t, unless $\mathbf{x}(t) = \mathbf{x}^*$, the unique $\mathbf{x}$ maximizing $\mathcal{U}(\mathbf{x})$. The function $\mathcal{U}(\mathbf{x})$ is thus a Lyapunov function for the dynamic system (13-15), and the theorem follows. □

3.3.2 A dual algorithm

Based on conditions (10-12), the dynamics can be

$$\dot{p}_l(t) = \kappa_l\left[\sum_s R_{ls}x_s - d_l'^{-1}\left(\frac{p_l}{B_l}\right)\right]^+_{p_l} \tag{17}$$

$$x_s(t) = f_i'^{-1}(q_s), \tag{18}$$

where $q_s = \sum_l R_{ls}p_l$, $B_l = \sum_s R_{ls}b_s > 0$, $\kappa_l > 0$, and $[f(x)]^+_x$ means

$$[f(x)]^+_x = \begin{cases} f(x) & x>0 \\ \max\{f(x),0\}, & x=0\end{cases}.$$

THEOREM 2. *Let*

$$\begin{aligned}\mathcal{V}(p) = &\sum\nolimits_s \int_0^{\sum_{l\in L(s)} p_l} f_s'^{-1}(\eta_s)d\eta_s \\ &-\sum\nolimits_l \int_0^{p_l} d_l'^{-1}\left(\tfrac{\eta_l}{\sum_s R_{ls}b_s}\right)d\eta_l.\end{aligned} \tag{19}$$

Function $\mathcal{V}(\mathbf{p})$ is strictly concave in $\mathbf{p}$. The unique value $\boldsymbol{\lambda}^*$ maximizing $\mathcal{V}(\mathbf{p})$ is a stable point of the dynamic system (17-18), to which all trajectories converge.

PROOF. The assumptions on f_s and d_l for all s and l ensure that $\mathcal{V}(\mathbf{p})$ is strictly concave in $\mathbf{p} \succeq 0$ with an interior maximum and the maximal $\mathbf{p}^*$ is thus unique. Setting the derivatives $\frac{\partial}{\partial p_l}\mathcal{V}(p_l) = (\sum_s R_{ls}f_s'^{-1}(\sum_l R_{ls}p_l)) - d_l'^{-1}\left(\frac{p_l}{\sum_s R_{ls}b_s}\right)$ to zero identifies the maximum. Further

$$\begin{aligned}&\tfrac{d}{dt}\mathcal{V}(\mathbf{p}(t)) = \sum\nolimits_s \tfrac{\partial \mathcal{V}}{\partial p_l}\cdot\tfrac{d}{dt}p_l(t)\\ &= \kappa\left[\sum\nolimits_s R_{ls}f_s'^{-1}\left(\sum\nolimits_l R_{ls}p_l(t)\right) - d_l'^{-1}\left(\tfrac{p_l(t)}{\sum_s R_{ls}b_s}\right)\right]^2,\end{aligned}$$

establishing that $\mathcal{V}(\mathbf{p}(t))$ is strictly increasing with t, unless $\mathbf{p}(t) = \mathbf{p}^*$, the unique $\mathbf{p}$ maximizing $\mathcal{V}(\mathbf{p})$. The function $\mathcal{V}(\mathbf{p})$ is thus a Lyapunov function for the dynamic system (17-18), and the theorem follows. □

3.4 Comparison with the Dynamics for Basic NUM

We compare the dynamics above with the dynamics in [6].

For the primal algorithms, what the sources do is the same in the dynamics in this work and the dynamics in [6], but the congestion price here is $p_l(t) = (\sum_s R_{ls}b_s)d_l'(y_l(t))$ where $y_l(t) = \sum_s R_{ls}x_s(t)$ while the price in [6] is $p_l(t) = p_l(\sum_s R_{ls}x_s(t))$. Hence we find that if in [6] the price is $p_l(y_l) = (\sum_s R_{ls}b_s)d_l'(y_l(t))$ then the algorithm actually is solving problem (9).

For the dual algorithms, what the sources do is the same in the dynamics in here and in [6], but the price dynamics here is $\frac{d}{dt}p_l(t) = \kappa\left(\sum_s R_{ls}x_s(t) - d_l'^{-1}\left(\frac{p_l(t)}{\sum_s R_{ls}b_s}\right)\right)$ while the price dynamics in [6] is $\frac{d}{dt}p_l(t) = \kappa(\sum_s R_{ls}x_s(t) - h_l(p_l(t)))$. Hence we find that if in [6] the price dynamics is $\frac{d}{dt}\lambda_l(t) = \kappa(\sum_s R_{ls}x_s(t) - h_l(p_l(t)))$ where $h_l(p_l(t)) = d_l'^{-1}\left(\frac{p_l(t)}{\sum_s R_{ls}b_s}\right)$ then the algorithm actually is solving problem (9).

Note that the analysis may also be applicable to the case where the function d_l is in terms of the packet loss due to limited buffer size of the queues.

3.5 The Impact

The dynamics developed here can be used for congestion control in a network with users who may have different types of traffic, e.g., traffic with fixed/variable rate, delay sensitive/insensitive, etc. For traffic which is of fixed rate but delay sensitive, like VoIP, the basic NUM does not take it into account. The new framework in this work can accomplish congestion control for mixed traffic in one framework.

To do this, we add a coefficient a_s to f_s, so that we can incorporate the fixed rate traffic, which corresponds to $a_s = 0$. Hence the problem becomes the following,

$$\begin{array}{ll}\text{maximize} & \sum_s a_s f_s(x_s) - \sum_s b_s \sum_l R_{ls}d_l(y_l) \\ \text{subject to} & \sum_s R_{ls}x_s = y_l\ \forall l.\end{array} \tag{20}$$

We can have the following types of traffic, Traffic A: fixed rate, with delay requirement ($a_s = 0, b_s > 0$), like VoIP; Traffic B: variable rate, with delay requirement ($a_s > 0, b_s > 0$), like real-time data; Traffic C: variable rate, no delay requirement ($a_s > 0, b_s = 0$), like file-downloading.

Note that for the source s with fixed rate traffic only, where $a_s = 0$, we do not need to update x_s. For the source s with delay insensitive traffic only, where $b_s = 0$, we may approximately let b_s be a small positive number, to ensure that $\sum_s R_{ls}b_s$ is positive for all l, so that (14) and (17) are well-defined update equations.

As discussed above, compared with the basic NUM, to get the optima, we only need to change the price dynamics.

4. APPLICATIONS

We apply the dynamics to a network with voice and data traffic. Superscript ˇ and ˜ are used to indicate voice and data, respectively.

Suppose that priority queueing is used, where voice packets have high priority, and data packets have low priority. Assume that at the source nodes, the arrival processes of voice and data are independent, Poisson, and independent of the service times. Two separate queues are maintained for voice packets and data packets, respectively.

We consider the nonpreemptive priority rule whereby a packet undergoing service is allowed to complete service without interruption even if a packet of higher priority arrives in the meantime. The case of preemptive priority is similar, and will not be presented here.

By the Kleinrock independence approximation [7], the average delay of a voice packet for source $\check{s}$ is [3]

$$\delta_{\check{s}}(\check{\mathbf{y}}) = \sum_{l \in L(\check{s})} \left(\frac{K}{R_l} + \frac{K}{2R_l} \frac{\check{y}_l + \tilde{y}_l}{R_l - \check{y}_l} \right) \tag{21}$$

and the average delay of a data packet for source $\tilde{s}$ is [3]

$$\delta_{\tilde{s}}(\tilde{\mathbf{y}}) = \sum_{l \in L(\tilde{s})} \left(\frac{K}{R_l} + \frac{K}{2(R_l - \check{y}_l)} \frac{\check{y}_l + \tilde{y}_l}{R_l - \check{y}_l - \tilde{y}_l} \right), \tag{22}$$

where $\check{y}_l = \sum_{s \in \check{S}(l)} x_{\check{s}}$ and $\tilde{y}_l = \sum_{s \in \tilde{S}(l)} x_{\tilde{s}}$ are the total voice and data traffic on link l, respectively, and K is the packet size. Note that for VoIP traffic, $\check{\mathbf{x}}$ and $\check{\mathbf{y}}$ are fixed.

Suppose link l has packet error rate ν_l. The end-to-end average packet error rate of source s (either $\tilde{s}$ or $\check{s}$) is

$$p_s = 1 - \prod_{l \in L(s)} (1 - \nu_l). \tag{23}$$

For voice traffic, we consider the utility function as a function of R-factor [1]. For source $\check{s}$, its R-factor is [1]

$$\begin{aligned} R^{fac}_{\check{s}} = \;& R_a - \alpha_1 \delta_{\check{s}} - \alpha_2 (\delta_{\check{s}} - \alpha_3) H \\ & - \beta_1 - \beta_2 \log(1 + 100\beta_3 \psi_{\check{s}}), \end{aligned} \tag{24}$$

where $\delta_{\check{s}}$ is the one-way end-to-end delay in milliseconds, $\psi_{\check{s}}$ is the packet loss, and the remaining parameters are constants defined as: $R_a = 94.2$, $\alpha_1 = 0.024$, $\alpha_2 = 0.11$, $\alpha_3 = 177.3$, $H = 0$ if delay $\delta < \alpha_3$, $H = 1$ otherwise, and β_1, β_2 and β_3 are codec dependent parameters, for example, for the G.729 codec: $\beta_1 = 12, \beta_2 = 15, \beta_3 = 0.6$. Suppose we only consider packet loss due to unreliable wireless link, then $\psi_{\check{s}} = p_{\check{s}}$, where $p_{\check{s}}$ is the end-to-end packet loss of source $\check{s}$, then $R^{fac}_{\check{s}}$ is a function of delay $\delta_{\check{s}}$.

For the utility function for data source $\tilde{s}$, we can use the following weighted sum of utility on throughput and utility on delay, with the weight $w_s \in [0, 1]$ indicating the relative importance of throughput and delay [10],

$$\begin{aligned} & U_{\tilde{s}}(x_s \rho_s, \delta_s) \\ & = w_s \frac{x_s\rho_s - (x_s\rho_s)^{min}}{(x_s\rho_s)^{max} - (x_s\rho_s)^{min}} - (1 - w_s) \frac{\delta_s - \delta_s^{min}}{\delta_s^{max} - \delta_s^{min}}, \end{aligned} \tag{25}$$

where $\rho_s = 1 - p_s$. Different users can have different weights for delay and throughput, and different maximum delay constraints based on their traffic types.

To characterize the tradeoff of voice and data, we optimize the weighted sum of voice utility and data utility [10]. The generalized NUM problem is formulated as

$$\begin{array}{ll} \text{maximize} & \frac{v}{|\check{S}|} \sum_{\check{s}} U_{\check{s}}(R^{fac}_{\check{s}}) + \frac{1-v}{|\tilde{S}|} \sum_{\tilde{s}} U_{\tilde{s}}(x_{\tilde{s}} \rho_{\tilde{s}}, \delta_{\tilde{s}}) \\ \text{subject to} & \check{y}_l + \tilde{y}_l \le R_l, \forall l \\ & R^{fac}_{\check{s}} \le R^{fac}_{\check{s}}(\delta_{\check{s}}), \forall \check{s} \\ & \delta_{\check{s}}(\check{\mathbf{y}}) \le \delta_{\check{s}}, \forall \check{s} \\ & \delta_{\tilde{s}}(\tilde{\mathbf{y}}) \le \delta_{\tilde{s}}, \forall \tilde{s} \\ \text{variables} & \mathbf{R}^{fac}, \tilde{\mathbf{x}}, \check{\boldsymbol{\delta}}, \tilde{\boldsymbol{\delta}}, \end{array} \tag{26}$$

and the box constraints on $\mathbf{R}^{fac}$, $\check{\boldsymbol{\delta}}$, $\tilde{\mathbf{x}}$ and $\tilde{\boldsymbol{\delta}}$, where $v \in [0, 1]$ is a constant, $R^{fac}_{\check{s}}(\delta_{\check{s}})$ is as in (24), $\delta_{\check{s}}(\check{\mathbf{y}})$ and $\delta_{\tilde{s}}(\tilde{\mathbf{y}})$ are as above. The utility function $U_{\tilde{s}}$ is given by (25) and for $U_{\check{s}}$, we use $U_{\check{s}}(R^{fac}_{\check{s}}) = R^{fac}_{\check{s}} / (R^{fac,max}_{\check{s}} - R^{fac,min}_{\check{s}})$.

In problem (26), the first constraint is the rate constraint. The second states that the network must provide an R-factor to each VoIP source that is no less than the requested R-factor. The third and fourth ensure the delay requirements.

Note that we use δ to denote delay as a function and as the value of this function (and similarly for R^{fac}).

4.1 A Distributed Algorithm

By adapting Algorithm 2 in [10] for the link composed by several sub-links, we can have a corresponding distributed algorithm for the case of every link having one sub-link (the case in this work). The algorithm needs every voice source, data source and link solve its optimization problem, and update either the willingness to pay or the congestion price.

For priority queueing, if the voice traffic is low compared with the link capacity, the average delay is small. When the average delay is less than 177.3 ms, by (24), in R-factor only the term of linear delay $\alpha_1 \delta$ affects the optimal solution. Adapting the dynamics in Section 3, we can explicitly find how to update the congestion price, and how to adjust the rate of the data source. Voice source doesn't need to solve any optimization problem, and the variables representing the willingness to pay are not needed for voice or data source. The dynamics becomes simpler than the one by Algorithm 2 adapted to the single sub-link case in [10].

The problem can be simplified to

$$\begin{array}{ll} \text{maximize} & -\sum_{\tilde{s}} v_{0\tilde{s}} \sum_{l \in L(\tilde{s})} \tilde{y}_l + \sum_{\tilde{s}} v_{1\tilde{s}} x_{\tilde{s}} \\ & -\sum_{\tilde{s}} v_{2\tilde{s}} \sum_{l \in L(\tilde{s})} \frac{1}{R_l - \check{y}_l - \tilde{y}_l} \\ \text{subject to} & \sum_{\tilde{s} \in \tilde{S}(l)} x_{\tilde{s}} = \tilde{y}_l, \forall l \\ \text{variables} & \tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \end{array} \tag{27}$$

where $v_{0\tilde{s}}$, $v_{1\tilde{s}}$ and $v_{1\tilde{s}}$ are corresponding coefficients.

To guarantee the convergence, we add $-\sum_{\tilde{s}} \epsilon x^2_{\tilde{s}}$ in the objective function, to make it strictly concave in $x_{\tilde{s}}$, where ϵ is a positive number, and it is sufficiently small to ensure the objective function to be increasing in $x_s \in [x_s^{min}, x_s^{max}]$, and to guarantee a small approximation error.

We adapt the dynamics of the dual algorithm in Section 3, and the following distributed algorithm is proposed to find the optimal solution. Note that the term of $\tilde{y}_l$ in problem (27) is not in the exactly same form as the term of y_l in problem (9), but easily we can adapt the dynamics for problem (9) to the dynamics for problem (27) by following the first order optimal conditions of problem (27) [2]. The algorithm can also be derived by Lagrange dual decomposition and subgradient method [2].

Distributed Algorithm:

At data source $\tilde{s}$:

At each iteration t, at data source $\tilde{s}$, its rate is computed,

$$x_{\tilde{s}}(t) = \min \left\{ x^{max}_{\tilde{s}}, \max \left\{ x^{min}_{\tilde{s}}, \frac{v_{1\tilde{s}} - \lambda^{\tilde{s}}(t)}{2\epsilon} \right\} \right\} \tag{28}$$

where $\lambda^{\tilde{s}}(t) = \sum_{l \in L(\tilde{s})} \lambda_l(t)$ is the end-to-end congestion price at iteration t.

At link l:

At each iteration t, at link l, its congestion price is updated by

$$p_l(t+1) = \left[p_l(t) - \alpha(t) \left(\tilde{y}_l(t) - \tilde{x}^l(t) \right) \right]^+, \forall l, \tag{29}$$

$\tilde{y}_l(t) = \min\{\tilde{y}_l^{max}, \max\{\tilde{y}_l^{min}, R_l - \check{y}_l - \sqrt{\frac{\sum_{\tilde{s}\in\tilde{S}(l)} v_{2\tilde{s}}}{p_l(t)-\sum_{\check{s}\in\check{S}(l)} v_{0\check{s}}}}\}\}$ if $p_l(t) > \sum_{\check{s}\in\check{S}(l)} v_{0\check{s}}$, otherwise $\tilde{y}_l(t) = \sum_{\tilde{s}\in\tilde{S}(l)} x_{\tilde{s}}^{min}$, where $\tilde{x}^l(t) = \sum_{\tilde{s}\in\tilde{S}(l)} x_{\tilde{s}}(t)$ is the aggregate rate of data traffic on link l at iteration t, $\alpha(t)$ is the step size, and $[a]^+ = \max\{a, 0\}$.

The message passing is needed for source to get its end-to-end congestion price $q^{\tilde{s}}(t) = \sum_{l\in L(\tilde{s})} p_l(t)$, and for link to get its aggregate rate $\tilde{x}^l(t) = \sum_{\tilde{s}\in\tilde{S}(l)} x_{\tilde{s}}(t)$, the aggregate coefficients of voice source and data source on it, i.e., $\sum_{\check{s}\in\check{S}(l)} v_{0\check{s}}$ and $\sum_{\tilde{s}\in\tilde{S}(l)} v_{2\tilde{s}}$.

For the subgradient method [2], the step size can be the one satisfying $\lim_{t\to\infty}\alpha(t) = 0$ and $\lim_{t\to\infty}\sum_{i=1}^{t}\alpha(i) = \infty$, e.g., $\alpha(t) = \beta/t$, $\beta > 0$. The algorithm can converge to the optimal solution of problem (27).

5. WITH COMMUNICATION DELAYS

In this section, we will concentrate on the dual congestion control scheme with dynamics at the links but a static source law. The stability properties of the undelayed system (assuming τ_s's be zeros) has been obtained in Theorem 2.

We consider communication delays, which describe the transmission and propagation time needed for the packets to reach the destination and acknowledgements to be received by the source. When the communication delays during packet/acknowledgement transfer are introduced, say, when τ_s's are in the dynamics, the dynamic analysis becomes more complicated, and a scalable analysis methodology is difficult. Here we will obtain two results that hold for arbitrary network topologies: one delay-independent and the other delay-dependent. The tools we use are based on [21].

Note that the delay $d_l(y_l)$ in the utility function for the delay-sensitive traffic considered here is mainly the queueing delay, which provides a richer profile for rate control; the delay τ_s in our dynamic analysis is the communication delay which includes the queueing delay, propagation delay, processing delay, and so on. For the simplicity of the analysis, τ_s's are assumed to be constants. In future work, we will consider the case where τ_s's are time varying.

The routing matrix R is assumed to be fixed and full row rank. This means that there are no algebraic constraints between link flows, i.e., they can vary independently by choice of source flows x_s. As a consequence, equilibrium prices are uniquely determined.

5.1 Dynamics for the Model with Communication Delays

The presence of delays is most of the times destabilizing and may affect greatly the system performance. Stability analysis of linear time delay systems has been investigated greatly in the past years [13]. Just as in the stability analysis of system described by linear *Ordinary* differential equations, there are in general two methodologies for investigating stability: using time-domain (Lyapunov) or frequency domain arguments. Using a frequency domain methodology more accurate descriptions of the stability boundaries can be obtained and this method is scalable for the special case of Internet congestion control [21]. On the other hand, Lyapunov-based arguments are more conservative; they are however useful for the exact investigation of the stability of nonlinear systems [8]. Here we will analyze the system described in Section 3 using a generalized Nyquist criterion.

The nonlinear delayed model when heterogeneous time-delays are taken into account becomes

$$\begin{aligned} \dot{p}_l(t) &= \kappa_l \left[\sum_{s=1}^{S} R_{ls} x_s(t-\tau_{s,l}^f) - d_l'^{-1}\left(\frac{p_l(t)}{B_l}\right)\right]_{p_l}^+ \\ x_s(t) &= f_s'^{-1}(q_s(t)) \\ q_s(t) &= \sum_{l=1}^{L} R_{ls} p_l(t-\tau_{s,l}^b). \end{aligned}$$

We define the following matrices to simplify the notation:

$$F = \mathrm{diag}\{f_s''(x_s^*)\} < 0, \quad D = \mathrm{diag}\{d_l''(y_l^*)\} > 0,$$
$$K = \mathrm{diag}\{\kappa_l\} > 0, \quad B = \mathrm{diag}\{B_l\} > 0,$$

$$[R_f(\theta)]_{ls} = \begin{cases} e^{-\theta\tau_{s,l}^f} & \text{if user } s \text{ uses link } l \\ 0 & \text{otherwise} \end{cases}$$

$$[R_b(\theta)]_{ls} = \begin{cases} e^{-\theta\tau_{s,l}^b} & \text{if user } s \text{ uses link } l \\ 0 & \text{otherwise} \end{cases}$$

where θ is the Laplace variable. Note that

$$R_b(\theta) = R_f(-\theta)\mathrm{diag}\{e^{-\theta\tau_s}\}.$$

Linearizing the nonlinear time-delayed system about the equilibrium

$$y_l^* = d_l'^{-1}\left(\frac{p_l^*}{B_l}\right), \quad x_s^* = f_i'^{-1}(q_s^*),$$

given the relations $y_l^* = \sum_s R_{ls} x_s^*$ and $q_s^* = \sum_{l=1}^{L} R_{ls} p_l^*$, we get:

$$\dot{\delta p_l}(t) = \kappa_l \left[\sum_s R_{ls}\delta x_s(t-\tau_{s,l}^f) - \frac{\delta p_l(t)}{B_l d_l''(y_l^*)}\right] \quad (30)$$

$$\delta x_s(t) = \frac{1}{f_s''(x_s^*)}\delta q_s(t) \quad (31)$$

$$\delta q_s(t) = \sum_{l=1}^{L} R_{ls}\delta p_l(t-\tau_{s,l}^b). \quad (32)$$

In the following two subsections, we will present two stability results for the linearization, one that is delay-independent and one that is delay-dependent.

5.2 Delay-Dependent Sufficient Condition for Stability

THEOREM 3. *Given Equations (30)-(32), for* $\kappa_l = \frac{1}{y_l^*}$ *and* $f_s''(x_s^*) = -\frac{M_s\tau_s}{x_s^*\alpha_s}$ *where* $\alpha_s < \pi/2$ *and* $M_s = \sum_{l=1}^{L} R_{ls}$ *the equilibrium is asymptotically stable.*

PROOF. Taking Laplace transforms and dropping all δ's:

$$\begin{aligned} \theta p(\theta) - p(0) &= K\left(R_f(\theta)x(\theta) - B^{-1}D^{-1}p(\theta)\right) \\ x(\theta) &= F^{-1}q(\theta), \quad q(\theta) = R_b^T(\theta)p(\theta). \end{aligned}$$

Combining we get:

$$\theta p(\theta) - p(0) = K\left(R_f(\theta)F^{-1}R_b^T(\theta) - B^{-1}D^{-1}\right)p(\theta).$$

Therefore,

$$\begin{aligned} p(0) = \; & \left(\theta I + KB^{-1}D^{-1}\right)p(\theta) \\ & + KR_f(\theta)(-F^{-1})T(\theta)R_f^T(-\theta)p(\theta). \end{aligned}$$

This system is stable if its poles lie in the left half plane, i.e., if the solution to

$$\det\left(\left(\theta+\frac{\kappa_l}{B_l\hat{d}_l}\right)I+\left(\theta+\frac{\kappa_l}{B_l\hat{d}_l}\right)G\right)=0$$

has only negative real parts, where the return ratio is

$$G=\operatorname{diag}\left\{\frac{\kappa_l}{s+\frac{\kappa_l}{B_l\hat{d}_l}}\right\}R_f(\theta)\operatorname{diag}\left\{\frac{\tau_s}{-\hat{f}_s}\right\}\operatorname{diag}\left\{\frac{e^{-\theta\tau_s}}{\tau_s}\right\}R_f^T(-\theta)$$

where $\hat{f}_s=f_s''(x_s^*)$ and $\hat{d}_l=d_l''(y_l^*)$. We now show that if $\hat{f}_s=-\frac{M_s\tau_s}{x_s^*\alpha_s}$ and $\kappa_l=\frac{1}{y_l^*}$, then for $\alpha_s<\pi/2$ the equilibrium is asymptotically stable for arbitrary topologies.

Since the open-loop system is stable, we need to ensure that the eigenvalues of the above function G, for $\theta=j\omega$ do not encircle the -1 point. These eigenvalues are the same as the eigenvalues of

$$\operatorname{diag}\left(\frac{j\omega}{j\omega+\frac{\kappa_l}{B_l\hat{d}_l}}\right)\hat{R}(j\omega)\operatorname{diag}\left(\frac{\alpha_s e^{-j\omega\tau_s}}{j\omega\tau_s}\right)\hat{R}^T(-j\omega)$$

where

$$\hat{R}(j\omega)=\operatorname{diag}\left\{\sqrt{\frac{1}{y_l^*}}\right\}R_f(j\omega)\operatorname{diag}\left\{\sqrt{\frac{x_s^*}{-M_i}}\right\}.$$

Note $\overline{\sigma}^2(\hat{R}(j\omega))\leq 1$ where $\overline{\sigma}^2(\hat{R}(j\omega))=\rho(\hat{R}(j\omega)\hat{R}^T(-j\omega))$, where $\rho(Z)$ denotes the spectral radius of a matrix Z. The argument is similar as the one in [21].

Now if λ is an eigenvalue of the above, then there exists a v for which $\|v\|=1$ such that

$$\operatorname{diag}\left(\frac{j\omega}{j\omega+\frac{\kappa_l}{B_l\hat{d}_l}}\right)\hat{R}(j\omega)\operatorname{diag}\left(\frac{\alpha_s e^{-j\omega\tau_s}}{j\omega\tau_s}\right)\hat{R}^T(-j\omega)v=\lambda v$$

and hence

$$\lambda=\frac{v^*\hat{R}(j\omega)\operatorname{diag}\left(\alpha_s\frac{e^{-j\omega\tau_s}}{j\omega\tau_s}\right)\hat{R}^T(-j\omega)v}{v^*\operatorname{diag}\left(1-\frac{k_l}{B_l\hat{d}_l j\omega}\right)v}$$

Since $\|\hat{R}^T(-j\omega)v\|\leq 1$ this implies $\lambda\in\frac{\operatorname{Co}\left(0\cup\left\{\alpha_s\frac{e^{-j\omega\tau_s}}{j\omega\tau_s}\right\}\right)}{\operatorname{Co}\left\{\left(1-\frac{k_l}{B_l\hat{d}_l j\omega}\right)\right\}}$ and since the eigen-loci cannot cross the real axis at or to the left of the point -1 for $\alpha_s<\pi/2$, the closed loop system is stable by the generalized Nyquist stability criterion [21]. □

We note that the condition is similar to the one that is given in [14] - however, the dynamics that we are considering are far more interesting, as they take into account delay-sensitive traffic.

REMARK 3. *In general, it may be difficult for the resources to be able to estimate y_l^* and for the sources to know M_s. However it is possible to relax the condition that $\kappa_l=\frac{1}{y_l^*}$ to the condition that $\kappa_l=\frac{1}{c_l}$, which can be implemented in a decentralized way. Note that due to the nature of the algorithm, $y_l^*<c_l$, i.e., the links may not be fully utilized.*

5.3 Delay-Independent Sufficient Condition for Stability

Here we present a delay-independent stability condition.

THEOREM 4. *Given Equations (30)-(32), if $\kappa_l=\frac{1}{y_l^*}$, $f_s''(x_s^*)=-\frac{M_s}{\alpha_s x_s^*}$, $\kappa_l>B_l d_l''(y_l^*)$ and $\alpha_s<1$ then the equilibrium is asymptotically stable.*

PROOF. Again, we write $\hat{f}_s=f_s''(x_s^*)$ and $\hat{d}_l=d_l''(y_l^*)$. The return ratio G in this case is the same as in the proof of Theorem 3.

The eigenvalues of G are the same as the ones of

$$\operatorname{diag}\left\{\frac{1}{\theta+\frac{\kappa_l}{B_l\hat{d}_l}}\right\}R_f(\theta)\operatorname{diag}\left\{\frac{1}{-\hat{f}_s}\right\}\operatorname{diag}\left\{e^{-\theta\tau_s}\right\}R_f^T(-\theta)\operatorname{diag}\left\{\kappa_l\right\}$$

In the same way as before, and imposing that $\hat{f}_s=-\frac{M_s}{x_s^*\alpha_s}$ and that $\kappa_l=\frac{1}{y_l^*}$ we get:

$$\lambda=\frac{v^*\hat{R}(j\omega)\operatorname{diag}\left(\alpha_s e^{-j\omega\tau_s}\right)\hat{R}^T(-j\omega)v}{v^*\operatorname{diag}\left(\frac{\kappa_l}{B_l\hat{d}_l}+j\omega\right)v}$$

In this case, for asymptotic stability we require that $\kappa_l>B_l\hat{d}_l$ and $\alpha_s<1$. This condition is delay-independent. □

Remark 3 also holds in this case. Note that the delay-independent condition allows us to have a fixed gain at the sources which is irrespective of the size of the delay, i.e., the sources don't have to compensate their gain for long delays. The price, of course, is performance degradation.

The delay-independent condition above ensures that the Nyquist plot not only does not encircle the -1 point (which is what we ensured in the delay-dependent condition), but rather that the whole Nyquist plot never leaves the unit disc.

5.4 A Simple Network

In order to put the above results in perspective, we will consider the stability conditions we get by looking at a simple, single-link single-source network with the ones given by Theorems 3 and 4 when they are reduced to this simple case.

The model for a simple such network reads:

$$\delta\dot{p}=\kappa\left(\frac{1}{\hat{f}}\delta p(t-\tau)-\frac{\delta p(t)}{B\hat{d}}\right).\qquad(33)$$

where $\hat{f}=f''(x^*)$ and $\hat{d}=d''(x^*)$. This system is delay-dependent stable [5] if $-\frac{1}{\hat{f}}>\frac{1}{B\hat{d}}$ and $\tau<\frac{\arccos\left(\frac{\hat{f}}{B\hat{d}}\right)}{\kappa\sqrt{\frac{1}{\hat{f}^2}-\frac{1}{B^2\hat{d}^2}}}$. Theorem 3 requires that $\kappa=\frac{1}{x^*}$, $\hat{f}=-\frac{\tau}{x^*\alpha}$, $\alpha<\pi/2$ and imposes $\tau<\frac{\arccos\left(-\frac{\tau}{\alpha B\hat{d}x^*}\right)}{\frac{1}{x^*}\sqrt{\frac{\alpha^2x^{*2}}{\tau^2}-\frac{1}{B^2\hat{d}^2}}}$. This condition is valid for $\alpha B\hat{d}x^*>\tau$. Let us introduce a variable $\mu=\frac{\tau}{x^*\alpha B\hat{d}}$, $\mu\in[0,1]$. Then the above condition reads $\alpha\sqrt{1-\mu^2}<\arccos(-\mu)$. Indeed this condition is only valid for $\alpha<\frac{\pi}{2}$. Hence the two conditions, the one given in Theorem 3 for arbitrary network sizes and the one specific to (5.4) are equivalent; therefore the delay-dependent condition is necessary and sufficient for the single-source single-link case.

System is delay-independent stable if $\frac{1}{B\hat{d}}>\frac{1}{-\hat{f}}$. The delay-independent condition that we have in Theorem 4 reduces in the single-source single-link case to $\kappa=\frac{1}{x^*}$, $\hat{f}=-\frac{1}{\alpha x^*}$, $B\hat{d}<\kappa$, $\alpha<1$. Combining these we get $B\hat{d}<-\alpha\hat{f}$, and since we required $\alpha<1$ in the conditions of Theorem 4, the two conditions are again equivalent in the simplest network case.

6. SIMULATION STUDIES

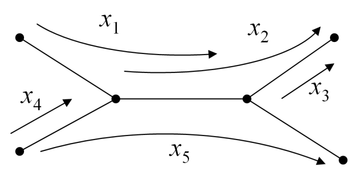

Figure 2: Network topology and flow routes.

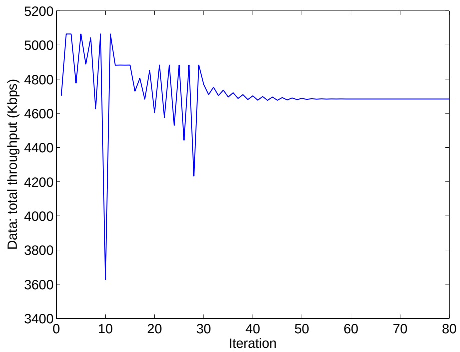

Figure 3: Convergence of Algorithm 1: Total data rate w.r.t. iterations.

6.1 Ignoring Communication Delays

In this section, we present numerical examples by analyzing a simple network, shown in Fig. 2. On each stream, there are both VoIP and data traffic. Let the maximum supportable rate on each link be $R_l = 2$ Mbps. Suppose the packet length is $K = 400$ bits. Suppose that every link is with average packet error rate $\nu = 0.1063$. Note that we assume a fixed packet error rate.

We use distributed Algorithm 1 to compute the optimal rates for the sources with data traffic by solving problem (26), where in the objective function the weight for the VoIP is $v = 0.5$ and the weight for the data throughput is $w = 0.5$. The total data rate of all the users with respect to the iteration is shown in Fig. 3. The convergence can be seen from the figure. The solution of the rates is readily verified to be optimal by comparing with the results from the centralized algorithm.

To see how the data and voice traffic tradeoff when they share the network, and to see how much advantage priority queueing can provide, we plot different curves for the tradeoff of the throughput of the data traffic and R-factor of the voice traffic in Fig. 4, and the corresponding average packet delay in Fig. 5. In the figures, v and $(1 - v)$ are the weights of the voice and data traffic in the objective function of problem (26), respectively, and w and $(1 - w)$ are the weights of the throughput and the delay of the data traffic, respectively. In Fig. 4, there is only one curve which is for the case where we include the packet loss due to the packet delay exceeding some deadline, as well as due to the unreliable wireless link.

From the figures, it can be seen that the priority queueing can achieve higher R-factor of voice traffic and higher throughput of data traffic, with the sacrifice of the packet delay of data traffic. The packet loss of voice traffic due to

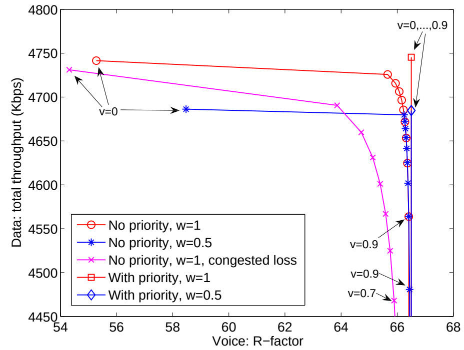

Figure 4: Tradeoff between voice and data traffic: R-factor and throughput.

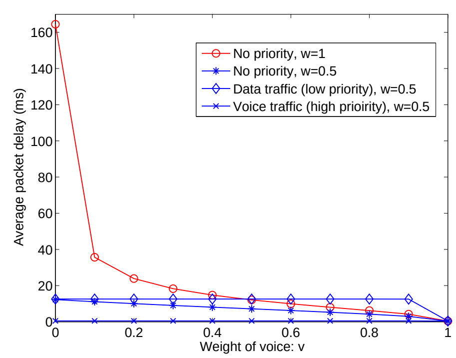

Figure 5: Average packet delay w.r.t different weight v.

exceeding the deadline can affect the R-factor a lot. Priority queueing not only helps the average packet delay of voice traffic, but also it reduces the packet loss due to exceeding the deadline.

In Fig. 4, it can be seen that no matter how we change the weight of the voice v, the achieved R-factor by priority queueing is almost the same. So the R-factor achieved is not adaptive. But if we allow some of the voice packets to have high priority and the remaining voice packets to have low priority, then the system can achieve all the possible R-factor from the one achieved by priority queueing to the one achieved by no priority, by adjusting the fraction of the packets with high priority.

In Fig. 5, it can be seen that for the cases of no priority, as the voice weight v increases, the packet delay decreases, which is because that the higher weight v indicates the system favors higher R-factor, and to get higher R-factor, the packet delay should be reduced. But for the cases of priority, if it is delay insensitive data traffic, $w = 1$, the average delay of the data packet is about 1500 ms, which is not shown in the figure; if it is delay sensitive data traffic, for $w = 0.5$, the figure shows that the average packet delay of data or voice almost keeps the same when v grows. This is because for the priority case, due to higher priority, the voice packet can get very small average packet delay in this example, so it is not sensitive to v, except when $v = 1$.

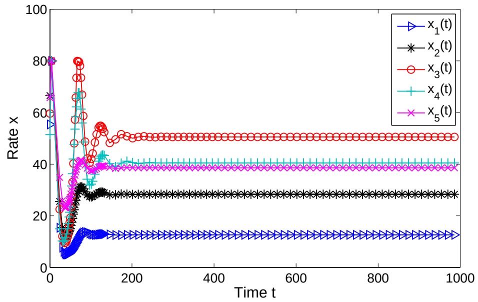

Figure 6: Stable (under delay-dependent condition): $(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5) = (10, 20, 10, 10, 20)$ **,** $\alpha_s = \pi/2 - 0.1$**.**

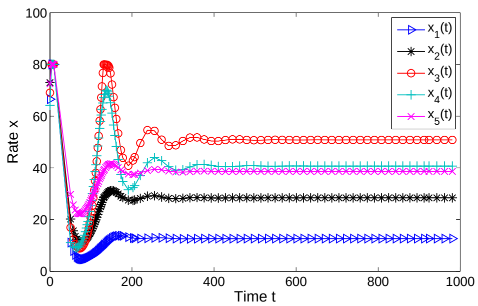

Figure 7: Stable (under delay-dependent condition): $(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5) = (20, 40, 20, 20, 40)$**,** $\alpha_s = \pi/2 - 0.1$**.**

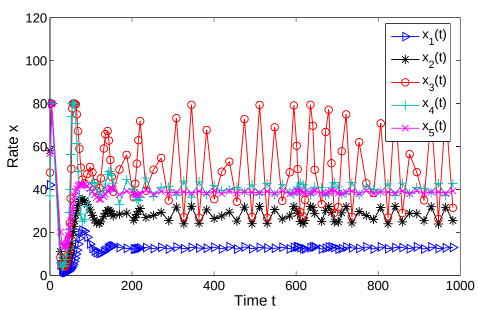

Figure 8: Unstable (violating delay-dependent condition): $(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5) = (10, 20, 10, 10, 20)$**,** $\alpha = \pi/2 + 1$**.**

6.2 Considering Communication Delays

In order to illustrate our results in the previous section, we will use the same network of 5 links and 5 users as shown in Fig. 2 as an example. Assume each flow is with delay sensitive data traffic.

6.2.1 *Delay-Dependent Condition for Stability*

To satisfy the delay-dependent sufficient condition as in Theorem 3, we let $f_s(x) = \frac{M_s \tau_s}{\alpha_s} x_s(1 - \log(x_s/x_s^{max}))$, where x_s^{max} is the maximum allowable rate of source s x_s. Then we have $f_s^{'-1}(x_s) = x_s^{max} \exp(-x_s \alpha_s/(M_s \tau_s))$. Suppose $d_l(y_l) = \frac{a}{c_l - y_l}$, where a is a constant, c_l is the capacity link of link l. We have $d_l^{'-1}(z) = c_l - \sqrt{\frac{a}{z}}$.

Set the parameters as $a = 1$, $b_s = 2$, $c_l = 200$, and $x_s^{max} = c_1$, for each user and each link. Let $\kappa_l = 1/c_l, \forall l$. Note that we relax the condition $\kappa_l = 1/y_l^*$ in Theorem 3 for the sake of implementation, as discussed in Remark 3.

Figures 6 and 7 are for fixed $\alpha = \pi/2 - 0.1$ (i.e., $\alpha < \pi/2$), but different delays τ_s. For Fig. 6, the delays are $\tau_{11}^f = 5, \tau_{11}^b = 5, \tau_{12}^f = 8, \tau_{12}^b = 2, \tau_{22}^f = 10, \tau_{22}^b = 10, \tau_{23}^f = 14, \tau_{23}^b = 6, \tau_{33}^f = 5, \tau_{33}^b = 5, \tau_{44}^f = 6, \tau_{44}^b = 4, \tau_{54}^f = 8, \tau_{54}^b = 12, \tau_{52}^f = 10, \tau_{52}^b = 10, \tau_{55}^f = 12, \tau_{55}^b = 8$, and for Fig. 7, the delays are doubled. It can be easily checked that the delay-dependent sufficient condition is satisfied. The figures show that the system is stable, and the system arrives at the equilibrium, which has been verified to be the optimum resource allocation. Figure 7 is with larger delay than Fig. 6. Comparing these two figures, it can be seen that as the delay τ becomes larger, the system gets to the equilibrium much more slowly as expected.

When α becomes larger, the system equilibrium becomes unstable. This can be seen if we compare Fig. 6 and Fig. 8, which have the same delays τ, but different α.

6.2.2 *Delay-Independent Condition for Stability*

Now we turn our attention to the delay-independent conditions. We let $f_s(x) = \frac{M_s}{\alpha_s} x_s(1 - \log(x_s/x_s^{max})), \forall s$ and $d_l(y_l) = a/(c_l - y_l), \forall l$. We set the parameters as $a = 1$, $b_s = 60000$, $\alpha_s = 0.99$, $c_l = 80$, and $x_s^{max} = c_1$, $\forall s$, $\forall l$. Let $\kappa_l = 1/y_l^*, \forall l$. It can be checked that the delay-independent sufficient condition of Theorem 4 holds.

Figures 9 and 10 show the rates of the five users $x_s(t)$. Figure 9 is for smaller delays τ_s, and Fig. 10 is for larger delays τ_s. It can be seen that no matter how large the delays are, the system is stable, and it converges to the optimal resource allocation. But the performance of the

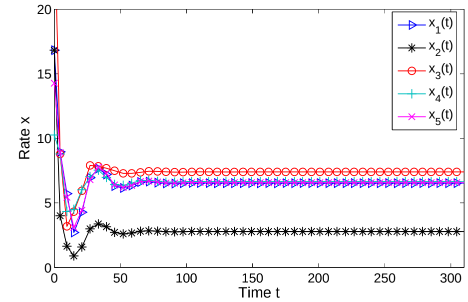

Figure 9: Stable (under delay-independent condition): $(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5) = (10, 20, 10, 10, 20)$ **,** $\alpha_s = 0.99$**.**

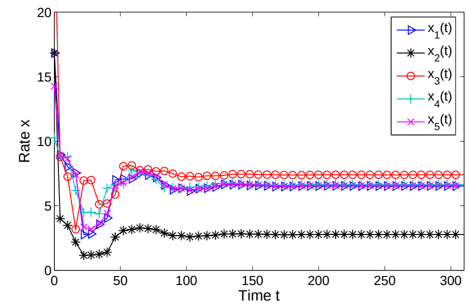

Figure 10: Stable (under delay-independent condition): $(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5) = (20, 40, 20, 20, 40)$**,** $\alpha_s = 0.99$**.**

system becomes worse as the delay is increased and is not as good as the one for the delay-dependent conditions.

6.2.3 Discussion

The conditions given in Theorem 3 and Theorem 4, are similar, but produce very different results.

Let us consider a scenario in which the condition $\kappa_l > B_l d_l''(y_l^*), \forall l$ holds, which is part of the delay-independent conditions as in Theorem 4. In this case, the delay-dependent conditions in Theorem 3 can be improved to $\alpha_s/\tau_s < 1$ (no limitation on α_s) instead of the original one $\alpha_s < \pi/2$, which means that stability is retained for even bigger gains α_s. If, however, the condition $\kappa_l > B_l d_l''(y_l^*)$ breaks and $\alpha_s > \pi/2$, then the system will become unstable.

7. CONCLUSIONS

Different applications (delay sensitive/insensitive) have different QoS requirements and hence users have utilities that are functions of both rate and delay. In this paper we considered a new network utility maximization problem with such utility functions, and we solved it in a decentralized way. The resulting link price dynamics are now functions of the delay, reflecting how rate and delay affect user's satisfaction on the QoS.

The results were applied to the system with voice (delay sensitive) and data traffic (delay sensitive/insensitive), and a distributed algorithm was proposed, that allocates the resource such that the utility of voice and data is jointly optimized. In particular, in a network with data and voice traffic with priority queueing, the algorithm can result in higher R-factor of voice traffic and higher throughput of data traffic, sacrificing packet delay of data traffic.

For the dual algorithm, in the presence of heterogeneous propagation delays, delay-independent and delay-dependent stability conditions were derived. This is the first dual algorithm that allows for delay-independent stability. The conditions in Theorems 3 and 4 involve knowledge of y_l^* at the links. This may not be possible, but as remarked in Remark 3, the two conditions can be relaxed to $\kappa_l = \frac{1}{c_l}$, where c_l is the capacity of the link. This relaxation would make the two conditions conservative, but is more easily implementable.

Future work will concentrate on developing a distributed algorithm for the case in which packet loss due to the delay exceeding a fixed deadline is included. Also, the choice of the delay-free utility function f_i at the users is restricted in Theorems 3 and 4, as $f_i''(x_i^*) = \frac{\beta}{x_i^*}$ for some $\beta < 0$ is required, which implies a particular shape of utility function. Future research will concentrate on primal-dual algorithms that would allow the sources to adapt to a Utility function of their choice at a much slower time-scale, like the one in [15].

8. REFERENCES

[1] *ITU-T Recommendation G.107: The Emodel, a computational model for use in transmission planning*, 1998.

[2] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[3] D. Bertsekas and R. Gallager. *Data networks*. Prentice Hall, 1992.

[4] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of IEEE*, 95(1):255–312, 2007.

[5] J. K. Hale and S. M. V. Lunel. *Introduction to Functional Differential Equations*. Applied Mathematical Sciences (99). Springer-Verlag, 1993.

[6] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[7] L. Kleinrock. *Queueing Systems, Vol.1: Theory.* Wiley, 1976.

[8] Y. Kuang. *Delay Differential Equations with Applications in Population Dynamics*. Academic Press, 1993. Mathematics in Science and Eng. (191).

[9] Y. Li, M. Chiang, and A. R. Calderbank. Congestion control in networks with delay sensitive traffic. To appear in Proceedings of the IEEE Globecom, 2007.

[10] Y. Li, M. Chiang, R. A. Calderbank, and S. N. Diggavi. Optimal rate-reliability-delay tradeoff in networks with composite links. In *Proceedings of IEEE INFOCOM*, May 2007.

[11] S. H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions on Networking*, 11(4):525–536, 2003.

[12] S. H. Low and D. E. Lapsley. Optimization flow control, I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 1999.

[13] S.-I. Niculescu. *Delay Effects on Stability: A Robust Control Approach*. Lecture Notes in Control and Information Sciences (269). Springer-Verlag, 2001.

[14] F. Paganini, J. Doyle, and S. Low. Scalable laws for stable network congestion control. In *Proceedings of the 40th IEEE Conf. on Decision and Control, Orlando, FL*, 2001.

[15] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low. Congestion control for high performance, stability and fairness in general networks. *IEEE/ACM Transactions on Networking*, 13:43–56, 2005.

[16] A. Papachristodoulou. Analysis of nonlinear time delay systems using the sum of squares decomposition. In *Proc. of the American Control Conference*, 2004.

[17] A. Papachristodoulou, L. Li, and J. C. Doyle. Methodological frameworks for large-scale network analysis and design. *ACM SIGCOMM Computer Communications Review*, (3):7–20, 2004.

[18] J. Pongsajapan and S. H. Low. Reverse engineering TCP/IP-like networks using delay-sensitive utility functions. In *Proc. of IEEE INFOCOM*, May 2007.

[19] J. S. Stidham. Pricing and congestion management in a network with heterogeneous users. *IEEE Trans. on Auto. Control*, 49(6):976–981, 2004.

[20] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhäuser, 2004.

[21] G. Vinnicombe. On the stability of end-to-end congestion control for the Internet. Technical report, Cambridge University, CUED/F-INFENG/TR.398, December 2000.

[22] L. Ying, G. Dullerud, and R. Srikant. Global stability of Internet congestion controllers with heterogeneous delays. *IEEE/ACM Transactions on Networking*, 14(3):579–591, 2006.