Selfish Misbehavior in the Optimal Cross-Layered Rate Control of Wireless Networks

Ghazale Hosseinabadi and Nitin Vaidya
Department of Electrical and Computer Engineering
University of Illinois at Urbana Champaign
{ghossei2,nhv}@illinois.edu

Technical Report
March 31, 2010

***Abstract*—We consider the problem of selfish misbehavior in the optimal cross-layered rate control mechanism of wireless networks. Rate control algorithms of wireless networks are designed under the assumption that network hosts will follow the algorithm specifications. In this paper, we explain a scenario in which a selfish user achieves extra throughput by misleading the scheduling component of the network. We find an equivalent optimization framework that captures misbehavior pattern of the selfish user. We present a solution to prevent or alleviate such a greedy behavior by imposing a cost term on the utility function of the users.**

I. INTRODUCTION

The problem of rate control in wireless networks has been the subject of intensive research in the past few years. The goal of any rate control (or congestion control) scheme is to avoid congestion in the network as well as providing fairness among users of the network. A cross-layered approach to rate control is one where the network jointly optimizes data rates of the users and link schedules. Cross-layered rate control of wireless networks can be mapped to a utility optimization framework. Moreover, the utility optimization framework can be decomposed into two components: rate control at the transport layer and scheduling at the MAC layer [1], [2], [3].

The rate control component controls the rates at which users inject data into the network so as to ensure that they are within the capacity region of the network. The capacity region is the set of all feasible arrival rate vectors. More precisely, the capacity region is defined as the set of all arrival rate vectors for which the queueing system is stable under some scheduling method. In a stable queueing system, queue lengths of links of the network remain bounded over time. Appropriate choice of the rate controller results in achieving some notion of fairness among users of the network [4].

In a wireless network, data collected in queues are transmitted over wireless links. The wireless medium is a shared medium and simultaneous data transmission over conflicting links is not possible. In other words, if two conflicting links are transmitting at the same time, none of them can transfer any useful data. A scheduling policy is a rule to determine a set of links to be activated at each time slot such that the interference constraints of the wireless network are not violated. The main goal of any scheduling algorithm is to stabilize the network, i.e. to determine the schedule in a way that queue lengths of the links do not grow to infinity. Furthermore, achieving good throughput characteristic is another important factor in designing a scheduling algorithm for wireless networks. The MAC-layer scheduling component of the rate control algorithm considers the interference relationship among wireless links and determines a set of non-conflicting links to be activated for data transmission at each time slot. Examples of wireless scheduling algorithms can be found in [6], [5], [7].

Cross-layered rate control schemes of wireless networks are mainly designed under the assumption that all network hosts will follow the algorithm specifications. In this paper, we will present a scenario in which a selfish host misbehaves in order to achieve a better throughput performance. We consider a wireless network in which the link capacities change over time and users of the network are involved in the process of measuring and estimating the link capacities. A selfish user misbehaves in this process and misleads the scheduler. As we will show, the primary goal of a selfish user is to improve its own performance, but its greedy misbehavior usually results in performance degradation of honest hosts.

Current literature addressing selfish misbehavior in wireless networks can be divided into two categories [8]:

1) Solutions that detect misbehaving hosts.
2) Solutions that modify the underlying algorithm or protocol in order to discourage selfish behavior.

In the first set of solutions, methods for identifying misbehaving users are designed. If a user is detected as greedy its use of the network is prohibited. The second set of approaches modify the protocol in a way that selfish misbehavior is discouraged, for example by penalty mechanisms. Our proposed solution in this paper belongs to the second category of solutions. Our modification to the original framework is by imposing a cost term on the data transmission of the users.

The rest of the paper is organized as follows. In Section II, we study the optimal cross-layered rate control algorithm of wireless networks. We explain greedy misbehavior model of a selfish user in Section III. In this section, we also study the effect of selfish behavior on the performance of the rate control mechanism. We present our proposed solution in Section IV. We finally conclude in Section V.

II. The Optimal Cross-Layered Rate Control of Wireless Networks

In this section, we explain the optimal cross-layered rate control mechanism presented in [1]. We first review the problem formulation. We then describe their proposed approach to solve the problem. In [1] and [2], the general category of multi-hop wireless networks is considered. Here, we present the single-hop version of their mechanism.

A. The Network Model

We consider a single-hop wireless network composed of L links. Set $\mathcal{L}$ denotes the set of all links in the network. In the single-hop traffic model, each route consists of only one link $l = (i, j)$ such that the transmission is from node i towards node j. We assume that associated with each link l is a user that uses the link for data transmission. Let x_l denote the arrival rate of link l, which is the rate with which data is injected to the input buffer of link l. It is usually assumed that x_l is upper bounded by a constant M_l, i.e. $0 \leq x_l \leq M_l$. Let $r_l(t)$ denote the transmission rate on link l during time slot t. $\bar{r}_l$ is the average transmission rate on link l, where the average is taken over time. $c_l(t)$ denotes the capacity of link l at time slot t, which is the maximum rate at which data can be transmitted on link l at time slot t. We assume that link capacities change over time. The user associated with link l has a utility function $U_l(x_l)$ which is a measure of satisfaction of the user when its allocated data rate is x_l. We assume that the utility function $U_l(\cdot)$ is strictly concave, non-decreasing and twice continuously differentiable on $(0, M_l]$.

B. Problem Formulation

Lin and Shroff formulated the optimal cross-layered rate control mechanism as follows [1].

1) We find the arrival rate of links $\vec{x} = (x_1, x_2, \cdots, x_L)$ to be the vector that maximizes the total utility of the network. In other words, $\vec{x}$ is the solution to the following maximization problem:

$$\max_{0 \leq x_l \leq M_l} \sum_{l=1}^{L} U_l(x_l) \tag{1}$$

subject to

$$x_l \leq \bar{r}_l \text{ for all } l \in \mathcal{L}$$

and

$$\vec{r} \in Co(\mathcal{R})$$

where $Co(\mathcal{R})$ denotes the convex hull of the average capacity region $\mathcal{R}$ of the wireless network. Precise definition of the capacity region can be found in [5].

2) We find the associated scheduling policy that stabilizes the system. In other words, we determine when and at which rate each link of the network transmits data.

Maximizing the total utility (1) is equivalent to achieving some fairness objective when the utility functions are chosen properly. A well-known family of utility functions is

$$U_l(x_l) = w_l \frac{{x_l}^{1-\beta}}{1-\beta}, \beta > 0 \tag{2}$$

Maximizing the total utility is equivalent to maximizing the weighted throughput as $\beta \to 0$, weighted proportional fairness as $\beta \to 1$, minimizing weighted potential delay as $\beta \to 2$, and max-min fairness as $\beta \to \infty$.

C. Solution to the cross-layered rate control problem

As discussed in [1] and [2], the solution to the problem (1) can be found iteratively. The iterative solution is as follows.

At each iteration t:

- The arrival data rates of the users are computed by

$$x_l(t) = \text{argmax}_{0 \leq x_l \leq M_l} \Big[U_l(x_l) - q_l(t) x_l \Big] \tag{3}$$

- The schedule is found by

$$\vec{r}(t) = \text{argmax}_{\vec{r} \in \mathcal{R}(t)} \sum_{l=1}^{L} q_l(t) r_l \tag{4}$$

- Each link updates its q_l by

$$q_l(t+1) = \Big[q_l(t) + \alpha_l [x_l(t) - r_l(t)] \Big]^+ \tag{5}$$

The equation (4) is known as the Backpressure scheduling algorithm, which is a centralized scheduling method. α_l is called the step size and should be chosen sufficiently small in order to achieve queue stability. Let Q_l denote the queue size of link l. In this framework, the approximate equation according which Q_l evolves is

$$Q_l(t+1) \approx \Big[Q_l(t) + [x_l(t) - r_l(t)] \Big]^+ \tag{6}$$

III. Misbehavior in the Optimal Cross-Layered Rate Control Mechanism

In this section, we address the problem of misbehavior in the framework of optimal cross-layered rate control. We explain how a misbehaving user associated with link m is able to increase its allocated throughput share x_m. We assume that the optimal cross-layered rate control algorithm of Section II-C is implemented in the network in order to determine both the arrival rates and the schedule. We consider a scenario in which each user computes (3) and updates (5) locally. On the other hand, the schedule is determined centrally according to (4). The scheduler needs the information about $q_l(t)$ and $\mathcal{R}(t)$ to determine the schedule at time slot t, i.e. the transmission rate vector $\vec{r}(t)$. In order to find the capacity region $\mathcal{R}(t)$, the scheduler needs to know the instantaneous link capacities $c_l(t)$ as well as the interference constraints of the network. In the case of time-varying link capacities, capacity region $\mathcal{R}(t)$ changes at each time slot t. We further assume that $Co(\mathcal{R}(t))$ is linear. In this case, a link is either not chosen for data transmission at time slot t, $r_l(t) = 0$, or it is scheduled to transmit data with the rate equal to its link capacity, $r_l(t) = c_l(t)$.

We now describe the misbehavior pattern of a cheating user m. We consider a case in which the scheduling component of the network believes that the link capacity of user m is $\hat{c}_m(t) = kc_m(t)$, with $k > 1$, while its actual link capacity is $c_m(t)$. This happens for example in a setting where the scheduler does not know the capacity of the links and is not able to measure them. Instead, each user is required to measure the capacity of its link and reports it to the scheduler. Another scenario is where the scheduler itself measures the link capacities, but the users are involved in the process of estimating and measuring the link capacities, and a cheater misbehaves in this process in order to mislead the scheduler.

If the cheater m is chosen for the data transmission at time slot t, it uses its actual link capacity $r_m(t) = c_m(t)$ in (5) for updating q_m. Equivalently, this means that the cheater m transmits with its actual link capacity $c_m(t)$ if it is scheduled. The reason for transmitting data with the actual link capacity $c_m(t)$ and not $\hat{c}_m(t)$ is that the data transmission on link m is not reliable if the transmission rate is higher than the link capacity $c_m(t)$.

Theorem 3.1: In the presence of a cheater m with the above explained misbehavior pattern, the corresponding utility optimization framework is

$$\max_{0\le x_l\le M_l} \sum_{l=1}^{L} \hat{U}_l(x_l) \tag{7}$$

subject to

$$x_l \le \bar{r}_l \text{ for all } l \in \mathcal{L} \tag{8}$$

and

$$\vec{\bar{r}} \in Co(\mathcal{R})$$

New utility functions $\hat{U}_l(\cdot)$ are related to the actual utility functions $U_l(\cdot)$ as follows:

$$\hat{U}_m(x_m) = kU_m(x_m)$$

$$\hat{U}_l(x_l) = U_l(x_l), \text{ for all } l \in \mathcal{L}, l \ne m$$

Proof: We define

$$\hat{q}_m(t) = kq_m(t), \hat{\alpha}_m(t) = k\alpha_m(t), \hat{U}_m(x_m) = kU_m(x_m)$$

For $l \in \mathcal{L}, l \ne m$, we define

$$\hat{q}_l(t) = q_l(t), \hat{\alpha}_l(t) = \alpha_l(t), \hat{U}_l(x_l) = U_l(x_l)$$

and rewrite (3)-(5) in the presence of the cheater. We first consider (3).

$$x_m(t) = \text{argmax}_{0\le x_m\le M_m}\Big[U_m(x_m) - q_m(t)x_m\Big]$$

$$= \text{argmax}_{0\le x_m\le M_m}\Big[U_m(x_m) - \frac{\hat{q}_m(t)}{k}x_m\Big] \tag{9}$$

$$= \text{argmax}_{0\le x_m\le M_m}\Big[kU_m(x_m) - \hat{q}_m(t)x_m\Big] \tag{10}$$

$$= \text{argmax}_{0\le x_m\le M_m}\Big[\hat{U}_m(x_m) - \hat{q}_m(t)x_m\Big] \tag{11}$$

(9) and (10) are equal because k is a positive constant. For the honest users,

$$x_l(t) = \text{argmax}_{0\le x_l\le M_l}\Big[\hat{U}_l(x_l) - \hat{q}_l(t)x_l\Big] \tag{12}$$

We now rewrite (4)

$$\vec{r}(t) = \text{argmax}_{\vec{r}\in\hat{\mathcal{R}}(t)}\Big[q_m(t)\hat{r}_m + \sum_{l=1,l\ne m}^{L} q_l(t)r_l\Big]$$

$$= \text{argmax}_{\vec{r}\in\mathcal{R}(t)}\Big[q_m(t)kr_m + \sum_{l=1,l\ne m}^{L} q_l(t)r_l\Big]$$

$$= \text{argmax}_{\vec{r}\in\mathcal{R}(t)}\Big[\hat{q}_m(t)r_m + \sum_{l=1,l\ne m}^{L} \hat{q}_l(t)r_l\Big]$$

$$= \text{argmax}_{\vec{r}\in\mathcal{R}(t)}\sum_{l=1}^{L}\hat{q}_l(t)r_l \tag{13}$$

The update equation of q_m is given by (5):

$$q_m(t+1) = \Big[q_m(t) + \alpha_m\Big(x_m(t) - r_m(t)\Big)\Big]^+$$

We multiply both sides of the above by k

$$kq_m(t+1) = \Big[kq_m(t) + k\alpha_m\Big(x_m(t) - r_m(t)\Big)\Big]^+$$

This is equivalent to

$$\hat{q}_m(t+1) = \Big[\hat{q}_m(t) + \hat{\alpha}_m\Big(x_m(t) - r_m(t)\Big)\Big]^+ \tag{14}$$

Since $\hat{q}_l = q_l$ and $\hat{\alpha}_l = \alpha_l$ for all $l \ne m$, their update equation is given by

$$\hat{q}_l(t+1) = \Big[\hat{q}_l(t) + \hat{\alpha}_l\Big(x_l(t) - r_l(t)\Big)\Big]^+ \tag{15}$$

Considering (11)-(15), we conclude that the optimal cross layered rate control algorithm in the presence of a cheater is given by the following iterations. For all l (including the cheater m) and at each iteration t,

$$x_l(t) = \text{argmax}_{0\le x_l\le M_l}\Big[\hat{U}_l(x_l) - \hat{q}_l(t)x_l\Big] \tag{16}$$

$$\vec{r}(t) = \text{argmax}_{\vec{r}\in\mathcal{R}(t)}\sum_{l=1}^{L}\hat{q}_l(t)r_l \tag{17}$$

$$\hat{q}_l(t+1) = \Big[\hat{q}_l(t) + \hat{\alpha}_l\Big(x_l(t) - r_l(t)\Big)\Big]^+ \tag{18}$$

(16)-(18) are subgradient descent iterations of the dual of the following problem.

$$\max_{0\le x_l\le M_l} \sum_{l=1}^{L} \hat{U}_l(x_l) \tag{19}$$

subject to

$$x_l \leq \bar{r}_l \text{ for all } l \in \mathcal{L} \tag{20}$$

and

$$\vec{\bar{r}} \in Co(\mathcal{R})$$

■

The above theorem considers the case where only one cheater is present in the network. We note that the theorem can be easily extended to the case of more than one misbehaving user. In this case, in the corresponding optimization framework, the utility function of each cheater is multiplied by the ratio of its claimed link capacity to its actual link capacity.

According to Theorem 3.1, a misbehaving link m is able to increase its utility function by a multiplicative factor k, while the utility function of the honest users remain unchanged. As a result, the cheater might receive higher throughput share than its fair share. In the following subsection, we show an example in which we compute how much the throughput share of a cheater is increased, when it misleads the scheduler about its link capacity.

A. A wireless network of L conflicting wireless links

In this section, we consider a wireless network composed of L links where all of them are conflicting with each other. This means that only one of them can transmit data at each time slot. For simplicity, we assume that the link capacity of each link is a constant c ($c_l = c$, $\forall l$). The utility function of user l is

$$U_l(x_l) = w_l \frac{{x_l}^{1-\beta}}{1-\beta}$$

The arrival rate vector $\vec{x}$ is determined by

$$\max_{0 \leq x_l \leq M_l} \sum_{l=1}^{L} w_l \frac{{x_l}^{1-\beta}}{1-\beta} \tag{21}$$

subject to

$$x_l \leq r_l \text{ for all } l$$

$$\sum_{l=1}^{L} r_l = c$$

In this example, $Co(\mathcal{R})$ is the line defined by $\sum_{l=1}^{L} r_l = c$. The corresponding Lagrange function of (21) is

$$L(x_l, \gamma) = \sum_{l=1}^{L} w_l \frac{{x_l}^{1-\beta}}{1-\beta} - \gamma(\sum_{l=1}^{L} x_l - c)$$

We solve

$$\frac{dL(x_l,\gamma)}{dx_l} =$$

$$w_l x_l^{-\beta} - \gamma = 0 \tag{22}$$

According to the KKT condition

$$\gamma(\sum_{l=1}^{L} x_l - c) = 0 \tag{23}$$

(23) has two solutions: $\gamma = 0$ and $\sum_{l=1}^{L} x_l = c$. $\gamma = 0$ results in $x_l = \infty$, which is not a feasible solution. So, KKT condition implies that

$$\sum_{l=1}^{L} x_l = c \tag{24}$$

Considering (22), for node 1 and node $l \neq 1$,

$$w_1 x_1^{-\beta} = w_l x_l^{-\beta}$$

$$x_l = (\frac{w_l}{w_1})^{1/\beta} x_1 \tag{25}$$

Considering (25) and (24), we find the solution to (21):

$$x_l = c \frac{{w_l}^{1/\beta}}{\sum_{l=1}^{L} {w_l}^{1/\beta}} \tag{26}$$

We now consider a case where user 1 misleads the scheduler such that the scheduler assumes that $c_1 = kc$, where $k > 1$. Later, if user 1 is chosen for the data transmission at time slot t, it transmits with its actual link capacity $r_1(t) = c$. As we explained in Section III, this selfish misbehavior changes the optimization framework to the following

$$\max_{0 \leq x_l \leq M_l} k w_1 \frac{{x_1}^{1-\beta}}{1-\beta} + \sum_{l=2}^{L} w_l \frac{{x_l}^{1-\beta}}{1-\beta} \tag{27}$$

subject to

$$x_l \leq r_l \text{ for all } l$$

and

$$\sum_{l=1}^{L} r_l = c$$

The solution to (27) is

$$x_1 = c \frac{(k w_1)^{1/\beta}}{(k w_1)^{1/\beta} + \sum_{l=2}^{L} {w_l}^{1/\beta}} \tag{28}$$

$$x_l = c \frac{{w_l}^{1/\beta}}{(k w_1)^{1/\beta} + \sum_{l=2}^{L} {w_l}^{1/\beta}}, \text{ for } l \neq 1$$

Comparing (26) and (28), we observe that since $k > 1$, the throughput share of the cheater is increased from $c\frac{{w_1}^{1/\beta}}{\sum_{l=1}^{L} {w_l}^{1/\beta}}$ to $c\frac{(kw_1)^{1/\beta}}{(kw_1)^{1/\beta}+\sum_{l=2}^{L} {w_l}^{1/\beta}}$ at the expense of decreasing the throughput share of honest users $l \neq 1$ from $c\frac{{w_l}^{1/\beta}}{\sum_{l=1}^{L} {w_l}^{1/\beta}}$ to $c\frac{{w_l}^{1/\beta}}{(kw_1)^{1/\beta}+\sum_{l=2}^{L} {w_l}^{1/\beta}}$.

IV. Cost Function for Preventing/Alleviating Greedy Misbehavior in a General Setting

In this section, we present a solution that cancels or alleviates the effect of a selfish user on the performance degradation of honest users. The selfish user misbehaves according to the pattern described in Section III. Our solution is based on imposing a cost (or equivalently a penalty) on the data transmission of the users. A lot of cost and reward metrics are developed in the networking literature. Penalties and rewards provide strong mechanisms for solving network problems and achieving performance objectives. A comprehensive introduction to the network penalties and rewards can be found in [9].

By imposing cost function $C_l(x_l)$ on the data transmission of the users, the optimization framework of (1) is modified as:

$$\max_{0\le x_l\le M_l} \sum_{l=1}^{L}\left[U_l(x_l)-C_l(x_l)\right] \tag{29}$$

subject to

$$x_l \le \bar{r}_l \text{ for all } l\in\mathcal{L}$$

and

$$\vec{\bar{r}} \in Co(\mathcal{R})$$

The optimal cost function $C_l(\cdot)$ is the one for which the following two conditions are satisfied:

1) In the absence of any cheater, the optimal point of (1) and (29) are the same.
2) In the presence of a cheater m, the arrival rate of the cheater determined by (29) is no greater than its fair share, which is its allocated throughput determined by (1) when it is not cheating. Furthermore, the arrival rate of each honest user determined by (29) is no less than its fair share determined by (1).

The second requirement means that when a user is cheating, its allocated throughput share should be less than or equal to its fair share when it is honest. At the same time, the allocated throughput share to the honest users should be greater than or equal to their fair share. For instance, in the example III-A, the solution to (29) when user 1 is cheating should be $x_1 \le c\frac{{w_1}^{1/\beta}}{\sum_{l=1}^{L}{w_l}^{1/\beta}}$, $x_l \ge c\frac{{w_l}^{1/\beta}}{\sum_{l=1}^{L}{w_l}^{1/\beta}}, l\neq 1$.

In the following subsection, we first look at the example of Section III-A and explain which cost function prevents selfish misbehavior and how it can be found. Later, we explain how in a general network the cost function is found.

A. Cost term of L conflicting wireless links

We consider the network described in Section III-A when the cost term λx_l is subtracted from the utility function of the users. We call λ the cost factor. We discuss if a linear cost function suffices to completely cancel the effect of the cheater, in this example network. If so, what value of λ achieves this objective. User 1 misbehaves in the way we explained earlier. The equivalent optimization framework for finding the arrival rate vector $\vec{x}$ is

$$\max_{0\le x_l\le M_l} k\left[w_1\frac{{x_1}^{1-\beta}}{1-\beta}-\lambda x_1\right]+\sum_{l=2}^{L}\left[w_l\frac{{x_l}^{1-\beta}}{1-\beta}-\lambda x_l\right]$$

subject to

$$x_l \le r_l, \text{ for all } l\in\mathcal{L} \tag{30}$$

and

$$\sum_{l=1}^{L} r_l = c$$

Our goal is to find λ such that no matter what k is the solution to (30) be

$$x_l = c\frac{{w_l}^{1/\beta}}{\sum_{l=1}^{L}{w_l}^{1/\beta}} \tag{31}$$

(31) is equal to the share of user l when link 1 does not misbehave, i.e. $k=1$, and no cost is imposed on data transmission of the users, i.e. $\lambda=0$. Such a λ completely cancels the effect of the misbehaving user on throughput degradation of honest users. We are interested to figure out if such an optimal λ exists or not in this example network. The corresponding Lagrange function of (30) is

$$L(x_l,\gamma)=$$

$$k\left[w_1\frac{{x_1}^{1-\beta}}{1-\beta}-\lambda x_1\right]+\sum_{l=2}^{L}\left[w_l\frac{{x_l}^{1-\beta}}{1-\beta}-\lambda x_l\right]-\gamma(\sum_{l=1}^{L}x_l-c)$$

$\frac{dL(x_l,\gamma)}{dx_l}=0$ results in

$$kw_1x_1^{-\beta}-k\lambda-\gamma=0 \tag{32}$$

$$w_lx_l^{-\beta}-\lambda-\gamma=0, l\neq 1 \tag{33}$$

According to the KKT condition

$$\gamma(\sum_{l=1}^{L}x_l-c)=0 \tag{34}$$

(34) has two solutions: $\gamma=0$ and $\sum_{l=1}^{L}x_l=c$. $\gamma=0$ results in

$$x_l=(\frac{w_l}{\lambda})^{1/\beta}, \forall l \tag{35}$$

By choosing $\gamma=0$, we find the global optimum point of the optimization problem, i.e. we solve the unconstrained version of the optimization problem (we ignore the constraints). (35) is feasible (i.e. the global optimum point is inside the feasibility region), if $\sum_{l=1}^{L}x_l\le c$ is satisfied. $\sum_{l=1}^{L}x_l\le c$ is hold only if

$$\lambda \ge [\frac{1}{c}\sum_{l=1}^{L}{w_l}^{1/\beta}]^{\beta} \tag{36}$$

We conclude that for λ satisfied in (36), the global optimum (35) is the same as the solution of the constrained problem (30).

By choosing $\lambda=[\frac{1}{c}\sum_{l=1}^{L}{w_l}^{1/\beta}]^{\beta}$, the allocated arrival rate to each user would be exactly the same as its arrival rate when user 1 does not misbehave. In other words, we have found a cost factor that achieves the fair share no matter if a node is cheating or not. We also observe that if $\lambda>[\frac{1}{c}\sum_{l=1}^{L}{w_l}^{1/\beta}]^{\beta}$, the constraint of the maximization problem is satisfied with strict inequality. This means that the network is under utilized, i.e. the allocated rate vector is not on the boundary of the capacity region, instead it is inside the capacity region. In order to fully utilize the network as well as completely canceling the effect of the cheater, we choose $\lambda = [\frac{1}{c}\sum_{l=1}^{L} {w_l}^{1/\beta}]^{\beta}$.

B. Finding the cost factor in a general setting

In a general wireless network, where network topology and utility functions of the users are arbitrary, finding the optimal cost function is not an easy task. In an arbitrary setting, the optimal cost term is not necessarily linear. In this work, we approximate the best cost function $C_l(x_l)$ with a linear term λx_l. Our goal is to find λ such that we prevent the cheater from obtaining extra throughput as much as possible, while utilizing the network resources completely.

Considering our computations in Section IV-A, we note that we solved an optimization problem (30) in order to find the cost factor. But in an arbitrary wireless network with a complex set of constraints, following this method for finding λ is difficult. It requires solving the corresponding optimization problem centrally, which might be highly complex.

Instead, we propose a heuristic for finding λ without solving the maximization problem centrally. Our proposed solution is an iterative method in which cost terms of the form λx_l with different λ's are imposed on the data transmission of the users. For each λ, the underlying network implements iterations (3), (4) and (5) and becomes stable at an arrival rate vector $\vec{x}$, which is the solution to the optimization problem (29). We start from $\lambda = 0$, and we increase λ up to a point where the network resources are under utilized, i.e. the λ for which some constraints of the optimization problem are satisfied with strict inequality. The value of λ after which the network is under utilized is chosen as the cost factor. This is the maximum cost that can be imposed on the data transmission of the users, without under utilizing the network resources. We note that in this method, no knowledge about the utility function of the users is required. The only information we need to know is the average capacity region $\mathcal{R}$, in order to determine the set of constraints.

C. Simulation Results

To make our method more clear, we run a simulation in which we implemented our proposed heuristic in order to find the cost factor λ. The network is composed of 6 wireless links with the corresponding link contention graph depicted in Fig. 1. In this figure, a vertex represents a link of the network. An edge connecting vertices i and j ($1 \leq i, j \leq 6$) denotes the conflict between links i and j, i.e. it shows that links i and j can not transmit at the same time. The actual link capacity of each link is $c_l = 1$. All links have logarithmic utility function $U_l(x_l) = \log(x_l)$. Link 1 misleads the scheduler such that the scheduler assumes that the capacity of link 1 is $c_1 = k$ instead of $c_1 = 1$. The utility maximization framework for this network is

$$\max_{0 \leq x_l \leq M_l} k\log(x_1) - k\lambda x_1 + \sum_{l=2}^{6}\left[\log(x_l) - \lambda x_l\right] \quad (37)$$

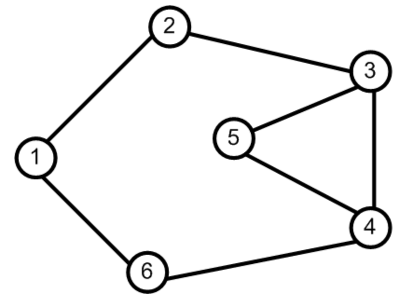

Fig. 1. Link Contention Graph

subject to

$$x_1 + x_2 + x_6 \leq 1, x_1 + x_2 + x_3 \leq 1, x_1 + x_4 + x_6 \leq 1$$

$$x_2 + x_3 + x_4 + x_5 \leq 1, x_3 + x_4 + x_5 + x_6 \leq 1$$

We first consider the case where $\lambda = 0$ (no cost) and $k = 1$ (link 1 is honest). The iterative algorithm of Section II-C stabilizes at the point

$$\vec{x} = (0.5, 0.25, 0.25, 0.25, 0.25, 0.25) \quad (38)$$

If $\lambda = 0$ (no cost) and $k = 5$ (link 1 is cheating), the result of the iterations is

$$\vec{x} = (0.64, 0.18, 0.18, 0.18, 0.46, 0.18)$$

We observe that the cheater obtains higher throughput than its fair share. Now, we consider the case where a linear cost term (λx_l) with $\lambda = 1$ is imposed on all links. With $k = 5$ (link 1 is cheating), the stable point of the iterations is

$$\vec{x} = (0.55, 0.22, 0.22, 0.22, 0.33, 0.22)$$

We observe that with this choice of λ, all constraints are satisfied with equality. So, we continue increasing λ. With $\lambda = 3$ and $k = 5$, the result of the iterations is

$$\vec{x} = (0.33, 0.28, 0.24, 0.24, 0.24, 0.28)$$

In this case, the network is under utilized because the first constraint is not satisfied with equality ($x_1 + x_2 + x_6 = 0.89 < 1$). By changing λ between 1 and 3, we find that $\lambda = 2.3$ is the point after which some of the constraints are satisfied with strict inequality. We conclude that $\lambda = 2.3$ should be chosen as the cost factor in this example network. The allocated arrival rates in this case ($\lambda = 2.3$ and $k = 5$) are

$$\vec{x} = (0.43, 0.29, 0.24, 0.24, 0.24, 0.29) \quad (39)$$

So, in a general network in order to find the cost factor, a central authority imposes different λ's, starting from $\lambda = 0$ and increasing up to a point where some constraints of the utility maximization problem are satisfied with strict inequality. This value of λ will be chosen as the cost factor for the rest of the network operation.

V. Conclusion

In this work, we considered the problem of selfish misbehavior in the optimal cross-layered rate control scheme of wireless networks. We explained how in this framework, a selfish user is able to increase its allocated throughput by misleading the scheduler about its link capacity. We have found what the effect of this misbehavior in the optimization framework of cross-layered rate control is. We also presented a solution to prevent or alleviate the effect of such a greedy user. Our solution is based on subtracting a linear cost term from the utility function of the users. We computed the optimal cost term for one example network. We also described how the appropriate cost term is found in a general network.

References

[1] X. Lin and N. B. Shroff, "Joint Rate Control and Scheduling in Multihop Wireless Networks," in *Proceedings of the IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[2] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," in *Proceedings of IEEE INFOCOM*, vol. 3, pp. 1804 - 1814, March 2005.

[3] Gaurav Sharma, Ness B. Shroff and Ravi R. Mazumdar, "Joint Congestion Control and Distributed Scheduling for Throughput Guarantees in Wireless Networks," in *Proceedings of IEEE INFOCOM*, pp. 2072-2080, May 2007.

[4] T. Bonald and L. Massouli, "Impact of fairness on Internet performance," in *Proceedings of ACM Sigmetrics*, Cambridge, MA, pp.82-91, June 2001.

[5] A. Gupta, X. Lin and R. Srikant, "Low-Complexity Distributed Scheduling Algorithms for Wireless Networks," in *Proceedings of IEEE INFOCOM*, pp. 1631 - 1639, May 2007.

[6] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control,* vol. 37, no. 12, pp. 19361948, December 1992.

[7] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in *Proceedings of 46th Annual Allerton Conference on Communication, Control, and Computing,* pp. 1511 - 1519, September 2008.

[8] A.A. Cardenas, S. Radosavac and J.S. Baras, "Evaluation of Detection Algorithms for MAC Layer Misbehavior: Theory and Experiments," *IEEE/ACM Transactions on Networking,* vol. 17, no. 2, pp. 605 - 617, April 2009.

[9] L. Georgiadis, M. Neely and L. Tassiulas, "Resource Allocation and Cross Layer Control in Wireless Networks," *Foundations and Trends in Networking,* vol. 1, no. 1, 2006.