Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks

Lijun Chen, Steven H. Low and John C. Doyle
Engineering & Applied Science Division, California Institute of Technology
Pasadena, CA 91125
{chen@cds., slow@, doyle@cds.}caltech.edu

***Abstract*— We present a model for the joint design of congestion control and media access control (MAC) for ad hoc wireless networks. Using contention graph and contention matrix, we formulate resource allocation in the network as a utility maximization problem with constraints that arise from contention for channel access. We present two algorithms that are not only distributed spatially, but more interestingly, they decompose vertically into two protocol layers where TCP and MAC jointly solve the system problem. The first is a primal algorithm where the MAC layer at the links generates congestion (contention) prices based on local aggregate source rates, and TCP sources adjust their rates based on the aggregate prices in their paths. The second is a dual subgradient algorithm where the MAC sub-algorithm is implemented through scheduling link-layer flows according to the congestion prices of the links. Global convergence properties of these algorithms are proved. This is a preliminary step towards a systematic approach to jointly design TCP congestion control algorithms and MAC algorithms, not only to improve performance, but more importantly, to make their interaction more transparent.**

***Index Terms*— Congestion control, Media access control, Convex optimization, Cross-layer design, Dual decomposition, Subgradient method, Ad hoc wireless network.**

I. INTRODUCTION

We consider the problem of congestion control over a multihop wireless ad hoc network. This has been an active research area over the past few years (see, e.g., [15], [5], [9], [30], [12], [37], [38], [6]) with many fascinating and complex issues, involving, e.g., mobility, channel estimation, power control, MAC, routing, etc. Unlike most of previous work however we focus on the interaction of congestion control at the transport layer and channel contention at the MAC layer, and ignore all other issues. Our goal is to present a systematic approach to jointly design TCP congestion control algorithms and MAC algorithms, not only to improve performance, but more importantly, to make their interaction more transparent.

This is motivated by two observations. First, wireless channel is a shared medium and interference-limited. Link is only a logical concept and links are correlated due to the interference with each other. Under the MAC strategies such as time-division multiple access and random access, these links contend for exclusive access to the physical channel. Unlike in the wireline network where flows compete for transmission resources only when they share the same link, here, network layer flows that do not even share a wireless link in their paths can compete. Thus, in ad hoc wireless networks the contention relations between link-layer flows provide fundamental constraints for resource allocation. Second, TCP congestion control algorithms can be interpreted as distributed primal-dual algorithms over the Internet to maximize aggregate utility, and a user's utility function is (often implicitly) defined by its TCP algorithm, see e.g. [18], [22], [21]. This series of work implicitly assumes a wireline network where link capacities are fixed and shared by flows that traverse common links. A natural formulation for the joint design of congestion and media access control is then the utility maximization framework with new constraints that arise from channel contention.

After a brief description of the interaction between TCP congestion control and MAC in Section II and a brief review of related work in Section III, we explain in Section IV contention graph and introduce contention matrix to model resource constraints in wireless networks, and state our utility maximization problem with MAC constraints. In Section V, we follow [18] and derive a primal algorithm to solve a relaxation of the problem, and prove its global convergence. The algorithm is not only distributed spatially, more interestingly, it decomposes vertically into two protocol layers where the MAC layer at the links generates congestion (contention) prices based on local aggregate source rates, and TCP sources adjust their rates based on the aggregate prices in their paths. Whereas congestion prices are generated by AQM (active queue management) algorithms in routers in wireline networks (e.g. [23]), here they are generated by the MAC layer. We discuss how to design contention resolution protocols to generate the necessary prices.

In Section VI, we apply duality theory to derive another decomposition of the system problem into congestion control subproblem and MAC subproblem. The key idea is to introduce the "effective capacity" of a link, which is the maximum average data rate a link can achieve without violating schedulability constraint. The Lagrangian of the resulting problem separates into two maximization subproblems, one over source rates, to be solved by TCP, and the other over effective capacity, to be solved by MAC. The introduction of the effective capacity makes the primal problem not strictly concave, and hence the dual function non-differentiable. A subgradient algorithm that generalizes the algorithm of [22] is derived to solve the dual problem, and proved to approach arbitrarily close to an optimal point starting from any initial condition. This algorithm motivates a joint design scheme where link-layer flows are scheduled according to congestion prices of the links. We illustrate with numerical examples of

0-7803-8968-9/05/$20.00 (C)2005 IEEE

such a design.

Finally, we conclude in Section VII with limitations of this paper and possible extensions.

II. MOTIVATION

TCP was originally designed for wireline networks, where the links are assumed to be reliable and with fixed capacities. This may not be true for wireless networks, where the links are "elastic" due to the fact that the wireless channel is unreliable (e.g., fading and node mobility) and interference-limited. We need to exploit the interaction between transport and MAC/physical layers, in order to improve the performance.

This paper does not consider the node mobility or channel fading, but focuses on the broadcast and interference-limited nature of wireless channel. In this context, a fundamental problem is to provide an efficient bandwidth sharing mechanism among the competing link-layer flows. Many existing wireless MAC protocols, such as distributed coordination function (DCF) specified in IEEE 802.11 standard[17], are traffic-independent and do not consider the actual requirements of the flows competing for the channel. These MAC protocols suffer from the unfairness problem, caused by the location dependency of the contentions, and exacerbated by the contention resolution mechanisms such as the binary exponential backoff algorithm adopted in DCF. When they interact with TCP, TCP will further penalize these flows with more contention. This

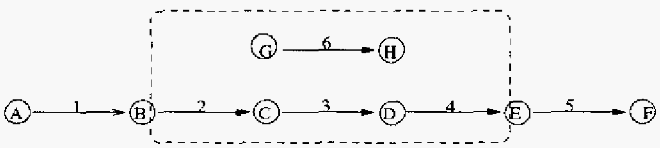

Fig. 1. Example of ad hoc wireless network

will result in significant TCP unfairness in ad hoc wireless networks [13], [28], [35], [36], [37]. To illustrate this, consider the example in Fig.1, and assume there are four network-layer flows $A \rightarrow B$, $C \rightarrow D$, $E \rightarrow F$ and $G \rightarrow H$. The flow $C \rightarrow D$ experiences more contention and will build up queue faster than the other three flows. TCP will further penalize it by reducing the congestion window more aggressively, and the resulting throughput of flow $C \rightarrow D$ will be much less than that of other flows.

In addition to the location dependency of contentions, correlation among links is also the key to understand the interaction between transport and MAC layers. In wireline networks, link bandwidth is well-defined and links are disjoint resources. But in wireless networks, as we mentioned above, links are correlated due to the interference with each other, and network-layer flows, which do not transverse a common link, may still compete with each other. Thus, congestion is located at some spatial contention region [37]. Consider again the example in Fig.1, and assume there are two network-layer flows $A \rightarrow F$ and $G \rightarrow H$. Link-layer flows 2, 3, 4 and 6 contend with each other, and congestion is located in the spatial contention region denoted by the rectangle. So, unlike wireline networks where link capacities provide constraints for resource allocation, in ad hoc wireless networks the contention relations between link-layer flows provide fundamental constraints for resource allocation.

In this paper we will model the contention relations between link-layer flows as a flow contention graph (see, e.g., [25], [11]). This construction captures the location-dependent contention among link-layer flows. Based on the contention graph, we will use a contention matrix to mathematically formulate the contention constraints imposed by the MAC layer. We then model the resource allocation for ad hoc wireless networks as a concave utility maximization problem with MAC layer constraints, with which we can explicitly exploit the interaction between transport and MAC layers, and systematically carry out joint design of congestion and media access control.

III. RELATED WORK

The work in [18], [22], [21], [23] provides a utility-based optimization framework for internet congestion control. The same framework has been applied to study the congestion control over ad hoc wireless networks (see, e.g., [6], [38]). In [38], the authors study congestion control in ad hoc wireless network with primary interference, and formulate rate allocation as a utility maximization problem with time constraint. It assumes that the MAC protocol is given, and does not consider the problem of how the link-layer flows share the congestion price generated by the constraint. In our work, we will consider the networks with both primary and secondary interference, and jointly design congestion control and MAC.

Many schemes have been proposed for fair bandwidth sharing at link layer (see, e.g., [25], [33], [24], [16], [29], [11]). Some of these schemes try to achieve weighted fairness, but they usually assume the weights are given and do not address the issue of how to choose those weights. In our work, these weights or their equivalent are related to the actual flow requirements or the congestion prices of the links, which guarantees some kind of network layer fairness. In [29], the authors propose a maximin fair scheduling which assigns congestion-dependent weights to the flows with primary interference and schedules the flows via maximum weighted matching. In [25], [11], the authors use the flow contention graph to characterize the contention among link-layer flows, and propose utility-based optimization to achieve MAC layer fairness. We will modify a multiple access scheme proposed in [25] to implement AQM for congestion control. Also, some of our discussions on the flow feasibility is recaptured from [11] for completeness.

In [37], the authors propose a neighborhood RED scheme to improve TCP fairness in ad hoc wireless networks. Basically, this scheme assigns more share of congestion price to the flows with less contention to alleviate TCP unfairness. We try to address the unfairness problem that arise in the MAC layer by using traffic-dependent MAC scheme.

Cross-layer design in communication networks, especially in wireless networks, have attracted great attention recently (see, e.g., [26] for an overview). Our work belongs to the category of cross-layer design via dual decomposition in optimization framework. Other work that can be put into this category includes TCP/IP interaction in [31], joint routing and resource allocation in [34] and joint TCP and power control in [6]. The work on joint congestion control and MAC design is the first step in our attempt to provide a unified framework for systematically carrying out cross-layer design through dual decomposition. We will extend the framework to include other layers in the future.

IV. System Model

Consider an ad hoc wireless network with a set V of vertices (nodes) and a set L of logical links. We assume a static topology and each link l has a fixed finite capacity c_l^0 packets per second when active, i.e., we implicitly assume a power control algorithm that maintains a constant data rate in the face of fading and other channel imperfections. Wireless channel is a shared medium and interference-limited. In this paper, we assume logical links contend for channel access and the successful link transmits at rate c_l^0 for the duration it holds the channel. We will focus on the interaction of MAC and TCP, and characterize the contention relations using contention graph and contention matrix. The joint MAC and TCP design is then formulated as a utility maximization problem with the constraints that arise from MAC layer contention.

A. Flow Contention Graph and Contention Matrix

Wireless nodes are assumed to be able to communicate with at most one other node at any given time. This follows from the fact that a node cannot transmit or receive simultaneously. Links mutually interfere with each other whenever either the sender or the receiver of one is within the interference range of the sender or receiver of the other. Under these assumptions, we can construct a flow contention graph that captures the contention relations between the links of the network (see, e.g., [25], [11]). In the contention graph, each vertex represents an active link, and an edge between two vertices denotes the contention between the corresponding links: two links interfere with each other and cannot be active at the same time. An accurate flow contention graph could be constructed based on the protocol model or physical SIR model, and also depends on the the basic multiple access strategy used. In practice, when we construct the flow contention graph, we can assume two links contend with each other if they are within each other's carrier sensing range.

Given a contention graph, we can identify all its maximal cliques[1]. Maximal cliques are local constructions and capture the local contention relations of the flows. Flows within the same maximal clique cannot transmit simultaneously, but flows in different cliques may transmit simultaneously. For example, Fig. 2 shows the flow contention graph that corresponds to the ad hoc wireless network of Fig. 1 with 6 active link-layer flows. Flows 1, 2 and 3, which are in the same clique, cannot transmit simultaneously, neither can flows 2, 3, 4 and 6 . But flows 1 and 6 can be activated simultaneously, since they belong to different cliques. Thus, each maximal clique in the contention graph represents a "channel resource" with flows

[1]A maximal clique of a graph is a maximal complete subgraph of the graph.

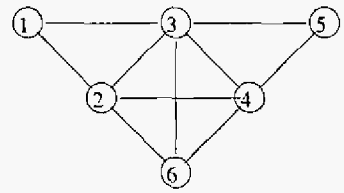

Fig. 2. Flow contention graph and maximal cliques: flows (1, 2, 3) and flows (3, 4, 5) are two maximal cliques of size 3, flows (2, 3, 4, 6) is a maximal clique of size 4.

in the clique contending for exclusive access to the resource [25]. The flows within the same clique share the "capacity" of the clique. A flow may belong to several cliques, and can successfully transmit if and only if it is the only active flow in all cliques to which it belongs.

We now consider the problem of determining if a set of link flows are feasible, i.e., whether a schedule can be found to achieve this set of flows (see, e.g., [14], [20]). This will be the constraint imposed by the MAC layer. Assume that we are given a L-dimensional vector y where y_l is the desired flow on link l, in packets per second. We refer to y as the link-layer flow vector. On average, given link flow y_l, the fraction of time required to send this amount of flow is y_l/c_l^0. We refer to y_l/c_l^0 as the normalized flow rate of link l. Since flows within the same clique cannot transmit simultaneously, we obtain a necessary scheduling constraint:

$$\sum_l \frac{y_l}{c_l^0} \leq 1$$

where the summation is over those links that belong to the same clique. We can represent the scheduling constraints in a compact form by introducing contention matrix. Suppose the flow contention graph can be decomposed into a set N of maximal cliques indexed by n. Each clique n contains a set $L_n \subset L$ of links. The sets L_n define a $N \times L$ contention matrix F

$$F_{nl} = \begin{cases} 1/c_l^0 & \text{if } l \in L_n \\ 0 & \text{otherwise} \end{cases}$$

Thus, the above scheduling constraints can be written as

$$Fy \leq \mathbf{1} \tag{1}$$

where $\mathbf{1}$ denotes a N-dimensional vector with each component being 1.

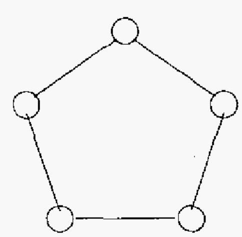

Fig. 3. Ring graph of size 5: by equation (1) the maximal normalized sum rate is $\frac{5}{2}$, but the actual maximal sum rate is 2.

Since the above description is a fluid-level description, i.e., we average the scheduling variables over time, constraint (1) is only a necessary condition for the feasibility of the flow vector y. To illustrate this, consider the example in Fig.3, where the contention graph is a ring of size 5. According to the constraint (1), each flow should attain a normalized rate of $1/2$ if the max-min fairness allocation criterion is used. However, scheduling the links according to the max-min fairness criterion allocates only a rate of $2/5$ to each link, since at anytime at most two links can transmit simultaneously.

Given a flow vector y, it is not an easy job to verify its feasibility, since this is equivalent to finding a schedule that achieves y. It can be shown that a feasible flow vector must be a convex combination of the characteristic vectors of all independent sets of the flow contention graph[2], and that the set of achievable flow vectors is a closed, convex and compact set (see [1], also cited in [11]). In addition, constraint (1) is also a sufficient condition for the feasibility of the flow vector if and only if the contention graph is a perfect graph[3] (see [1], also cited in [11]). According to the strong perfect graph theorem [8], [7], a graph is perfect if and only if it has no induced subgraph that is isomorphic to an odd hole[4], or its complement. Therefore if there exist odd holes in a contention graph, the sum of the normalized flow rates of any clique that includes edges of an odd hole should be reduced.

In general, it is hard to tell whether a graph is perfect or not. Such classification may require the global topology information of the graph (e.g., an odd hole can span the whole graph). Since the algorithms for ad hoc networks are desired to be distributed and depend at most on local message passing, we need to trade off the accuracy (and even some performance optimality) for the simplicity of the design. Hence, we will not verify whether a graph is perfect or not, but reduce the sum of the normalized rates of a clique to ensure flow feasibility. Determining exactly by how much we should reduce the sum rate is difficult and also depends on the basic fairness criterion we choose. In this paper, we will not further discuss this issue, but assume a maximal clique sum rate vector ε. The value of ϵ will depend on local topology of the contention graph. Thus, the constraint imposed by the MAC layer can be written as

$$Fy \leq \varepsilon \tag{2}$$

We will see later that we do not need to know the value of ε, since in the joint design in section V we will relax the constraint (2), and in the joint design in Section VI this constraint can be replaced with the constraint (1) with some additional constraint on the value that y can take.

Note that the contention graph and contention matrix is a rather general construction. It includes wireline networks as a special case where the contention matrix F is a $L \times L$ identity matrix, since there is no interference among the links. It can be used to characterize the interference relations among wireless and wired links in hybrid wireline-wireless networks. It can also be modified to characterize the contention relations in the frequency-division or other strategies for channel access.

B. Problem Formulation

Assume the network is shared by a set S of sources indexed by s. Each source s uses a set $L^s \subset L$ of links. The sets L^s define an $L \times S$ routing matrix

$$R_{ls} = \begin{cases} 1 & \text{if } l \in L^s \\ 0 & \text{otherwise} \end{cases}$$

We will fix the routing matrix R and focus on congestion control. Each source s attains a utility $U_s(x_s)$ when it transmits at rate x_s packets per second. We assume U_s is continuously differentiable, increasing, strictly concave, and unbounded as $x_s \to 0$. Our objective is to choose source rates x so as to [18], [22], [21]:

$$\max_{x_s \geq 0} \quad \sum_s U_s(x_s) \tag{3}$$

$$\text{subject to} \quad FRx \leq \varepsilon \tag{4}$$

The constraint (4) follows from (2) with $y = Rx$. A unique maximizer exists, since the objective function is strictly concave and feasible set is convex and compact.

We can see the system problem (3)-(4) from two complement perspectives. On one hand, it is a utility-based congestion control problem with the MAC layer constraints. As such, the congestion prices are not decided by the link capacity, but determined by the contention region. In other words, the MAC layer imposes the ultimate constraints to the achievable rates. On the other hand, it is a media access control problem, which is to allocate physical bandwidth to each link, with the objective of maximizing aggregate end user utilities. As such, the resulting MAC protocol is traffic-dependent and will allocate more bandwidth to the links with more contention to alleviate flow congestion.

Solving the system problem (3)-(4) directly requires coordination among possibly all sources and is impractical in real network. According to the theory of convex optimization, distributed algorithms can be derived by considering its relaxation and dual problem. In the next two sections, we will solve these two problems and give them different interpretations in the context of joint design of congestion control and media access control.

V. Joint Design I: Generating Congestion Price Directly from the MAC Layer

In this section, a primal algorithm is derived by solving the relaxation of the system problem (3)-(4), first proposed in [18]. Based on the algorithm, we propose a traffic-dependent scheme for media access control and generate congestion price directly from the MAC layer.

A. Primal Algorithm and Its Convergence

Instead of solving the system problem (3)-(4), let us consider its relaxation:

$$\max_{x_s \geq 0} \quad V(x) \tag{5}$$

[2]An independent set of a graph is a subset of the vertices such that no two vertices in the subset are adjacent.

[3]A graph is perfect if for every induced subgraph its chromatic number is equal to the clique number of the graph [8].

[4]A hole is a graph induced by a chordless cycle of length at least 4. A hole is odd if it contains an odd number of vertices [7].

with

$$V(x) = \sum_s U_s(x_s) - \sum_n \int_0^{z_n(x)} \lambda_n(v)dv \qquad (6)$$

where $z_n(x) = \sum_{ls} F_{nl} R_{ls} x_s$ is normalized sum rate of clique n for given source rates x, and $\lambda_n(\cdot)$ is the penalty function, which can be interpreted as the price for sending traffic at normalized rate z_n on clique n. We further assume $\lambda_n(\cdot)$ is a non-negative, non-decreasing, continuous function, and not identically zero.

TABLE I
SUMMARY OF MAIN NOTATION

Term	Definition
c_l^0	capacity of link l when active
c_l	effective capacity of link l
y_l	aggregate flow on link l
x_s	source rate of source s
z_n	normalized sum rate of clique n
λ_n	price of clique n
p_l	congestion price of link l
R	routing matrix
F	contention matrix
γ_t, γ	stepsize
Π	feasible rate region

Lemma 1: Under the above assumption, the function $V(x)$ defined in (6) is strictly concave. Thus, the problem (5) admits a unique solution in the interior of the feasible set.

Proof: Let

$$f(x) = \sum_n \int_0^{z_n(x)} \lambda_n(v)dv$$

Since $\lambda_n(\cdot)$ is non-decreasing, for any $x, \bar{x} \geq 0$

$$\begin{aligned} f(x) - f(\bar{x}) &= \sum_n \int_{z_n(\bar{x})}^{z_n(x)} \lambda_n(v)dv \\ &\geq \sum_n \lambda_n(z_n(\bar{x})) \\ &\quad \times \sum_{ls} F_{nl} R_{ls}(x_s - \bar{x}_s) \\ &= \sum_s (x_s - \bar{x}_s) \frac{\partial f}{\partial \bar{x}_s}(\bar{x}) \end{aligned}$$

Thus, according to the first-order condition of convexity for differentiable functions [4], $f(x)$ is a convex function and $-f(x)$ is a concave function. Since $U_s(\cdot)$ is strictly concave, $V(x)$ is the sum of a strictly concave function and a concave function. Thus, $V(x)$ is strictly concave. Note that $V(x) \to -\infty$ as $x_s \to 0$ or as $x_s \to \infty$ for any $s \in S$. So, the problem (5) admits a unique solution that is in the interior of the convex set $x \geq 0$. ∎

The optimal source rates satisfy

$$\frac{\partial V}{\partial x_s} = 0, \quad s \in S$$

which gives

$$U_s'(x_s) - \sum_{nl} \lambda_n(z_n(x)) F_{nl} R_{ls} = 0, \;\; s \in S$$

Define $q_s = \sum_{nl} \lambda_n(z_n) F_{nl} R_{ls}$. Applying the gradient method to (5)–(6), we obtain the following congestion control algorithm

$$\dot{x}_s = \kappa_s \left(U_s'(x_s(t)) - q_s(t) \right), \quad s \in S \qquad (7)$$

where κ_s is a positive. Note that the primal algorithm (7) is completely distributed.

Here, the aggregate normalized price $q_s(t)$ is a feedback signal source s observes. As discussed in [18], $\lambda_n(z_n)$ can be interpreted as a congestion (contention) price that measures the degree of contention in clique n when the total normalized flow through the clique is z_n. Hence, $q_s(t)$ measures the degree of contention in all the cliques that contains any link in source s's path (a larger $q_s(t)$ indicates a greater degree of contention). The congestion control mechanism for each source is to adjust its rate $x_s(t)$ according to the network contention it perceives. In the next subsection, we will design a MAC protocol to generate these 'contention prices' in a distributed manner.

The following theorem, following [18], shows that the primal algorithm (7) is globally stable, i.e., the unique solution to problem (5) is a stable point, to which all trajectories converge.

Theorem 2: Starting from any initial rates $x(0) \geq 0$, the congestion control algorithm (7) will converge to the unique solution of the problem (5).

Proof: From lemma 1, $V(x)$ is a strictly concave function, and problem (5) admits a unique solution x^*. Further

$$\dot{V} = \sum_s \frac{\partial V}{\partial x_s} \dot{x}_s = \sum_s \kappa_s \left(U_s'(x_s) - q_s \right)^2 \geq 0$$

Note that $\dot{V} > 0$ for $x \neq x^*$ and is equal zero for $x = x^*$. Thus, $V(x(t))$ is strictly increasing with t, unless $x(t) = x^*$. More precisely, choose $V(x^*) - V(x)$ as a Lyapunov function for system (7). By Lyapunov's theorem [19], the trajectories of (7) converge to x^*, starting from any initial condition $x(0)$. ∎

Note that algorithm (7) solves the system problem (3)-(4) only approximately. By choosing appropriate price functions $\lambda_n(\cdot)$, the optimal solution can be guaranteed to satisfy the constraint (4), and even solve the system problem (3)-(4) exactly [32]. In practice, the price functions $\lambda_n(\cdot)$ determine the efficiency of the congestion control scheme, as we will further discuss in the next subsection.

B. Generating Congestion Price from the MAC Layer

Unlike the price function in wireline networks which is a function of aggregate flow rate of the link [18], [22], [21], the price function $\lambda_n(\cdot)$ is required to be a function of the normalized sum rate z_n of clique n. This is consistent with the fact that, in wireless networks, link is only a logical concept and the contention region is the "resource" that flows share and contend for access. However, the clique is only a virtual entity and no centralized controller exists to monitor its congestion status, how can we implement the congestion price? We need to design an active queue management scheme where each logical link generates or shares a portion of the congestion price such that their summation is equal to $\lambda_n(z_n)$ for clique n. Observe that a similar problem appears in scheduling flows over ad hoc wireless networks, and that each logical link will get the right portion of the congestion price automatically if the links are granted channel access according to the flow requirements. We propose a multiple access scheme and generate congestion price directly from it.

In multiple access protocols, contention resolution is usually achieved through two mechanisms: persistence and backoff [25]. In the persistence mechanism, each contending node or link-layer flow maintains a persistence probability and contends for the channel with this probability. In the backoff mechanism, each contending node or link-layer flow maintains a backoff window and waits for a random amount of time bounded by the backoff window before a transmission. When multiple simultaneous transmissions cause collisions, the persistence probability or backoff window is adjusted appropriately so that collisions are reduced. Thus, the persistence probability and backoff window are functions of the estimated contention, and different contention resolution algorithms differ in terms of how they adjust these parameters in response to collisions and successful transmissions.

In our problem, the normalized sum rate $z_n = \sum_{ls} F_{nl} R_{ls} x_s$ is the natural measure of the contention in clique n. Thus, the design of multiple access is to adjust persistence probability or/and backoff window according to z_n. The intuition behind this is the same with that behind congestion control algorithm (7), which suggests that we can jointly design congestion control and media access control, and generate congestion price directly from the MAC layer. Note that the normalized flow rate $\sum_s F_{nl} R_{ls} x_s$ is the fraction of time that is required to transmit the amount of flow $y_l = \sum_s R_{ls} x_s$, and the normalized sum rate of a clique must not exceed 1 (see constraint (1)). It has a natural interpretation as a probability. Thus, in our proposed scheme, we approximate the normalized flow rate y_l/c_l^0 as a persistence probability with which the flow l contends for the channel. Furthermore, since each flow l contends for the channel with the probability y_l/c_l^0, the flows should contend for the channel in the same way after they decide to contend, consistent with the fact that the congestion price is a function of the normalized sum rate. This implies that all flows should have the same backoff window.

To be more specific, define $p_l = \min\{\frac{y_l}{c_l^0}, 1\}$, and let w denote the backoff window. The joint design of congestion control and media access control works as follows: each link-layer flow y_l will contend for the channel with probability p_l when it senses the channel is idle. If it decides to contend for the channel, it randomly chooses a waiting time B_l from the interval $[0, w]$ uniformly. After the waiting time, the flow senses the channel and acquires the channel if it is idle. If either the channel is busy or there is collision, the flow will drop or mark the packet as the congestion signal. Upon receiving the congestion signal, the source will adjust its rate according to algorithm (7). We can see that the bandwidth is allocated in proportional to the normalized flow rate of each link. Thus, we obtain a traffic-dependent multiple access scheme.

Note that links needn't know explicitly flow contention graph and the cliques they belong to. But, in order to be consistent with the derivation and convergence analysis of the primal algorithm, the congestion price λ_n of clique n must be a function of the normalized sum rate z_n. Unfortunately, the proposed MAC scheme is very difficult to analyze. For the simple case with no backoff, i.e., $w = 0$, under the assumption of Poisson arrival process, the above scheme does generate approximately the right price function

$$\lambda_n = 1 - e^{-z_n} - z_n e^{-z_n}$$

This price is just the probability when there are two or more packets, and can be readily derived following similar analysis carried out for Aloha [2]. For the general case with backoff, we have not yet obtained an explicit price function.

We can also implement active queue management through designing other kinds of traffic-dependent multiple access schemes. In practice, different designs will give different price functions, which in turn will determine the performance of the congestion control schemes.

VI. Joint Design II: Scheduling Link-Layer Flows According to Congestion Price

In this section, a dual algorithm is derived by solving the dual problem of the system problem (3)-(4)[22], [23]. The solution to the dual problem motivates a scheme for media access control in which link-layer flows are scheduled according to congestion prices.

A. Dual Algorithm and Its Convergence

The system problem (3)-(4) does not involve explicitly the variables for links. We now introduce an auxiliary variable c, which is a L-dimensional vector with each component c_l interpreted as effective or average capacity of link l. Consider the following problem:

$$\max_{x_s \geq 0, c_l \geq 0} \quad \sum_s U_s(x_s) \tag{8}$$

$$\text{subject to} \quad Rx \leq c \;\&\; Fc \leq \varepsilon \tag{9}$$

The first constraint says that the aggregate source rate at any link l does not exceed the effective link capacity. The second constraint says that the effective link capacities satisfy the MAC layer constraint. It is easy to show that this problem is equivalent to the system problem (3)-(4).

Consider the dual problem

$$\min_{p \geq 0} D(p) \tag{10}$$

with partial dual function

$$D(p) = \max_{x_s \geq 0, c_l \geq 0} \quad \sum_s U_s(x_s) - p^T(Rx - c) \tag{11}$$

$$\text{subject to} \quad Fc \leq \varepsilon \tag{12}$$

where we relax only the constraints $Rx \leq c$ by introducing Lagrange multiplier p. The maximization problem in (11) can be decomposed into the following two subproblems

$$D_1(p) = \max_{x_s \geq 0} \sum_s U_s(x_s) - p^T Rx \tag{13}$$

and

$$D_2(p) = \max_{c \geq 0} p^T c \quad \text{subject to} \quad Fc \leq \varepsilon \tag{14}$$

The first subproblem is just TCP [22], [23], and the second one is the scheduling which is to maximize the weighted sum of effective link capacities with the congestion prices as the weights. Thus, by dual decomposition, the flow optimization problem decomposes into separate "local" optimization problems of transport and link layers, respectively, and these two layers interact through the congestion prices.

Note that the objective function $\sum_s U_s(x_s)$ is not strictly concave with respect to variable (x, c), hence the dual function $D(p)$ might not be differentiable. Indeed, the problem (13) admits a unique maximizer

$$x_s(p) = U_s'^{-1}\left(\sum_l p_l R_{ls}\right) \tag{15}$$

and $D_1(p)$ is differentiable, but problem (14) may have multiple maxima and $D_2(p)$ is a piecewise linear function and not differentiable. Thus, $D(p)$ is not differentiable at every point p [3], and we cannot use the usual gradient methods, which are developed for differentiable problems, to solve the dual problem. Here we will solve the dual problem using subgradient method.

Suppose $c(p)$ is a maximizer of the problem (14), i.e.,

$$c(p) \in \arg\max_{c \geq 0} \ p^T c \quad \text{subject to} \quad Fc \leq \varepsilon \tag{16}$$

then

$$g(p) = c(p) - Rx(p) \tag{17}$$

is a subgradient[5] of dual function $D(p)$ at point p. To see this, consider any two points p and $\bar{p}$, by definition

$$\begin{aligned} D(\bar{p}) \;\; = \;\; & \max_{x_s \geq 0, c_l \geq 0} \quad \sum_s U_s(x_s) - \bar{p}^T (Rx - c) \\ & \text{subject to} \quad Fc \leq \varepsilon \end{aligned}$$

hence

$$\begin{aligned} D(\bar{p}) &\geq \sum_s U_s(x_s(p)) - \bar{p}^T (Rx(p) - c(p)) \\ &= D(p) + (\bar{p}^T - p^T)(c(p) - Rx(p)) \end{aligned}$$

Thus, by the subgradient method [3], we obtain the following algorithm for price adjustment for link l

$$p_l(t+1) = [p_l(t) + \gamma_t(\sum_s R_{ls} x_s(p(t)) - c_l(p(t)))]^+ \tag{18}$$

where γ_t is a positive scalar stepsize, and '+' denotes the projection onto the set $\Re^+$ of non-negative real numbers. (15), (16) and (18) are the congestion control algorithm. The algorithm has a nice interpretation in terms of law of supply and demand and their regulation through price. Eq.(18) says that, if the demand $\sum_s R_{ls} x_s(p(t))$ for bandwidth at link l exceeds the supply c_l, the price p_l will rise, which will in turn decrease the demand (see eq. (15)) and increases supply (see eq. (16)). Also, note that equations (15) and (18) are completely distributed. We will study the distributed solution to problem (14) in the next subsection.

Subgradient may not be a direction of descent at point p, but makes an angle less than 90 degrees with all descent directions at p. The new iteration may not improve the dual cost for all values of the stepsize. There exists many results on the convergence of the subgradient method [27], [3]. For constant stepsize, the algorithm is guaranteed to converge to within a range of the optimal value[6]. For diminishing stepsize, the algorithm is guaranteed to converge to the optimal value. For our purposes, we would like an asynchronous implementation of the subgradient algorithm, and thus a constant stepsize is desired. Note that the dual cost will usually not monotonically approach the optimal value, but wander around it under the subgradient algorithm. The usual criterion for stability and convergence is not applicable. Here we define convergence in a statistical sense.

Definition 3: Let p^* denote an optimal value of the dual variable. The algorithm (15), (16) and (18) with constant stepsize is said to converge *statistically* to p^*, if for any given $\delta > 0$ there exists a stepsize γ such that $\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} D(p(\tau)) - D(p^*) \leq \delta$.

The following theorem guarantees the statistical convergence of the subgradient method. Clearly, an optimal value p^* exists.

Theorem 4: Let p^* be an optimal price. Let γ denote the constant stepsize. If the norm of the subgradients is bounded, i.e., there exists G such that $||g(t)||_2 \leq G$ for all t, then the algorithm (15), (16) and (18) converges *statistically* to within $\gamma G^2/2$ of the optimal value.

Proof: By equation (18), we have

$$\begin{aligned} & ||p(t+1) - p^*||_2^2 \\ = \;& ||[p(t) - \gamma g(p(t))]^+ - p^*||_2^2 \\ \leq \;& ||p(t) - \gamma g(p(t)) - p^*||_2^2 \\ = \;& ||p(t) - p^*||_2^2 - 2\gamma g(p(t))^T (p(t) - p^*) \\ & + \gamma^2 ||g(p(t))||_2^2 \\ \leq \;& ||p(t) - p^*||_2^2 - 2\gamma (D(p(t)) - D(p^*)) \\ & + \gamma^2 ||g(p(t))||_2^2 \end{aligned}$$

where the last inequality follows from the definition of subgradient. Applying the inequalities recursively, we obtain

$$\begin{aligned} ||p(t+1) - p^*||_2^2 \;\; \leq \;\; & ||p(1) - p^*||_2^2 - 2\gamma \sum_{\tau=1}^{t} (D(p(\tau)) \\ & - D(p^*)) + \gamma^2 \sum_{\tau=1}^{t} ||g(p(\tau))||_2^2 \end{aligned}$$

Since $||p(t+1) - p^*||_2^2 \geq 0$, we have

$$2\gamma \sum_{\tau=1}^{t} (D(p(\tau)) - D(p^*))$$

[5]Given a convex function $f : \mathcal{R}^n \mapsto \mathcal{R}$, a vector $d \in \mathcal{R}^n$ is a subgradient of f at a point $u \in \mathcal{R}^n$ if $f(v) \geq f(u) + (v - u)^T d, \ v \in \mathcal{R}^n$.

[6]The gradient algorithm with constant stepsize converges to the optimal value, provided the stepsize is small enough.

$$\leq \quad ||p(1) - p^*||_2^2 + \gamma^2 \sum_{\tau=1}^{t} ||g(p(\tau))||_2^2$$
$$\leq \quad ||p(1) - p^*||_2^2 + t\gamma^2 G^2$$

From this inequality we obtain

$$\frac{1}{t}\sum_{\tau=1}^{t} D(p(\tau)) - D(p^*) \leq \frac{||p(1) - p^*||_2^2 + t\gamma^2 G^2}{2t\gamma}$$

Thus

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} D(p(\tau)) - D(p^*) \leq \frac{\gamma G^2}{2} \tag{19}$$

i.e., the algorithm converges statistically to within $\gamma G^2/2$ of the optimal value. ■

The assumption of bounded norm for subgradient $g(p)$ is reasonable, since c is finite and we can also enforce an upper bound to x. We see that, by choosing appropriate value of the stepsize, the algorithm can approach the optimal value arbitrarily close within a finite number of steps.

The system described by equations (15), (16) and (18) is a hybrid system. Although Theorem 4 guarantees that its dynamics is bounded in an average sense, it is unstable in the strict sense. It may have complex behaviors such as limit cycles, i.e., it may go through an ergodic sequence. The reason for this instability is that the dual function is nondifferentiable or nonsmooth. One way to avoid instability is to add some regularization terms, such as strictly convex/concave terms, to make the dual function differentiable. For example, in our problem we can add a concave utility $V_l(c_l)$ to each link l. The resulting system is stable but may not maximize the end user utilities. So, there exists a tradeoff between stability and end user utility maximization (see also [31]). However, in our problem the oscillatory behavior in the "steady state" corresponds to the scheduling process.

B. Scheduling Link-layer Flows according to Congestion Price

Scheduling is to decide which links and when to transmit, which is equivalent to choosing an independent set of flow contention graph to be active at each time slot. However, solving problem (14) cannot guarantee that we obtain a rate vector corresponding to an independent set.

Recall that the reason why constraint (1) may not be a sufficient condition is that it is a fluid level description. However, when the flow vector y is such that each component y_l takes value at 0 or c_l^0 while satisfying constraint (1), it is also feasible. Such a flow vector corresponds to an independent set of flow contention graph. Thus, we propose to replace the constraint in the problem (14) with $Fc \leq \mathbf{1}$, and solve the following scheduling problem with an additional discrete constraint

$$\begin{aligned} \max_{c\geq 0} \quad & p^T c \\ \text{subject to} \quad & Fc \leq \mathbf{1} \\ & c_l = 0 \text{ or } c_l^0, \quad l \in L \end{aligned} \tag{20}$$

Having done that, we need to clarify with respect to which system problem the above algorithm converges. To see this, we first represent an independent set i as a L-dimensional rate vector r^i with

$$r_l^i = \begin{cases} c_l^0 & \text{if } l \in i \\ 0 & \text{otherwise} \end{cases}$$

The feasible rate region Π at the link-layer is then defined to be the convex hull of these rate vector [1]

$$\Pi := \{r : r = \sum_i a_i r^i, a_i \geq 0, \sum_i a_i = 1\}$$

It is easy to verify that solving problem (20) is equivalent to solving the following problem

$$\begin{aligned} \max_{c\geq 0} \quad & p^T c \\ \text{subject to} \quad & c \in \Pi \end{aligned}$$

Thus, the whole joint congestion control and scheduling algorithm is to solve the following system problem

$$\begin{aligned} \max_{x_s\geq 0} \quad & \sum_s U_s(x_s) \\ \text{subject to} \quad & Rx \leq c \;\&\; c \in \Pi \end{aligned}$$

Note that the original problem (8)-(9) is a relaxation to the above problem.

We now come to solve the problem (20). If the contention graph is perfect, all the extreme points of constraint $Fc \leq \mathbf{1}$ are independent sets. In this situation, we can just solve the problem (20) by neglecting the discrete constraint, which has the same optimal solution as the original discrete problem. This is similar to what happens in network flow optimization problems [3]. When the contention graph is not perfect, not all the extreme points of $Fc \leq \mathbf{1}$ are independent sets. In this situation, we will first solve the relaxed problem without discrete constraint, and then round up the solution to the nearest independent set, since the objective function $p^T c$ is continuous with respect to c.

Although the computational complexity of linear programming is polynomial, the known algorithms for general linear programming are not suitable for large scale optimization problems such as those in networks. Instead, an efficient, distributed algorithm with only local information is required for these systems. In our problem, we assume that each link only knows its own weight and the constraints it is involved in. We will again use dual decomposition and subgradient method to obtain a distributed algorithm to solve problem (20). Note that by solving the dual problem we obtain the optimal dual variable, but the optimal primal variable is not immediately available and need to be recovered with care. One simple way to obtain feasible primal solution is to add a small regularization term to the primal function. Here, we add a small quadratic term to the objective function, and maximize

$$p^T c - \delta c^T c$$

where δ is a small positive number. As δ approaches zero, the solution obtained approaches an exact solution to the original problem. This approach is closely related to penalty and augmented Lagrangian methods for solving the dual of a convex program [3].

Consider the dual problem

$$\min_{\lambda \geq 0} L(\lambda) \tag{21}$$

with

$$L(\lambda) = \max_{c \geq 0} \quad p^T c - \delta c^T c - \lambda^T (Fc - 1)$$

The gradient algorithm to the dual problem (21) is

$$c_l(t) = \left[\left(p_l - \sum_n \lambda_n F_{nl}\right) / (2\delta)\right]^+ \tag{22}$$

$$\lambda_n(t+1) = \left[\lambda_n(t) + \beta\left(\sum_l F_{nl} c_l(t) - 1\right)\right]^+ \tag{23}$$

where β is a positive stepsize. The convergence analysis of such algorithms is well-known [3]. Let $\bar{O}$ denote the maximal size of cliques, and $\bar{N}$ the largest number of cliques that contain the same link. The range of the stepsize with which the algorithm converges can be defined as in [22]:

$$0 < \beta < \frac{4\delta}{\bar{O}\bar{N}}$$

After obtaining a value of c_l, link l rounds it up to c_l^0 or 0, whichever is closer. This does not guarantee that the resulting c is optimal or even an independent set all the time, but we can use the notion of ϵ-subgradient[7] to analyze the effect of error [3].

Theorem 5: Suppose at each iteration t a ϵ_t-subgradient is used. Assume that $\epsilon_t \leq \epsilon$ for all t or $\lim_t \epsilon_t \to \epsilon$, then under the same assumptions as in Theorem 4 the algorithm (15), (16) and (18) converges statistically to within $\gamma G^2/2 + \epsilon$ of the optimal value.

Proof: We skip the details, since it is the same as the proof of Theorem 4 except that we use ϵ-subgradient here. ∎

To derive a distributed algorithm for scheduling, we have assumed that each link knows its own constraints. In order to achieve this, each link will collect its local flow information[8], constructs its local contention graph and decomposes it into a set of maximal cliques. Since the clique is only a virtual entity, the price adjustment algorithm (23) for a clique will be carried out by the links within the clique. To be able to calculate new price for a clique, each link needs to exchange new flow rate information, which is calculated by links using algorithm (22), with all its contending flows within one hop. This can be done by periodically broadcasting the flow rate information.

In order for this joint design to work, we require that scheduling be carried out at a much faster time scale than congestion control. Within a time interval γ, the MAC layer should be able to decide which links to transmit and then finish the transmissions. The time scale matching problem is difficult to solve for cross-layer design in general. The key to solving this issue is to be able to design fast, efficient algorithms. For example, in our joint design we can carry out scheduling by heuristically identifying the set of concurrently active links that can achieve the maximization in (14) approximately (see, e.g., [10]).

[7]Given a convex function $f : \mathcal{R}^n \mapsto \mathcal{R}$ and $\epsilon \geq 0$, a vector $d \in \mathcal{R}^n$ is a ϵ-subgradient of f at a point $u \in \mathcal{R}^n$ if $f(v) \geq f(u) - \epsilon + (v-u)^T d$, $v \in \mathcal{R}^n$.

[8]This can be achieved by passively listening to other links broadcasting flow information or actively sending inquiring message to other links to ask for flow information.

C. A Numerical Example

To illustrate the characteristics of the joint congestion control and scheduling algorithm (15), (16) and (18), and their implications for the algorithm's implementation in ad hoc wireless networks, we consider a simple example with the network in Fig. 1. We assume that all the links have the same capacity when active. We further assume $c_l^0 = 1$, $l \in L$, and that all network layer flows s have the same utility $U_s(x_s) = log(x_s)$.

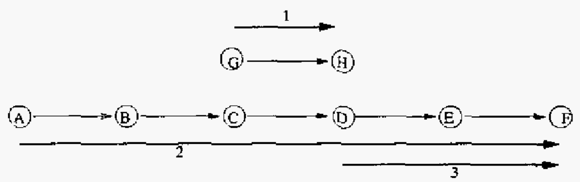

Fig. 4. Ad hoc wireless network with three network layer flows.

Suppose there are three network layer flows $G \to H$, $A \to B$ and $D \to F$ in the network as shown in Fig. 4, with the rates denoted by x_1, x_2 and x_3. We simulate the algorithm (15), (16) and (18) with different choices of stepsize γ. The left panel of Fig. 5 shows the evolution of dual function with the stepsize $\gamma = 0.1$. We can see that the dual function approaches the optimal very fast, but not monotonically. It will oscillate around the optimal. As we have discussed before, this oscillating behavior mathematically results from the non-differentiability of the dual function and physically can be interpreted as corresponding to the scheduling process. The

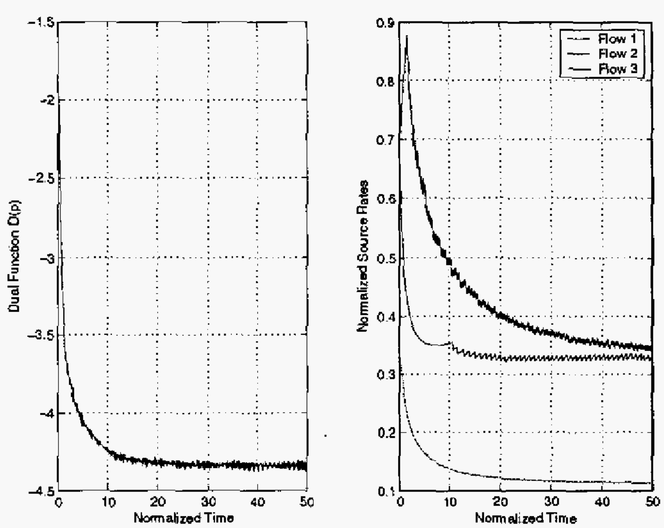

Fig. 5. The evolution of dual function and source rates with stepsize $\gamma = 0.1$. The optimal flow rates are (1/3,1/9,1/3).

right panel of Fig. 5 shows the evolution of source rate of each flow. Similarly, the flow rates approach the primal optimal very fast, but not monotonically. We also note that the performance of the algorithm is much better than the bound $\gamma/2$ specified in Theorem 4. Thus, we can say that, if a protocol is design based on this algorithm, it will likely converge fast.

The choice of the stepsize γ is important. It characterizes the "optimality" of the algorithm, as shown in Theorem 4. Fig. 6 shows the evolutions of the dual function and source rates with the same initial state but different stepsize $\gamma = 0.5$. Compared with the case with stepsize $\gamma = 0.1$, it almost has the same convergence speed, but with a bigger oscillation. Note that, near the primal optimal, the flow rates oscillates between the feasible set and non-feasible set of the constraint (4). The bigger oscillation means that the network will be underloaded and overloaded more often. Thus it will has poorer performance such as lower throughput. So, a smaller stepsize leads to a better performance.

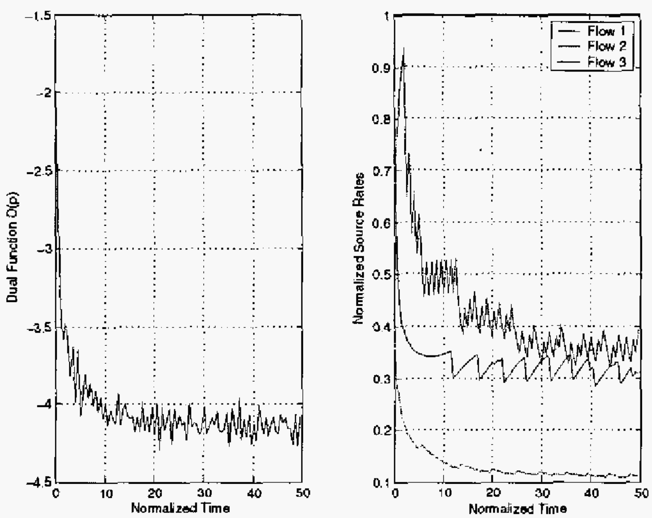

Fig. 6. The evolution of dual function and source rates with stepsize $\gamma = 0.5$. The optimal flow rates are (1/3,1/9,1/3).

However, the stepsize γ also specifies an upper bound for the length of time slot used in the scheduling. As we mentioned before, within time interval γ the MAC layer should decide which links to transmit and then finish the transmissions. So, the stepsize cannot be too small. Thus, there exists a tradeoff between congestion control, which prefers a smaller stepsize, and the scheduling, which prefers a larger stepsize. In practice, the stepsize should take value of order of from ms to tens of ms.

In all the simulations, we use distributed algorithm (22)-(23) to solve the scheduling in (16). To evaluate the performance of our scheduling algorithm, we also use a linear programming software to solve the scheduling. We do not find any distinguishable difference between the simulations using the linear programming software and the algorithm (22)-(23).

Our simulations are based on ideal implementation of the algorithm (15), (16) and (18). In its practical implementation in ad hoc wireless networks, we need to take into consideration such issues as the signaling overhead, the propagation delay, and the time used to make scheduling decision, etc. To design a practical protocol based on this algorithm will be one of our future work.

VII. Conclusion

We have presented a model for the joint design of congestion control and media access control for ad hoc wireless networks, where the resulting algorithms are to solve a utility maximization problem with constraints that arise from contention for the wireless channel. We have derived two algorithms that are not only distributed spatially, but more interestingly, they decompose vertically into two protocol layers where TCP and MAC jointly solve the system problem. The first is a primal algorithm which motivates a joint design where the multiple access scheme is traffic dependent and the congestion prices are generated directly from the MAC layer. The second is a subgradient algorithm for the dual problem and it motivates a joint design where link-layer flows are scheduled according to the congestion prices of the links.

This paper is a preliminary step towards a systematic approach to jointly design TCP congestion control algorithms and MAC algorithms, not only to improve performance, but more importantly, to make their interaction more transparent. Much work remains. First it would be interesting to derive a formal MAC protocol in our joint design I, prove that it generates correct prices, and analyze its dynamic properties. Second, for our joint design II, we will need a faster and more efficient algorithm to solve the scheduling problem if it is to be applied to broadband wireless environment. Third, in cross-layer design through dual decomposition, we often encounter objective functions that are not strictly concave or feasible sets that are not convex. This results in non-differentiable dual function. While subgradient method is applicable to derive a distributed solution, the resulting algorithm is often not stable in the usual sense. This instability that arises from cross-layer interactions need to be understood in order to control cross-layer interactions and to characterize the performance of the design.

Acknowledgments

The authors would like to thank Mung Chiang, Babak Hassibi and Jiantao Wang for helpful discussions, and the anonymous reviewers for helpful comments.

References

[1] A. Bar-Noy, A. Mayer, B. Schieber and M. Sudan. Guaranteeing fair service to persistent dependent tasks. *SIAM J. COMPUT.*, 27(4):1168-1189, August 1998.

[2] D. Bertsekas and R. Gallager, *Data Networks*, 2nd edition, Prentice Hall, 1992.

[3] D. Bertsekas, *Nonlinear Programming*, 2nd edition, Athena scientific, 1999.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[5] K. Chandran, S. Raghunathan, S. Venkatesam and R. Prakash. A feedback-based scheme for improving TCP performance in ad hoc wireless networks, *Proc. IEEE personal Communication Magazine*, 8(1), Febuary 2001.

[6] M. Chiang, To layer or not to layer: balancing transport and physical layers in wirless multihop networks, *Proc. IEEE Infocom*, March 2004.

[7] G. Cornuejols, The strong perect graph theorem, preprint, March 2003.

[8] R. Diestel, *Graph Theory*, Springer-verlag, 1997.

[9] T. D. Dyer and R. V. Boppana, A comparison of TCP performance over three routing protocols for mobile ad hoc networks, *Proc. ACM MobiHoc*, 2001.
[10] T. Elbatt and A. Ephremides, Joint scheduling and power control for wireless ad-hoc networks, *Proc. IEEE Infocom*, 2002.
[11] Z. Fang and B. Bensaou, Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks, *Proc. IEEE Infocom*, March 2004.
[12] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang and M. Gerla, The impact of multihop wireless channel on TCP throughput and loss, *Proc. IEEE Infocom*, March 2003.
[13] M. Gerla, R. Bagrodia, L. Zhang, K. Tang and L. Wang, TCP over wireless multihop protocols: Simulation and experiments, *Proc. IEEE ICC*, June 1999.
[14] B. Hajeck and G. Sasaki, Link scheduling in polynomial time, *IEEE Transactions on Information Theory*, 34(5):901-917, 1988.
[15] G. Holland and N. H. Vaidya, Analysis of TCP performance over mobile ad hoc networks, *Proc. IEEE/ACM Mobicom*, August 1999.
[16] X. L. Huang and B. Bensaou, On max-min fairness and scheduling in wireless ad-hoc networks: Analytical framework and implementation, *Proc. ACM Mobihoc*, 2001.
[17] IEEE, Wirless LAN Media Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Standard 802.11, June 1999.
[18] F. P. Kelly, A. K. Maulloo and D. K. H. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, *Journal of Operations Research Society*, 49(3):237-252, March 1998.
[19] H. Khalil, *Nonlinear Systems*, 2nd edition, Prentice Hall, 1996.
[20] M. Kodialam and T. Nandagopal, Charaterizing achievable rates in multi-hop wireless networks: The joint routing and scheduling problem, *Proc. ACM Mobicom*, September 2003.
[21] S. Kunniyur and R. Srikant, End-to-end congestion control schemes: Utility functions, random losses and ECN marks, *IEEE/ACM Transactions on networking*, 11(5):689-702, October 2003.
[22] S. H. Low and D. E. Lapsley, Optimal flow control, I: Basic algorithm and convergence, *IEEE/ACM Transactions on networking*, 7(6):861-874, December 1999.
[23] S. H. Low, A duality model of TCP and active queue management algorithms, *IEEE/ACM Transactions on Networking*, Octerber 2003.
[24] H. Luo and S. Lu, A topology independent fair queueing model in ad hoc wireless networks, *Proc. IEEE ICNP*, August 2000.
[25] T. Nandagopal, T. E. Kim, X. Gao and V. Bharghhavan, Achieving MAC layer fairness in wireless packet networks, *Proc. ACM Mobicom*, 2000.
[26] S. Shakkottai, T. S. Rappaport and P. C. Karlsson, Cross layer design for wireless networks, *IEEE Communications Magazine*, April 2003.
[27] N. Z. Shor, *Monimization Methods for Non-Differentiable Functions*, Springer-Verlag, 1985.
[28] K. Tang and M. Gerla, Fair sharing of MAC under TCP in wireless ad hoc networks, *Proc. IEEE MMT*, October 2003.
[29] L. Tassiulas and S. Sarkar, Maximin fair scheduling in wireless networks *Proc. IEEE Infocom*, June 2002.
[30] F. Wang and Y. Zhang, Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response, *Proc. ACM MobiHoc*, 2002.
[31] J. Wang, L. Li, S. H. Low and J. C. Doyle, Can TCP and shortest-path routing maximize utility? *Proc. IEEE Infocom*, April 2003.
[32] J. T. Wen and M. Arcak, A unifying passivity framework for network flow control, *Proc. IEEE Infocom*, April 2003.
[33] N. H. Vaidya, P. Bahl and S. Gupta, Distributed fair scheduling in a wireless LAN, *Proc. ACM Mobicom*, 2000.
[34] L. Xiao, M. Johnasson and S. Boyd, Simultaneous routing and resource allocation for wireless networks, *Proc. IEEE Conference on Decision and Control*, 2001.
[35] S. Xu, T. Saadawi, Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks? *IEEE Communications Magazine*, 39(6), June 2001.
[36] K. Xu, S. Bae, S. Lee and M. Gerla, TCP behavior across multihop wireless networks and the wired internet, *ACM WoWMoM*, 2002.
[37] K. Xu, M. Gerla, L. Qi and Y. Shu, Enhance TCP fairness in ad hoc wireless networks using neighborhood RED *Proc. ACM Mobicom*, September 2003.
[38] Y. Yi and S. Shakkottai, Hop-by-hop congestion control over a wireless multi-hop network, *Proc. IEEE Infocom*, March 2004.