Rate Control for Delay-Sensitive Traffic in Multihop Wireless Networks

Soroush Jahromizadeh and Veselin Rakocevic
Information Engineering Research Centre
City University, London
UK
+44 (0)20 70408136
{s.jahromizadeh, v.rakocevic}@city.ac.uk

ABSTRACT

We propose two multipath rate control algorithms that guarantee bounded end-to-end delay in multihop wireless networks. Our work extends the previous research on optimal rate control and scheduling in multihop wireless networks, to support inelastic delay requirements. Using the relationship between dual variables and packet delay, we develop two alternative solutions that are independent from any queuing model assumption, contrary to the previous research. In the first solution, we derive lower bounds on source rates that achieve the required delay bounds. We then develop a distributed algorithm comprising scheduling and rate control functions, which requires each source to primarily check the feasibility of its QoS before initiating its session. In the second solution we eliminate the admission control phase by developing an algorithm that converges to the utility function weights that ensure the required delay bounds for all flows. Both solutions carry out scheduling at slower timescale than rate control, and consequently are more efficient than previous cross-layer algorithms. We show through numerical examples that even when there are no delay constraints, the proposed algorithms significantly reduce the delay compared to the previous solutions.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network architecture and design – *wireless networks.*

G.1.6 [**Numerical Analysis**]: Optimization – *constrained optimization, convex programming.*

General Terms

Algorithms, Performance.

Keywords

Cross-layer optimization, QoS, Multihop wireless networks, Delay, Rate control.

PM2HW2N'09, October 26, 2009, Tenerife, Canary Islands, Spain.

1. INTRODUCTION

Multihop wireless networks are becoming an important part of the future communication systems because of their flexibility and self-organizing features. Communication between nodes in these networks is performed over multiple hops and paths, resulting in an improved network capacity and coverage. As such, these networks can be potentially used for real-time voice and video communications in areas with no access to communication infrastructure. However supporting such delay-sensitive applications pose challenging problems as they typically have stringent QoS requirements for bandwidth and delay. Higher bit rates allow audio or video streams to be encoded at higher qualities. Moreover, packets arriving later than their playout deadline are discarded, resulting in severe quality degradation as decoding errors propagate to the subsequent packets.

In this paper we are concerned with the design of multipath rate control strategies for supporting delay-sensitive traffic in multihop wireless networks. We assume that the network uses multipath routing where each source can send its data over multiple paths to their destination. Furthermore, we assume that the delay-sensitive applications require a bounded end-to-end packet delay, but are flexible in their bandwidth requirement. Specifically, the signal quality is a strictly concave function of bandwidth. Utilizing the elasticity of these applications, we design multipath rate control strategies that ensure bounded queuing delay and fair allocation of resources. As network layers and flows are tightly coupled due to the shared nature of wireless medium, we adopt a cross-layer optimization approach similar to [1], which also incorporates the resource allocation decisions at the underlying layers (which are referred to as scheduling in the literature). Such cross-layer optimization approach also enables us to design a more efficient layering of the protocol stack, as discussed in [2]and [3].

Previous work on multipath routing has largely focused on maximizing the probability of successful reception of data. Examples include [4] where a multipath scheme is proposed based on diversity coding, which chooses the optimal data allocation to minimize packet drop rate and improve end-to-end delay. Moreover, routing protocols for ad hoc networks which support multipath routing such as [5], use redundant paths as means of increasing network robustness. The problem of joint rate control, routing and scheduling in multihop wireless networks has been extensively studied (see [2] and [3] for a comprehensive survey), however, the main focus has been on the scheduling problem (see e.g. [1] and [6]) and the delay requirements have not yet been included in the model. Explicit modeling of QoS requirements in the optimization framework has been shown to be intractable in many cases (see e.g. [7] and references therein). In [7], a convex programming formulation that captures average and probabilistic delay requirements for wired networks is presented. The optimization model is nevertheless based on M/G/1 queue approximation of link dynamics, and the proposed algorithm for solving the optimization problem is partly centralized.

We propose an alternative approach for guaranteeing bounded end-to-end delay in multihop wireless networks. We use the same optimization framework as in [1] but, instead of explicit formulation of the delay constraints, we use the relationship between dual variables and packet delay to develop two alternative solutions that are independent from any queuing model assumption. In the first solution, we derive lower bounds on source rates that achieve the required delay bounds. The lower bounds on source rates can be interpreted as the inflection points of the rate-adaptive applications utility functions described in [9]. With the additional rate constraints, however, the optimization problem may become infeasible. The feasibility of a required bandwidth depends on the required delay bounds as well as utility function weights. The proposed algorithm therefore includes an admission control phase in which each source checks the feasibility of its required QoS given the current network priorities.

In the second solution we eliminate the admission control phase by developing an algorithm that converges to the utility function weights that ensure the required delay bounds for all flows. In other words, the proposed algorithm adjusts the QoS or utility gained by a particular bandwidth, on the basis of the impact of the optimal bandwidth on the end-to-end delay. This way, the algorithm also defines a new measure of fairness for allocating the available bandwidth among sources, based on their delay requirements. Both algorithms allow distributed implementation over rate control and scheduling layers. Moreover, at equilibrium they achieve equal end-to-end delay on active paths, in contrast to the dual-based algorithms used in [1] and [8] for solving multipath network utility maximization problems, which result in equal number of packets on active paths.

The rest of this paper is organized as follows. The system model and the problem definition are presented in Section 2. Our solutions are developed in Section 3. In Section 4 we compare the performance of our proposed algorithms with the dual based algorithm that solves the same optimization problem without the delay constraints. We conclude the paper in Section 5.

2. SYSTEM MODEL

We consider a multihop wireless network with L links and S sources of delay-sensitive traffic. Let $\overline{L}=\{1,\ldots,L\}$ and $\overline{S}=\{1,\ldots,S\}$ denote the set of links and sources, respectively. Let $\overline{N}_s=\{1,\ldots,N_s\}$ denote the set of available paths for source $s\in\overline{S}$. The set of links used by source s's paths are given by $L\times N_s$ matrix R^s with elements

$$R^s_{li}=\begin{cases}1, & \text{if path i} \in \overline{\text{N}}_\text{s} \text{ uses link l} \in \overline{\text{L}}\\ 0, & \text{otherwise}\end{cases}.$$

Let $N=\sum_{s\in\overline{S}}N_s$ be the total number of paths. The $L\times N$ routing matrix R is defined by $R=[R^1\ldots R^S]$. Each source s transmits at the rate of x^s_i over path $i\in N_s$, and at the total rate of $\hat{x}^s=\sum_{i\in\overline{N}_s}x^s_i$. Let $x^s=\left(x^s_1,\ldots,x^s_{N_s}\right)$ and $x=\left(x^1,\ldots,x^S\right)$ be the vectors of path rates for source s and path rates for all sources, respectively.

Let $c=\left(c_1,\ldots,c_L\right)$ denote the vector of link transmission rates. Because of the shared nature of the wireless medium, link rates depend on the scheduling policy used. A scheduling policy may incorporate power control in which case link rates are a function of global power assignments. If power control is not allowed, then a set of non-conflicting links are scheduled to transmit at their fixed rate at each time slot. In either case it is assumed that at different timeslots different sets of links are scheduled to achieve maximum capacity. Specifically, the feasible link rate region is defined as

$$c\in\mathrm{Co}\left(\overline{C}\right) \tag{1}$$

Where $\overline{C}$ is the set of feasible link rates and $\mathrm{Co}\left(\overline{C}\right)$ is the convex hull of $\overline{C}$ and assumed to be compact. Moreover, total flow rates on links cannot exceed their achievable rates, i.e.

$$Rx\le c \tag{2}$$

Each source s is assumed to have a maximum playout delay denoted by d_s. Packets arriving later than the playout delay will be lost, resulting in severe quality degradation. The end-to-end delay requirement is therefore formulated as an inelastic QoS constraint. In particular, let $T=\left(T_1,\ldots,T_L\right)$ denote the vector of packet delay at each link. The packet end-to-end delay on each path must not be more than the required playout delay, i.e.

$$R^{sT}T\le d_s,\ s\in\overline{S} \tag{3}$$

The bandwidth allocated to each source is considered as an elastic QoS requirement. Thus, each source s obtains a utility $w_s f_s\left(\hat{x}^s\right)$, $w_s>0$ when it transmits at total rate of $\hat{x}^s$ packets per second. The functions $f_s\left(\hat{x}^s\right)$ are assumed to be continuously differentiable, increasing and strictly concave.

The cross-layer multipath rate control problem is to find path rates x, and link rates c, such that

$$\underset{x,c\ge 0}{\mathrm{Max}}\sum_{s\in\overline{S}}w_s f_s\left(\hat{x}^s\right)\ \text{Subject to (1)-(3)} \tag{4}$$

3. DISTRIBUTED ALGORITHMS

3.1 Case with no Bounded Delay Requirements

To develop a distributed algorithm for the optimization problem (4), we first consider the solution to the problem without the end-to-end delay constraints (3), which has also been addressed in detail in [1].

The optimization problem (4) without constraint (3) is a convex problem and can be conveniently solved using dual methods. For now, we ignore the lack of strict concavity of the objective function in (4) with respect to x and c, and the problems it causes in recovering primal optimal solutions, when dual method is used. This issue will be addressed when the algorithm for the original problem is developed.

The partial dual problem for (4) without constraint (3) is defined as

$$\operatorname*{Min}_{\lambda \geq 0} \; g(\lambda) \tag{5}$$

where

$$g(\lambda) = \operatorname*{Max}_{x, c \geq 0} \sum_{s \in \overline{S}} w_s f_s\left(\hat{x}^s\right) - \lambda^T \left(Rx - c\right) \text{ Subject to (1)}$$

where $\lambda = \left(\lambda_1, \ldots, \lambda_L\right)$ denotes the vector of Lagrange multipliers, which can be interpreted as link implicit costs. Problem (5) can be decomposed into two subproblems $g(\lambda) = g_1(\lambda) + g_2(\lambda)$ where

$$g_1(\lambda) = \operatorname*{Max}_{x \geq 0} \sum_{s \in \overline{S}} \left(w_s f_s\left(\hat{x}^s\right) - \lambda^T R^s x^s\right) \tag{6}$$

and

$$g_2(\lambda) = \operatorname*{Max}_{c \geq 0} \lambda^T c \text{ Subject to (1)} \tag{7}$$

Subproblems (6) and (7) correspond to the rate control and scheduling problems, respectively. The two problems are coupled via link implicit costs λ.

Consequently, the master dual problem (5) can be solved using the projected gradient method as follows

$$\lambda_l^{(k+1)} = \left(\lambda_l^{(k)} + \beta\left(\sum_{s \in \overline{S}} \sum_{i \in \overline{N}_s} R_{li}^s x_i^{s(k)} - c_l^{(k)}\right)\right)^+, \; l \in \overline{L} \tag{8}$$

where $x_i^{s(k)} = x_i^{s*}\left(\lambda^{(k)}\right)$ and $c_l^{(k)} = c_l^*\left(\lambda_l^{(k)}\right)$ are the solutions of (6) and (7), respectively. Algorithms (6) and (8) can be implemented in a fully distributed fashion. Specifically, in (8), each link implicit price is updated using the current link transmission rate, and aggregate flow that passes over the link. Each source updates its rate by solving (6), given its utility function and its current paths implicit costs. The link transmission rates are updated by solving (7). The scheduling problem (7) is generally a complex problem, however, for some simple interference models efficient distributed algorithms have been developed [2].

3.2 Case with Bounded Delay Requirements

Algorithm (8) shows that the link implicit cost and the queue length are closely related. Let $\eta_l^{(k)}$ denote the queue length at step k, then

$$\eta_l^{(k)} = \lambda_l^{(k)} \big/ \beta, \; l \in \overline{L} \tag{9}$$

If the link rates were fixed at each iteration of (8), i.e. $c_l^{(k)} = c_l^0$, $l \in \overline{L}$, modifying the step size in (8) to $\beta = \hat{\beta} \big/ c_l^0$, relates the link implicit cost directly to the packet delay instead

$$T_l^{(k)} = \lambda_l^{(k)} \big/ \hat{\beta}, \; l \in \overline{L} \tag{10}$$

Consider the partial Lagrangian for the original problem (4)

$$L_1(x, \lambda, \omega) = \sum_{s \in \overline{S}} w_s f_s\left(\hat{x}^s\right) + \omega^T x - \lambda^T \left(Rx - c\right)$$

Where $\omega = \left(\omega^1, \ldots, \omega^s\right)$ and $\omega^s = \left(\omega_1^s, \ldots, \omega_{N_s}^s\right)$. The Karush-Kuhn-Tucker (KKT) optimality conditions for the original problem (4) with respect to x is then given by

$$w_s f_s'\left(\hat{x}^{s*}\right) - \sum_{l \in \overline{L}} \lambda_l^* R_{li}^s + \omega_i^{s*} = 0, s \in \overline{S}, i \in N_s \tag{11}$$

$$x_i^{s*} \omega_i^{s*} = 0, \; s \in \overline{S}, \; i \in N_s \tag{12}$$

From (12) ω_i^{s*} must be zero for paths with positive flow; paths with no flow can be ignored as they clearly have zero delay. Thus, replacing (10) in (11) results in the following relationship between the end-to-end delay on each path, and the source's utility at its optimal rate

$$\sum_{l \in \overline{L}} T_l^* R_{li}^s = w_s f_s'\left(\hat{x}^{s*}\right) \big/ \hat{\beta}, \; s \in \overline{S}, i \in N_s . \tag{13}$$

Equation (13) also implies that for each source, paths with positive flow have the same end-to-end delay. Based on the above result, we propose two alternative solutions for the optimization problem (4):

3.2.1 Solution 1: Replacing delay bounds (3) with minimum acceptable source rates

From (13) we conclude that the following inequalities

$$w_s f_s'\left(\hat{x}^{s*}\right) \big/ \hat{\beta} \leq d_s, \; s \in \overline{S} \tag{14}$$

guarantee the delay bounds (3) at optimality conditions. Assuming that the sources' utility functions can be approximated by a logarithmic function through an appropriate choice of weights, i.e. $w_s f_s\left(\hat{x}^s\right) = w_s \log\left(\hat{x}^s\right)$, we obtain the following lower bound on source rates

$$w_s \big/ \hat{\beta} d_s \leq \hat{x}^s, \; s \in \overline{S} \tag{15}$$

Rewriting the KKT conditions for the optimization problem (4) with (3) now replaced by (15)

$$w_s f_s'\left(\hat{x}^{s*}\right) - \sum_{l \in \overline{L}} \lambda_l^* R_{li}^s + \omega_i^{s*} + \tau_s^* = 0, s \in \overline{S}, i \in N_s \tag{16}$$

$$\tau_s^* \left(w_s \big/ \hat{\beta} d_s - \hat{x}_i^{s*}\right) = 0, \; s \in \overline{S} \tag{17}$$

where $\tau = (\tau_1, \ldots, \tau_s)$ is the vector of Lagrange multipliers associated with (15). In order for (13) to hold in this case, τ must be zero. From (17), a sufficient condition for $\tau = 0$ is that $\hat{x}^{s*}$ to be strictly feasible with respect to the constraints (15). To conclude, the solution of the following optimization problem guarantees the end-to-end delay bounds (3), providing it is strictly feasible with respect to the constraint (15)

$$\underset{x,c\geq 0}{\text{Max}} \sum_{s\in \overline{S}} w_s f_s\left(\hat{x}^s\right) \text{ Subject to (1), (2) and (15)} \qquad (18)$$

The idea of modeling the utility of delay-sensitive applications as a strictly concave function subject to the minimum acceptable source rates is also consistent with the performance characteristics of rate-adaptive real-time applications described in [9]. The utility of rate-adaptive applications is strictly concave at rates greater than the bandwidth associated with the minimally acceptable signal quality. At rates smaller than this point the utility function becomes convex and the network can be overloaded. The lower bound in (15) can therefore be mapped to the inflection point of the rate-adaptive applications utility function, since it is the bandwidth below which the signal quality degrades severely as a result of end-to-end delay violations.

Because of the rate bounds (15), the optimization problem (18) may not always have a feasible solution. Therefore in this solution the network requires an admission control mechanism, just like the networks supporting rate-adaptive applications [9]. The feasibility of (18) can be established by solving the following optimization problem [10]

$$\underset{x,t,c\geq 0}{\text{Min}} \; 1^T t \qquad (19)$$

Subject to (1), (2) and

$$w_s \big/ \hat{\beta} d_s - \hat{x}^s \leq t_s, \; s \in \overline{S} \qquad (20)$$

where $t = (t_1, \ldots, t_s)$. The problem (18) has a feasible solution when $t^* = 0$. If there exists a source $s \in \overline{S}$ for which $t_s^* > 0$, then the corresponding constraint in (15) is infeasible, i.e. its minimum required bandwidth cannot be achieved by the optimal bandwidth allocation policy, which is based on the utility function weights. The feasibility of a required bandwidth therefore depends on the choice of utility function weights. We will exploit this property in our second approach to eliminate the admission control phase. Non-zero dual variables associated with a feasible source (i.e. $s \in \overline{S}$ for which $t_s^* = 0$) will further identify constraints in (15) that are not strictly feasible. The feasibility problem (19) can be solved in the same way as problem (4) without constraint (3) described in section 3.1. We summarize the proposed admission control policy as follows

Algorithm A1: Admission control - Performed by each source

1. Solve the feasibility problem (19) jointly with other sources
2. Do not initiate session if $t_s^* > 0$, or the associated dual variable is not zero, or the associated constraint in (20) is not strictly feasible

To develop a distributed algorithm for solving (18) the dual decomposition method used in section 3.1cannot be applied, since link rates $c_l^{(k)}$ vary at each iteration of gradient algorithm (8) and the relation (10) does not hold, as explained previously. We instead use primal decomposition to transform (18) into a scheduling master problem

$$\underset{c\geq 0}{\text{Max}} \; \Phi(c) \text{ Subject to (1)} \qquad (21)$$

where $\Phi(c)$ is the optimal utility of the following rate control subproblem for a given *c*

$$\underset{x,c\geq 0}{\text{Max}} \sum_{s\in \overline{S}} w_s f_s\left(\hat{x}^s\right) \text{ Subject to (2) and (15)} \qquad (22)$$

It is easy to show that $\Phi(c)$ is a concave function of *c* and $\lambda^*(c)$ is one of its subgradients, where $\lambda^*(c)$ is the optimal Lagrange multiplier corresponding to constraint (2) [10]. The decomposition of (18) as above allows rate control problem (22) to be solved for a fixed link rate vector *c*, thus eliminating the problem of variation of *c* in the dual decomposition approach. Moreover, in this approach scheduling is performed at a slower timescale than rate control, which is highly desirable due to high computational complexity of the scheduling problem [2]. This is in contrast to the dual decomposition approach used in section 3.1, in which the scheduling problem has to be solved for every link price update.

The rate control problem (22) can be solved using the dual approach. However since the objective function in (22) is not strictly concave with respect to *x*, primal optimal solutions cannot be easily computed from the dual optimal solutions. Moreover, as discussed in [8], primal variables will not converge when dual approaches are used to solve this problem. Hence, the following equivalent problem is solved instead [11]

$$\underset{x,y\geq 0}{\text{Max}} \sum_{s\in \overline{S}} w_s f_s\left(\hat{x}^s\right) - \frac{1}{2\overline{c}} \left\| x - y \right\|_2^2 \qquad (23)$$

Subject to (2) and (15).

The problem (23) is strictly concave with respect to *x* for fixed *y*, and strictly concave with respect to *y* for fixed *x*. The partial dual problem for (23), when *y* is fixed, is

$$\underset{\lambda\geq 0}{\text{Min}} \; g(\lambda) \qquad (24)$$

where

$$g(\lambda) = \underset{x\geq 0}{\text{Max}} \sum_{s\in \overline{S}} w_s f_s\left(\hat{x}^s\right) - \frac{1}{2\overline{c}} \left\| x - y \right\|_2^2 - \lambda^T \left(Rx - c\right) \quad (25)$$

Subject to (15)

The primal problem (23) can be solved using the Proximal Optimization algorithm [11], but here we use the algorithm proposed in [8] as it is more suitable for distributed implementation. The algorithm for solving (22) is then summarized as follows

Algorithm A2: Mutipath rate control- Performed jointly by each source and links that belong to its paths

1. Set

$$\lambda^{(t'+1,0,0)} = \lambda^{(t')} c^{(t')} / c^{(t'+1)}, \ \forall l : c_l^{(t'+1)} > 0 \qquad (26)$$

where $\lambda^{(t')} = \lambda^{*}\left(c^{(t')}\right)$ is the optimal value of λ given $c^{(t')}$.

2. *Solving (22) using algorithm proposed by [8]:* At step $t+1$

   a. Fix $y = y(t)$

   b. *Solving (23):* At step $k+1$ (repeat for $K > 0$ times), given $\lambda_l^{(t'+1,t,k)}$:

$$\lambda_l^{(t'+1,t,k+1)} = \left(\lambda_l^{(t'+1,t,k)} + \frac{\hat{\beta}}{c_l^{(t'+1)}} \left(\sum_{s \in \overline{S}} \sum_{i \in \overline{N}_s} R_{li}^{s} x_i^{s(t,k)} - c_l^{(t'+1)} \right) \right)^{+} \qquad (27)$$

$\forall l : c_l^{(t'+1)} > 0$

where $x^{(t,k)}$ is the solution of (25) with $x_i^{s} = 0$ if $c_l^{(t'+1)} = 0$ for any $l \in \overline{L} : R_{li}^{s} = 1$.

   c. Set $\lambda^{(t'+1,t+1,0)} = \lambda^{(t'+1,t,k)}$

   d. Set $y(t+1) = y(t) + \alpha\left(z(t) - y(t)\right)$, where $z(t)$ solves (25) for $\lambda = \lambda^{(t'+1,t+1,0)}$ and $0 < \alpha_i^{s} \le 1, \ s \in \overline{S}, i \in N_s$.

As explained previously, the step size in (27) ensures that packet delay at each link is related to the dual variable (link implicit cost) through equation (10). Since the link rates are updated at each iteration of A2, dual variables are initialized according to (26), in order for (10) to hold at the start of A2. Similar to algorithms (6) and (8) in section 3.1, A2 computation is carried out by sources and links using their local information.

Computation of optimal schedule in (21) is less straight forward, as the feasible rate region (1) is generally difficult to characterize (see e.g. [6]), and no description of the objective function is available other than its value and its subgradient at query points. Cutting-plane methods [12] appear to be a suitable technique for this case, which also enables distributed implementation of the solution, as we next demonstrate for the case where scheduling does not incorporate power control.

3.2.1.1 Solving scheduling problem (21)

We consider the case where scheduling does not incorporate power control. Our formulation of the feasible rate region (1) is based on the model presented in [6], as it allows a distributed implementation of scheduling. In this approach, the notion of flow contention graph and contention matrix is used to model interference relations among links. In a flow contention graph, each vertex represents an active link and an edge between two vertices represents contention between the corresponding links, i.e. two links interfere with each other and cannot be active simultaneously. Maximal cliques in the contention graph capture local contention relation of the links and can be viewed as "channel resources". Links within a maximal clique mutually interfere with each other and share the capacity of the clique. If a link belongs to several cliques, it can be active if and only if it is the only active link in all cliques it belongs to. Let $\overline{K} = \{1, \dots, K\}$ denote the set of maximal cliques in the contention graph. The *K*×*L* contention matrix *F* is defined by

$$F_{nl} = \begin{cases} 1/c_l^0 & \text{if link l belongs to clique n} \in \overline{\mathrm{K}} \\ 0 & \text{otherwise} \end{cases}$$

where c_l^0 denotes the transmission rate of link *l*, when active. Since links in a maximal clique share the capacity of the clique, the feasible rate region can be written as

$$Fc \le \varepsilon, \ 0 \le c \le c_l^0 \qquad (28)$$

where $\varepsilon \le 1$. The value of ε depends on the local topology of the contention graph (e.g. $\varepsilon = 1$ for perfect graphs), and is difficult to determine in general. However, if $c = 0$ or c_l^0, i.e. *c* corresponds to the independent sets of the contention graph, then $\varepsilon = 1$.

Throughout the rest of this paper, we consider the scheduling problem (21) with feasible rate region characterized as (28). We will show later that in our proposed cutting plane algorithm it is not necessary to know the value of *ε*. Starting from the upper bound value *ε* = 1, the proposed algorithm checks for the feasibility of the query point using a simple heuristic and improves its estimation for *ε* as it converges to the optimal solutions. The proposed algorithm for solving the scheduling problem (21), and consequently the optimization problem (18), then proceeds as follows

Algorithm A3

1. *Admission control:* Run algorithm A1
2. *Solving scheduling problem (21) using cutting-plane method:* Given initial polyhedron $\mathrm{P}_0 = \left\{ c \,\middle|\, Fc \le 1, \ 0 \le c \le c_l^0, l \in \overline{L} \right\}$, at step $t'+1$:

   a. Choose a point $c(t'+1)$ in $\mathrm{P}_{t'} = \mathrm{P}_{t'}^{o} \cap \mathrm{P}_{t'}^{f}$, where $\mathrm{P}_{t'}^{o}$ and $\mathrm{P}_{t'}^{f}$ denote the polyhedron of objective cuts and feasibility cuts, respectively.

   b. Check if $c(t'+1)$ is feasible using algorithm A4. If $c(t'+1)$ is feasible, go to step c. Else, update $\mathrm{P}_{t'}^{f}$ by replacing previous feasibility cuts with new ones, using estimated $\varepsilon(t'+1)$ returned by A4:

$\mathrm{P}^{f}_{t'+1} = \left\{ c \middle| Fc \le \varepsilon\left(t'+1\right),\ 0 \le c \le c^{0}_{l}, l \in \overline{L} \right\}$, and return to step a.

c. Solve the rate control problem (22) given $c\left(t'+1\right)$, using algorithm A2. If $\lambda^{*}\left(c\left(t'+1\right)\right) = \lambda^{\left(t'+1\right)} = 0$, quit ($c\left(t'+1\right)$ is optimal). Else, update $\mathrm{P}^{o}_{t'}$ by adding new objective cut: $\mathrm{P}^{o}_{t'+1} = \mathrm{P}^{o}_{t'} \cap \left\{ c \middle| \lambda^{(t'+1)T} c \le \lambda^{(t'+1)T} c\left(t'+1\right) \right\}$

Step b in algorithm A3 uses algorithm A4 to establish the feasibility of the query point $c\left(t'+1\right)$. A4 uses a simple heuristic based on the distributed scheduling algorithm proposed in [6], where the feasible rate region in the scheduling subproblem (7) is defined as

$$Fc \le 1,\ c = 0 \text{ or } c^{0}_{l} \qquad (29)$$

Note that since c is binary, it corresponds to the independent sets of the contention graph and thus $\varepsilon = 1$. Using dual decomposition, suboptimal schedule for (7) is then obtained by distributed computation over cliques of the contention graph, primarily ignoring the discrete constraint on c and then rounding up the value of c to c^{0}_{l} or 0, whichever is closer (see [6] for details).

Algorithm A4 operates on a time frame consisting of a number of timeslots. During each time slot, a schedule is computed by solving (7) as described above. The link weights in the objective function are then updated based on the difference between the query point value and the average link rate achieved up to the current time slot. If this difference becomes sufficiently small at some point, the query point is assumed feasible. On the other hand, if after a certain number of timeslots this difference is still large, the query point is assumed infeasible and a new estimate for ε is returned based on the attained average link rate. Algorithm A4 is summarized as below

Algorithm A4: Check feasibility of $c\left(t'+1\right)$

Given $\theta_{0} = 1$, at timeslot j

1. Solve linear binary program

   $$\operatorname*{Max}_{c} \theta^{T}_{j} c \ \text{ s.t. (29)} \qquad (30)$$

   using algorithm proposed in [6].

2. Update link weights θ^{j} according to

   $$\theta^{j}_{l} = 1 - \sum_{i=1}^{j} c^{i}_{l} \Big/ j c_{l}\left(t'+1\right) \qquad (31)$$

3. If $\left\|\theta^{j}\right\| \le \delta$, $c\left(t'+1\right)$ is feasible. Else, if j = M, declare $c\left(t'+1\right)$ as infeasible and compute a new estimate for ε

   $$\varepsilon\left(t'+1\right) = 1^{T} F \sum_{i=1}^{j} c^{i} \Big/ j\,K \qquad (32)$$

The maximum number of iterations M is chosen based on the size of the network and the required algorithm efficiency.

Step a in A3 can use standard methods such as maximum volume ellipsoid (MVE) or analytic centre cutting plane methods (ACCPM) which result in speedy convergence of the cutting plane algorithms. These methods, however, involve solving a global optimization problem to find a new query point. It remains the subject of our future research to design efficient decentralized algorithms for choosing query points for this particular problem. The remaining steps of A3 use distributed algorithms as explained previously. Note also that in A3 scheduling is performed at a slower timescale than multipath rate control. Since scheduling has higher computational complexity than rate control, this can potentially result in significantly better performance than algorithms in section 3.1, where both scheduling and rate control run at a same frequency.

3.2.2 Solution 2: Adjusting utility function weights to guarantee the delay bounds (3)

Motivated by the fact that the feasibility of a required bandwidth in (15) (and thus the required delay bound) depends on the choice of utility function weights, in this approach instead of adding lower bound constraints on source rates, the utility function weights are dynamically adjusted based on (14) so that the end-to-end delay is bounded at optimality conditions. In other words, the end-to-end delay constraints (3) are incorporated in the sources' utility functions. The modified utility function therefore reflects the QoS gained by a particular bandwidth taking also into account its impact on the end-to-end delay. As a result, similar to best-effort networks, this approach does not require an admission control phase, as it does not impose any hard constraints on source rates. The proposed algorithm only modifies step c of A3 as described below

Algorithm A5

1. Run steps 1 to 2.b of algorithm A3
2. *Modified step c of A3:*

   a. Update utility weights according to

   $$w^{(t'+1)}_{s} = \mathrm{Min}\left[w^{(t')}_{s}, \hat{\beta} d_{s} \Big/ f'_{s}\left(\hat{x}^{(t')}_{s}\right) \right] \qquad (33)$$

   b. Solve the rate control problem (22) given $c\left(t'+1\right)$ and $w\left(t'+1\right)$, using A2.

   c. If $\lambda^{*}\left(c\left(t'+1\right)\right) = \lambda^{\left(t'+1\right)} = 0$, quit ($c\left(t'+1\right)$ is optimal). Else, if $w^{(t'+1)} = w^{(t')}$, update $\mathrm{P}^{o}_{t'}$ by adding new objective cut: $\mathrm{P}^{o}_{t'+1} = \mathrm{P}^{o}_{t'} \cap \left\{ c \middle| \lambda^{(t'+1)T} c \le \lambda^{(t'+1)T} c\left(t'+1\right) \right\}$, else, update $\mathrm{P}^{o}_{t'}$ by replacing previous objective cuts with new ones: $\mathrm{P}^{o}_{t'+1} = \left\{ c \middle| \lambda^{(t'+1)T} c \le \lambda^{(t'+1)T} c\left(t'+1\right) \right\}$

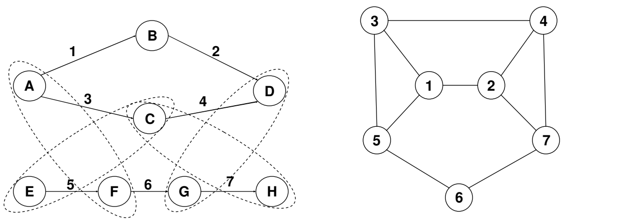

Figure 1: Example network topology (left) and its contention graph (right).

At step c of A5, if utility weights have been modified, the previous objective cuts are discarded, since the objective function has been modified, and replaced with new ones based on the gradient of the modified objective function. The following theorem proves the convergence of the proposed algorithm:

Theorem 1: Algorithm A5 converges to the points $\left(x^*, c^*, w^*\right)$, where $\left(x^*, c^*\right)$ are the stationary points of following optimization problem

$$\underset{x,c\geq 0}{\text{Max}} \sum_{s\in \overline{S}} w_s^* f_s\left(\hat{x}_i^s\right) \text{ Subject to (1)-(2)} \qquad (34)$$

and the utility function weights w^* guarantee the maximum end-to-end delay (3) at optimality condition.

Proof: We only provide an outline of the proof due to space limitation. It is easy to see that the sequence $\left\{w_s\left(t'\right)\right\}$ is non-increasing and bounded from below, hence it converges to the finite and unique limit point w_s^*. Let $\Phi\left(t'\right)$ be the objective function of (21) for $w_s = w_s\left(t'\right)$. It then follows that $\left\{\Phi\left(t'\right)\right\} \to \Phi^*$, after which point algorithm A5 converges to the optimal solution of Φ^*. ■

4. NUMERICAL RESULTS

To evaluate the performance of the proposed algorithms we consider the network in Figure 1. There are two flows: *A-D* and *E-H*. Flow *A-D* has two available paths which contain links (1,2) and (3,4), respectively. Flow *E-H* has only one available path containing links (5,6,7). The interference regions are shown by dashed lines. Although paths do not share any links, they contend for the wireless channel due to interference among some of their links. For example nodes F and C are within the interference range of nodes A and E, respectively, and thus links 3 and 5 cannot be active simultaneously. Moreover, nodes can neither transmit and receive at the same time, nor transmit to more than one node at a time. These link activation constraints are captured in the contention graph in Figure 1. The utility functions for flows *A-D* and *E-H* are $2\log\left(\hat{x}_1\right)$ and $\log\left(\hat{x}_2\right)$, respectively. The links data rate is $c_l^0 = 1$ packets/msec for all links.

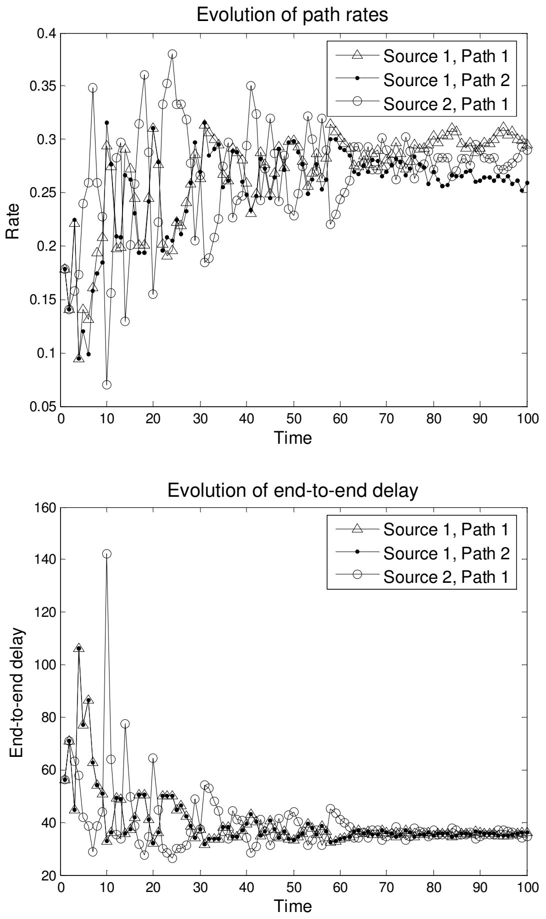

Figure 2. Evolution of path rates and end-to-end delay for A3 without delay control.

We evaluate the proposed algorithms by considering two scenarios. First, we show that even when there are no delay constraints, algorithm A3 significantly reduces end-to-end delay compared to the joint rate control and scheduling algorithm (6)-(8) presented in section 3.1. In the latter algorithm, since the ε value in the feasible rate region (28) is unknown, we use the estimated value for ε computed by A3. In the second scenario, we demonstrate that algorithm A5 can achieve the end-to-end delay bounds that are infeasible for algorithm A3.

Figure 2 shows the evolution of path rates and end-to-end delay, when A3 is run when there are no delay constraints. Aggregate source rates at the end of experiment are $\hat{x}_1 = 0.55$ and $\hat{x}_2 = 0.29$ packets/msec. End-to-end delay on both paths of flow *A-D* is 36, and on flow *E-H* path is 34 msec. The estimated value for ε in (28) is 0.85. The evolution of end-to-end delay for the joint rate control and scheduling algorithm (6)-(8) is shown in Figure 3. ε is set to 0.85 already estimated by A3. It can be seen that end-to-end delay exceeds 850 msec on all paths, which is more than 10 times higher than the delay achieved by A3 without employing delay control.

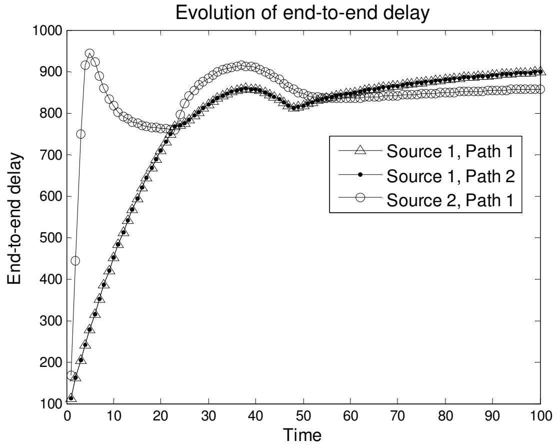

Figure 3. Evolution of end-to-end delay for algorithm (6)-(8).

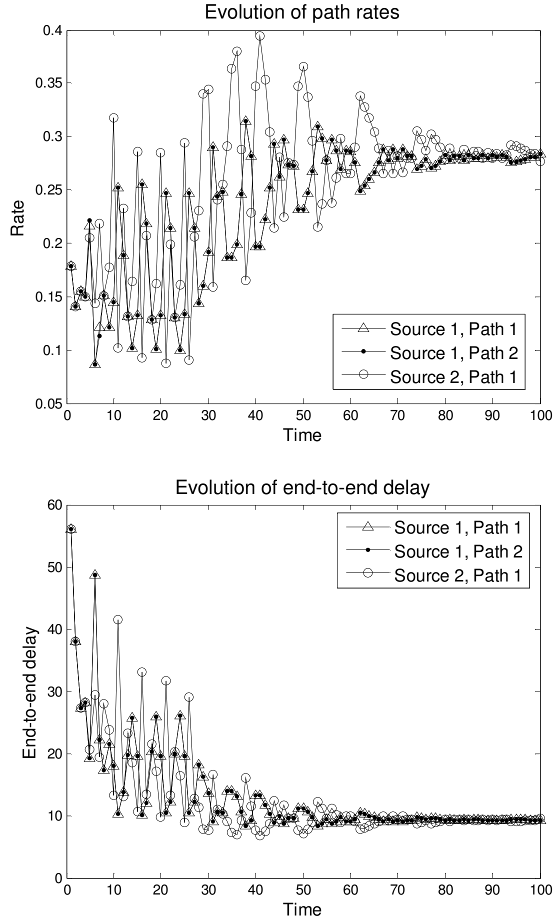

Figure 4. Evolution of path rates and end-to-end delay for A5.

Next, we consider the case where the required end-to-end delay bound on all paths is 30 msec. From (15), this needs minimum source rates of $\hat{x}_1 = 0.67$ and $\hat{x}_2 = 0.33$, which cannot be supported by A3. As Figure 4 shows, algorithm A5 achieves an end-to-end delay of less than 10 msec, by reducing the utility weights of flows *A-D* and *E-H* to 0.52 and 0.26, respectively. Aggregate source rates at equilibrium are $\hat{x}_1 = 0.57$ and $\hat{x}_2 = 0.28$ packets/msec, which are very close to rates attained in the first scenario.

5. CONCLUSION AND FUTURE WORK

In this paper, we develop two distributed rate control algorithms that guarantee bounded end-to-end delay in multihop wireless networks, using an optimization framework. The first algorithm solves an optimization problem with delay bounds replaced by lower bounds on source rates, and includes an admission control phase. The second algorithm instead dynamically adjusts the utility function weights in order to support the required QoS for all flows. Both algorithms are performed over scheduling and rate control levels. Numerical examples show that both algorithms achieve better than expected delay performance; however their efficiency could still be improved. Moreover, fully distributed solutions for the scheduling problem remain an open problem. Addressing theses issues will be the subject of our future research.

6. REFERENCES

[1] Lin, X., and Shroff, N.B. 2006. The Impact of Imperfect Scheduling on Cross-Layer Congestion Control. Wireless Networks. IEEE/ACM Trans. Netw. 14, 2, (2006), 302-315.

[2] Lin, X., and Shroff, N.B. 2006. A Tutorial on Cross-Layer Optimization in Wireless Networks. IEEE J. Sel. Areas Comm. 24, 8, (2006), 1452-1463.

[3] Chiang, M., Low, S., Calderbank, A.R., and Doyle, J.C. 2007. Layering As Optimization Decomposition: A Mathematical Theory of Network Architectures. In Proc. of the IEEE. 95, 1, (2007), 255-312.

[4] Tsirigos, A., and Haas, Z.J. 2004. Analysis of Multipath Routing-Part I: The Effect on The Packet Delivery Ratio. IEEE Trans. Wireless Comm. 3, 1, (2004), 138-146.

[5] Nasipuri, A., and Das, S.R. 1999. On-demand Multipath Routing for Mobile Ad Hoc Networks. In Proceedings of ICCCN'99, 64-70.

[6] Chen, L., Low, S.H., and Doyle, J.C. 2005. Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks. In Proceedings of IEEE INFOCOM 2005, 2212 – 2222.

[7] Saad, M., Leon-Garcia, A., and Yu, W. 2007. Optimal Network Rate Allocation under End-To-End Quality-Of-Service Requirements. IEEE Trans. Netw. and Serv. Management. 4, 3, (2007), 40-49.

[8] Lin, X., and Shroff, N.B. 2006. Utility Maximization for Communication Networks with Multipath Routing. IEEE Trans. Automatic Control. 51, 5, (2006), 766-781.

[9] Shenker, S. 1995. Fundamental Design Issues for the Future Internet. IEEE J. Sel. Areas Comm. 13, 7, (1995),

[10] Boyd, S., and Vandenberghe, L. 1999. Convex Optimization. Cambridge University Press.

[11] Bertsekas, D.P., and Tsitsiklis, J.N. 1989. Parallel and Distributed Computation: Numerical Methods. Prentice-Hall.

[12] Bertsekas, D.P. 1995. Nonlinear Programming. Athena Sci.