Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks

Lijun Chen[†], Steven H. Low[†], Mung Chiang[‡] and John C. Doyle[†]
[†]Engineering & Applied Science Division, California Institute of Technology, USA
[‡]Department of Electrical Engineering, Princeton University, USA

***Abstract*— This paper considers jointly optimal design of cross-layer congestion control, routing and scheduling for ad hoc wireless networks. We first formulate the rate constraint and scheduling constraint using multicommodity flow variables, and formulate resource allocation in networks with fixed wireless channels (or single-rate wireless devices that can mask channel variations) as a utility maximization problem with these constraints. By dual decomposition, the resource allocation problem naturally decomposes into three subproblems: congestion control, routing and scheduling that interact through congestion price. The global convergence property of this algorithm is proved. We next extend the dual algorithm to handle networks with time-varying channels and adaptive multi-rate devices. The stability of the resulting system is established, and its performance is characterized with respect to an ideal reference system which has the best feasible rate region at link layer.**

We then generalize the aforementioned results to a general model of queueing network served by a set of interdependent parallel servers with time-varying service capabilities, which models many design problems in communication networks. We show that for a general convex optimization problem where a subset of variables lie in a polytope and the rest in a convex set, the dual-based algorithm remains stable and optimal when the constraint set is modulated by an irreducible finite-state Markov chain. This paper thus presents a step toward a systematic way to carry out cross-layer design in the framework of "layering as optimization decomposition" for time-varying channel models.

I. Introduction

We consider the problem of congestion control and resource allocation (through routing and scheduling) over a multi-hop wireless ad hoc network. Traditionally, network protocols take a strictly layered structure and implement congestion control, routing and scheduling independently at different layers. However, wireless spectrum is a scarce resource, and it is important to use the wireless channel efficiently. In order to achieve high end-to-end throughput and efficient resource utilization, congestion control, routing and scheduling should be jointly designed while the architectural separation among them is preserved.

The need for joint design across these three layers is motivated by three observations. First, wireless channel is a shared medium and interference-limited. Unlike in wireline networks where links are disjoint resources with fixed capacities, in ad hoc wireless networks the link capacities are "elastic" and the contention among links provide a fundamental constraint for resource allocation (see e.g. [3]), i.e., they determine the feasible rate region at link layer.

Second, most routing schemes for ad hoc networks select paths that minimize hop count (see e.g. [12], [25]). This implicitly predefines a route for any source-destination pair of a static network, independent of the pattern of traffic demand and interference/contention among links. This may result in congestion at some region while other regions are under-utilized. To use the wireless spectrum more efficiently, we should exploit multiple paths based on the pattern of traffic demand and interference/contention among links. As we will see below, routing is then determined from the rate and scheduling constraints.

Lastly, TCP congestion control algorithms can be interpreted as distributed primal-dual algorithms over the Internet to maximize aggregate utility, see e.g. [13], [20], [15]. This series of work implicitly assumes a network where link capacities are fixed and routes are pre-specified. Here, we extend the basic utility maximization formulation with rate constraints at nodes and additional constraints on scheduling at link layer.

We model the contention relations between wireless links as a conflict graph (see e.g. [11]). This construction indicates which groups of links mutually interfere and cannot be active simultaneously. The feasible rate region at link layer is the convex hull of the corresponding rate vectors of independent sets of the conflict graph. We introduce multi-commodity flow variables to formulate rate constraint at the network layer, and formulate resource allocation in wireless ad hoc networks with fixed channel or single-rate devices as a utility maximization problem with those constraints. We then apply duality theory to decompose the system problem vertically into congestion control subproblem and routing/scheduling subproblem that interact through congestion prices. Based on this decomposition, a distributed subgradient algorithm for joint congestion control, routing and scheduling is obtained, and proved to approach arbitrarily close to the optimum of the system problem. This algorithm motivates a joint design where the source adjusts its sending rate according to the congestion price generated locally at the source node, and backpressure from the differential price of neighboring nodes is used for optimal scheduling and routing. We next extend the dual subgradient algorithm to wireless ad hoc networks with time-varying channels and adaptive multi-rate devices. The stability of the resulting system is proved, and its performance is characterized with respect to an ideal reference system.

We then extend the aforementioned results to a generalized model of queueing network that is served by a set of interdependent parallel servers with time-varying service capabilities. This general technique leads to results regarding

the stability and optimality of dual algorithm in face of time-varying parameters, extending most of the earlier publications in this area that assumes deterministic channel models. We show that for a general convex optimization problem where a subset of variables lie in a polytope and the rest in a convex set, the dual-based algorithm remains stable and optimal when the constraint set is modulated by an irreducible finite-state Markov chain.

II. Related work

The work in [13], [15], [20], [21] 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., [4], [36], [35], [3], [18]). In [3], the authors study joint congestion control and media access control for ad hoc wireless network, and formulate rate allocation as a utility maximization problem with the constraints that arise from contention for channel access. This paper substantially extends [3] to include routing and to study the network with time-varying channel and multi-rate devices.

In [22], the authors use multi-commodity flow variables to characterize the network capacity region for a wireless network with time-varying channel, and propose a joint routing and power allocation policy to stabilize the system whenever the input rates are within this capacity region. In [11], the authors study the impact of interference on multi-hop wireless network performance. They model wireless interference using the conflict graph, and show that there is an opportunity for achieving throughput gains by employing an interference-aware routing protocol. We use the same construction to model the contention relations among wireless links. In [7], [14], the authors use a similar model to study the problem of jointly routing the flows and scheduling the transmissions to determine the achievable rates in multi-hop wireless networks. All these works focus on the interaction between link and network layers, and try to characterize the achievable rate region at network layer. We include the end-to-end transport layer, and as such, the network uses congestion control to automatically explore the achievable rate region while optimizing some global objective for the end users.

The stochastic Lyapunov function method is a powerful tool to prove the stability of Markovian system [1], [29]. Especially, Theorem 3.1 in [29] provides sufficient conditions for the stability of general Markov chain. We combine convex analysis with stochastic Lyapunov method to establish the stability and optimality properties of networks with time-varying channels. Our result is applicable to a variety of time-varying systems that can be solved or modelled by dual algorithms. Similar result is obtained in other contexts through different techniques [28], [6].

Our goal is to present a systematic approach to cross-layer design, not only to improve the performance, but more importantly, to make the interactions between different layers more transparent. Motivated by the duality model of TCP/AQM, which is an example of "horizontal" decomposition via dual decomposition, researchers have extended the utility maximization framework to provide a general cross-layer design methodology. As we will see in this paper, duality theory leads to a natural "vertical" decomposition into separate designs of different layers that interact through congestion price. Recent publications along this line of "layering as optimization decomposition" [5] includes [31], [8] for TCP/IP interaction, [34] for routing and resource allocation, [4], [16] for TCP and physical layer, and [3], [17], [18], [32] for joint TCP and media access control or scheduling.

III. Model

Consider an ad hoc wireless network with a set N of nodes and a set L of logical links. These links are directed, though we assume connectivity to be symmetric, i.e., link $(j,i) \in L$ if and only if $(i,j) \in L$. We assume a static topology and each link l has a fixed finite capacity c_l bits per second when active, i.e., we implicitly assume that the wireless channel is fixed or some underlying mechanism is used to mask the channel variation so that the wireless channel appears to have a fixed rate. This assumption will be relaxed in Section V. Wireless channel is a shared medium and interference-limited where links contend with each other for exclusive access to the channel. We will use the conflict graph to capture the contention relations among links. The feasible rate region at link layer is then a convex hull of the corresponding rate vectors of independent sets of the conflict graph. We will further introduce multi-commodity flow variables, which correspond to the link capacities allocated to the flows towards different destinations, to describe the rate constraint at network layer. The resource allocation is then formulated as a utility maximization problem with schedulability and rate constraints.

A. Schedulability and Rate Constraint

In this paper, we consider a network with primary interference: links that share a common node cannot transmit or receive simultaneously, but links that do not share nodes can do so. The same interference model has been used in e.g. [14], [36]. It models a wireless network with multiple channels available for transmission. For example, simultaneous communications in a neighborhood are enabled by using orthogonal CDMA or FDMA channels. Under this interference model, we can construct a conflict graph [11] that captures the contention relations among the links. In the conflict graph, each vertex represents a link, and an edge between two vertices denotes the contention between the two corresponding links: these links cannot transmit at the same time. Fig.1 shows an example of a wireless ad hoc network and its conflict graph.

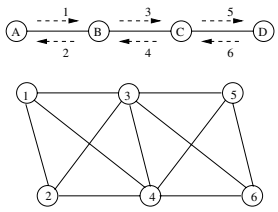

Fig. 1. Example of an ad hoc wireless network with 4 nodes and 6 logical links and the corresponding conflict graph

Given a conflict graph, we can identify all its independent sets of vertices[1]. The links in an independent set can transmit simultaneously. Let E denote the set of independent sets with each independent set indexed by e. We represent an independent set e as a $|L|$-dimensional rate vector r^e, where the lth entry is

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

The feasible rate region Π at the link layer is then defined as the convex hull of these rate vectors

$$\Pi := \left\{ r : r = \sum_e a_e r^e,\ a_e \geq 0,\ \sum_e a_e = 1 \right\} \tag{1}$$

Thus, given a link flow vector y, the schedulability constraint says that y should satisfy $y \in \Pi$.

Let D denote the set of destination nodes of network layer flows. Let $f_{i,j}^k \geq 0$ denote the amount of capacity of link (i,j) allocated to the flow to destination k. Then $f_{i,j} := \sum_{k \in D} f_{i,j}^k$ is the aggregate capacity on link (i,j). From the schedulability constraint, $f := \{f_{i,j}\}$ should satisfy

$$f \in \Pi \tag{2}$$

Let $x_i^k \geq 0$ denote the flow generated at node i towards destination k. Then the aggregate capacity for its incoming flows and generated flow to the destination k should not exceed the summation of the capacities for its outgoing flows to k:

$$x_i^k \leq \sum_{j:(i,j)\in L} f_{i,j}^k - \sum_{j:(j,i)\in L} f_{j,i}^k,\ i \in N,\ k \in D,\ i \neq k \tag{3}$$

Equation (3) is the rate constraint for resource allocation. It is similar to multicommodity flow model for the routing of data flows in the network, but we give multi-commodity flow variables a different interpretation as the amount of link capacities allocated to the flows of different destinations.

B. Problem Formulation

We use $l \in L$ or alternatively node pair $(i,j) \in N \times N$ to denote a link. We also stack up the entries of any tensor $t_{i,j}$ (or t_i^j) to construct a vector, denoted by $\{t_{i,j}\}$ (or $\{t_i^j\}$) or just t. Assume the network is shared by a set S of sources indexed by s. For notational simplicity, we assume that there is at most one flow between any node and destination pair $[i,k]$[2], and use s or alternatively node pair $[i,k] \in S \times D$ to denote a network layer flow.

Assume 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, and strictly concave. Our objective is to choose source rates x_s and allocated capacities $f_{i,j}^k$ so as to solve the following global problem:

$$\max_{x_s \geq 0, f_{i,j}^k \geq 0} \quad \sum_s U_s(x_s) \tag{4}$$

$$\text{subject to} \quad x_i^k \leq \sum_{j:(i,j)\in L} f_{i,j}^k - \sum_{j:(j,i)\in L} f_{j,i}^k \tag{5}$$

$$f \in \Pi \tag{6}$$

[1]An independent set of vertices is a set of vertices that has no edges between each other.

[2]The extension to the situation with multiple flows between any node-destination pair is straightforward.

where $i \in N,\ k \in D,\ i \neq k$, and $x_i^k = 0$ if $[i,k] \notin S \times D$.

Solving the system problem (4)-(6) directly requires coordination among possibly all sources and links, thus is impractical in real network. Since (4) is a convex optimization problem with strong duality, distributed algorithms can be derived by formulating and solving its Lagrange dual problem. In the next section, we will solve the dual problem and interpret the resulting algorithm in the context of joint design of congestion control, routing and scheduling.

IV. Cross-layer design via dual decomposition

A. Dual Algorithm

Consider the dual to the primal problem (4,5,6):

$$\min_{p \geq 0} D(p) \tag{7}$$

with partial dual function

$$D(p) = \max_{x_s \geq 0, f_{i,j}^k \geq 0} \quad \sum_s U_s(x_s) - \sum_{i \in N, k \in D, i \neq k} p_i^k (x_i^k - \sum_{j:(i,j)\in L} f_{i,j}^k + \sum_{j:(j,i)\in L} f_{j,i}^k) \tag{8}$$

$$\text{subject to} \quad f \in \Pi \tag{9}$$

where we relax only the constraint (5) by introducing Lagrange multiplier p_i^k for node $i \in N$ and destination $k \in D$. The maximization problem in (8) can be decomposed into the following two subproblems

$$D_1(p) = \max_{x_s \geq 0} \sum_s U_s(x_s) - \sum_s x_s p_s \tag{10}$$

and

$$D_2(p) = \max_{f_{i,j}^k \geq 0} \quad \sum_{i,k} p_i^k \left(\sum_j f_{i,j}^k - \sum_j f_{j,i}^k \right) \tag{11}$$

$$\text{subject to} \quad f \in \Pi \tag{12}$$

where we use p_s to denote the multiplier p_i^k if $[i,k] \in S \times D$. If we interpret p_i^k as the congestion price, the first subproblem is congestion control [20], [21], and the second one is the joint routing and scheduling since to solve it we need to determine the amount of capacity $f_{i,j}^k$ that link (i,j) is allocated to transmit the data flow towards destination k. Thus, by dual decomposition, the flow optimization problem decomposes into separate "local" optimization problems of transport and network/link layers, respectively, and they interact through congestion prices.

The congestion control problem (10) admits a unique maximizer

$$x_s(p) = {U_s'}^{-1}(p_s) \tag{13}$$

which adjusts the source rate according to the congestion price of the source node. In contrast to traditional TCP congestion control where the source adjusts its sending rate according to the aggregate price along its path, in our algorithm the congestion price is generated locally at the source node.

Note that, since

$$\sum_{i,k} p_i^k \left(\sum_j f_{i,j}^k - \sum_j f_{j,i}^k \right) = \sum_{i,j,k} f_{i,j}^k \left(p_i^k - p_j^k \right),$$

problem (11)-(12) is equivalent to the following problem

$$D_2(p) = \max_{f_{i,j}\geq 0} \quad \sum_{i,j} f_{i,j} \max_k \left(p_i^k - p_j^k\right) \tag{14}$$

$$\text{subject to} \quad f \in \Pi \tag{15}$$

This motivates the following joint scheduling and routing algorithm:

1) For each link (i,j), find destination k^* such that $k^* \in \arg\max_k (p_i^k - p_j^k)$, and define $w_{i,j} = p_i^{k^*} - p_j^{k^*}$.

2) Scheduling: choose $\widetilde{f}_{i,j}$ such that

$$\widetilde{f} \in \arg\max_{f\in\Pi} \sum_{(i,j)\in L} w_{i,j} f_{i,j} \tag{16}$$

There may exist multiple maximizers, but we always pick an extreme point maximizer[3]. An extreme point maximizer corresponds to a maximal independent set of the flow contention graph. The scheduling (16) is a difficult problem for ad hoc wireless network. We will discuss its solution in detail in Subsection IV-C.

3) Routing: over link (i,j), send an amount of bits for destination k^* according to the rate determined by the above scheduling.

The $w_{i,j}$ values represent the maximum *differential* congestion price of destination k packets between nodes i and j. The above algorithm uses *back-pressure* to do optimal scheduling and find optimal routing[4]. Note that 1)–3) is equivalent to solve the problem (11)-(12) by the following assignment

$$f_{i,j}^k = \begin{cases} \widetilde{f}_{i,j} & \text{if } k = k^* \\ 0 & \text{if } k \neq k^* \end{cases}$$

Now we come to solve the dual problem (7). Note that the dual function $D(p)$ is not differentiable, as $D_2(p)$ is a piecewise linear function and not differentiable. Therefore, we cannot use the usual gradient methods, we will instead solve the dual problem using subgradient method.

Suppose $(f_{i,j}^k)$ is the solution from the above joint routing and scheduling algorithm. It is easy to verify that

$$g_i^k(p) = \sum_j f_{i,j}^k(p) - \sum_j f_{j,i}^k(p) - x_i^k(p) \tag{17}$$

is a subgradient[5] of dual function $D(p)$ at point p. Thus, by the subgradient method [26], [2], we obtain the following algorithm for price adjustment for node destination pair (i,k)

$$\begin{aligned} p_i^k(t+1) \quad = \quad & [p_i^k(t) + \gamma_t(\ x_i^k(p(t)) \\ & -(\sum_{j:(i,j)\in L} f_{i,j}^k(p(t)) - \sum_{j:(j,i)\in L} f_{j,i}^k(p(t))\ )\ )]^+ \end{aligned} \tag{18}$$

where γ_t is a positive scalar stepsize, and '+' denotes the projection onto the set $\Re^+$ of non-negative real numbers. Eq.(18) says that, if the demand $x_i^k(p(t))$ for bandwidth at node i for the flow to destination k exceeds the effective capacity $\sum_j f_{i,j}^k - \sum_j f_{j,i}^k$, the price p_i^k will rise, which will in turn decrease the demand (see eq.(13)) and increases effective capacity (see eq.(14)). Also, note that eq.(18) is distributed and can be implemented by individual nodes using only local information.

The above dual algorithm motivates a joint congestion control, routing and scheduling design where at the transport layer sources s individually adjust their rates according to the local congestion price, and nodes i individually update their prices according to (18); and at the network/link layer nodes i solve the scheduling (16) and route data flows accordingly. In summary, we have the following

Algorithm 1: Joint Design Algorithm

At time t:

1) Each node i implicitly updates its price with respect to destination k

$$\begin{aligned} p_i^k(t+1) \quad = \quad & [p_i^k(t) + \gamma_t(\ x_i^k(p(t)) \\ & -(\sum_{j:(i,j)\in L} f_{i,j}^k(p(t)) - \sum_{j:(j,i)\in L} f_{j,i}^k(p(t))\ )\ )]^+, \end{aligned}$$

and passes the price p_i^k to all its neighbors. Note that $p_i^k(t)$ is interpreted as the congestion price at the beginning of time slot t.

2) Congestion control: each source node s adjusts its sending rate for the period t, according to local congestion price

$$x_s(t) = {U_s'}^{-1}(p_s(t))$$

3) Each node i collects congestion price information from its neighbor j, finds destination $k(t)$ such that $k(t) \in \arg\max_k (p_i^k(t) - p_j^k(t))$, and calculates differential price $w_{i,j}(t) = p_i^{k(t)}(t) - p_j^{k(t)}(t)$ and passes this information to its neighbors.

4) Scheduling: each node i collects differential price information from its neighbors in the previous period, and in the beginning of period t allocates a capacity $\widetilde{f}_{i,j}(t)$ over link (i,j) such that

$$\widetilde{f}(t) \in \arg\max_{f\in\Pi} \sum_{(i,j)\in L} w_{i,j}(t) f_{i,j}$$

5) Routing: over link (i,j), send an amount of bits for destination $k(t)$ according to the rate determined by the scheduling.

B. Convergence Analysis

In this subsection, we prove the convergence property of Algorithm 1. Subgradient may not be a direction of descent, but makes an angle less than 90 degrees with all descent directions. Thus, the new iteration may not improve the dual cost for all values of the stepsize. Using results on the convergence of the subgradient method [26], [2], we show that, for constant stepsize, the algorithm is guaranteed to converge to within a neighborhood of the optimal value. For diminishing stepsize, the algorithm is guaranteed to converge to the optimal value. We would like a distributed implementation of the subgradient algorithm, and thus a constant stepsize $\gamma_t = \gamma$ is more convenient. Note that the dual cost usually will not

[3]A point in a convex set is an extreme point if it cannot be written as a convex combination of other points in the convex set.

[4]The above joint routing and scheduling recovers the DRPC policy in reference [22], except that step 2 is scheduling here and power allocation there, and data is routed based on destination here and "commodity" there. We show that the DRPC policy follows mathematically from dual decomposition. Similar decomposition result for the network with deterministic wireless channel is also revealed in the journal version of [22] and [18].

[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 $f(v) \geq f(u) + (v-u)^T d,\ \forall v \in \mathcal{R}^n$.

monotonically approach the optimal value, but wander around it under the subgradient algorithm. The usual criterion for stability and convergence is not applicable. We will use a similar definition of convergence as in [3]. Let $\overline{p}(t) := \frac{1}{t}\sum_{\tau=1}^{t} p(\tau)$ be the average price by time t.

Definition 1: Let p^* denote an optimal value of the dual variable. Algorithm 1 with constant stepsize is said to converge *statistically* to p^*, if for any $\delta > 0$ there exists a stepsize γ such that $\limsup_{t\to\infty} D(\overline{p}(t)) - D(p^*) \leq \delta$.

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

Theorem 2: Let p^* be an optimal price. If the norm of the subgradients is uniformly bounded, i.e., there exists G such that $||g(p)||_2 \leq G$ for all p, then Algorithm 1 converges statistically to p^*.

The assumption of bounded norm for subgradient $g(p)$ is reasonable, since f is finite and we always have an upper bound on x in practice. Note that $D(p) \geq D(p^*)$ always holds. Since $D(p)$ is a continuous function, Theorem 2 implies that the congestion price p approaches p^* statistically when the stepsize γ is small enough.

Let the primal function (the total achieved network utility) be $P(x)$ and achieve its optimum at x^*. Define $\overline{x}(t) := \frac{1}{t}\sum_{\tau=0}^{t} x(\tau)$, the average data rate up to time t. As time goes to infinity, $\overline{x}(t)$ must be in the feasible rate region (determined by eqs. (5)-(6)), otherwise $\overline{p}(t)$ will go unbounded as time goes to infinity, which contradicts Theorem 2.

Theorem 3: Let x^* be the optimal source rates. Under the same assumption of Theorem 2, the following inequality holds

$$\liminf_{t\to\infty} P(\overline{x}(t)) \geq P(x^*) - \frac{\gamma G^2}{2}. \tag{19}$$

Note that $P(x) \leq P(x^*)$ holds for any x in the feasible rate region. Since $P(x)$ is continuous, Theorem 3 implies that the average source rate approaches the optimal x^* when γ is small enough.

C. Scheduling over Ad Hoc Networks

We now come to the scheduling problem (16). Scheduling over ad hoc network is a difficult problem and in general NP-hard. To see this, note that problem (16) is equivalent to a maximum weight independent set problem over the conflict graph, which is NP-hard for general graphs. However, with the primary interference model we show that problem (16) can be reduced to the maximum weighted matching problem[6], which is polynomial time solvable. As one of the extensions in Subsection VI-A, we will see a NP-hard scheduling problem.

The scheduling problem (16) is to maximize the weighted sum of the link capacities with the schedulability constraint. It is defined on a weighted digraph whose link weights $w_{i,j}$ can take negative value. To see how it is related to the maximum weighted matching problem, first note that $w_{j,i} > 0$ if $w_{i,j} < 0$ and vice versa. Second, note that links (i,j) and (j,i) mutually interfere and have the same interference/contention relations with other links. Corresponding to each directed link pair $(i,j),(j,i) \in L$, define an undirected link $\langle i,j\rangle$ with weight

$$w'_{i,j} = \max\{w_{i,j}c_{i,j}, w_{j,i}c_{j,i}\}.$$

Let L' denote the set of undirected links and W' the corresponding set of weights, the scheduling problem (16) is then equivalent to the maximum weighted matching problem on the weighted graph $G' = (N, L', W')$. Note that an (maximal) independent set in the conflict graph will correspond to a (maximal) matching in this undirected graph.

Maximum weighted matching problem can be computed in polynomial time (see, e.g., [23]), but this requires centralized implementation. If implemented over an ad hoc network, each node needs to notify the central node of its weight and local connectivity information such that the central node can reconstruct the network topology as a weighted graph. This will lead to an immense communication overhead which is expensive in time and resources. There also exist simpler greedy sequential algorithms to compute a weighted matching at most a factor of 2 away from the maximum (see. e.g., [24]). But they also require centralized implementation. We seek a distributed algorithm where each node participates in the computation itself using only local information.

A few distributed approximation algorithms exist for maximum weighted matching problem, see e.g. [30], [33], [9]. In [9], the author presents a simple distributed algorithm to compute a weighted matching at most a factor of 2 away from the maximum in linear running time $O(|L'|)$. This algorithm is a distributed variant of the sequential greedy algorithm presented in [24]. We utilize this algorithm to solve our scheduling problem (16) distributedly, as summarized below.

Algorithm 2: Distributed Scheduling Algorithm

Each node i carries out the following steps:

1) Calculate weight $w'_{i,j} = \max\{w_{i,j}c_{i,j}, w_{j,i}c_{j,i}\}$ for each directed link pair $(i,j),(j,i) \in L$ incident upon it. Ties are broken randomly.

2) Find node j^* such that w'_{i,j^*} is maximized over all links $\langle i,j\rangle \in L'$ with free neighbors j:

–If having received a *matching* request from j^*, then link $\langle i,j^*\rangle$ is a matched link. Node i sends a *matched* reply to j^* and a *drop* message to all other free neighbors.

–Otherwise, node i sends a *matching* request to node j^*.

3) Upon receiving a *matching* request from neighbor j:

–If $j = j^*$, then link $\langle i,j\rangle$ is a matched link. Node i sends a *matched* reply to node j and a *drop* message to all other free neighbors.

–If $j \neq j^*$, node i just stores the received message.

4) Upon receiving a *matched* reply from neighbor j, node i knows link $\langle i,j\rangle$ is a matched link, and send a *drop* message to all other free neighbors.

5) Upon receiving a *drop* message from neighbor j, node i knows that j is in a matched link, and excludes j from its free neighbors set.

6) If node i is in a matched link or has no free neighbors, no further action is taken. Otherwise, it will repeat steps 2)–5).

7) Matched links are allowed to transmit. Nodes i,j in a matched link $\langle i,j\rangle$ will schedule the directed link, which gives

[6]A matching in a graph is a subset of links, no two of which share a common node. The weight of a matching is the total weight of all its links. A maximum weighted matching in a graph is a matching whose weight is maximized over all matchings of the graph.

value $w'_{i,j}$, to transmit.

Steps 2)–6) is the distributed algorithm for maximum weighted matching problem. A link that has been chosen to be in the matching is called matched link. Nodes that are not incident upon any matched link are called free. A *matching* request is sent to inquire the possibility to choose the link with a neighbor as a matched link. A *matched* reply is sent to confirm that the link with a neighbor is matched. A node sends *drop* message to tell its neighbors that it is not free anymore. Define a link $\langle i, j\rangle$ to be locally heaviest link if for both i and j, its weight is maximized over all links with free neighbors. We can see that this algorithm selects locally heaviest links as matched. Thus, Algorithm 2 is a locally optimal scheduling.

Comparing to the known results in the literature, the above distributed scheduling algorithm for ad hoc wireless network is one of the best distributed algorithms in terms of computational complexity and performance bound. It has a linear complexity $O(|L'|)$. Such a low complexity is important for the scalability and efficiency of ad hoc wireless network. It achieves a performance of $1/2$ of the maximum weight in the worst case and, in practice, our numerical simulations show it typically achieves a performance within about $4/5$ of the maximum weight.

As for the overall performance of our cross-layer design with this approximate scheduling, we can extend the results in [19] to show that the performance is no worse than that achieved by an exact design with a feasible rate region $\frac{1}{2}\Pi$ (and in practice, $\frac{4}{5}\Pi$) at the link layer. Moreover, in Section VII we will see that this distributed scheduling algorithm only results in a very small degradation in the performance of the cross-layer design for the network with time-varying channel, since in this situation the exact solution of the scheduling is not as important and reasonable approximations work well.

D. Implementation Issues

Utility function and global parameter: The utility function is determined by the objective of the end user such as fairness requirement. The smaller the global parameter γ, the closer does the algorithm converge to the optimal point. Its value can be chosen guided by simulations.

Congestion price and queueing: A natural choice of congestion price is queue length. Each node does not need to keep per flow information but distinguishes flows by their destinations. Therefore, each node should manage separate queues for flows going to different destination nodes.

Message passing and communication overhead: Each node needs to communicate its congestion price information to its neighbors. This can be achieved by periodically broadcasting this information to its neighbors or its neighbors can actively send inquiring message to ask for this information.

We now examine the implications of our design to the layered and distributed network architecture. Our congestion control is not an end-to-end scheme. Each source node adjusts its sending rate according to the local congestion price. Thus, there is no communication overhead for congestion control. This is very different from the end-to-end congestion control where the "global" aggregate congestion price along the whole path needs to be fed back to the source node. Also, there is no communication overhead for routing, since we basically get routing for free from the scheduling. The majority of communication overhead is for scheduling. Let K denote the maximum degree of nodes in the network, the communication overhead for scheduling is $O(K|L|)$ per node per time slot. Thus, our design has a low communication overhead.

V. Joint Design in Networks with Time-varying Channels

In the last section, we consider the joint congestion control, routing and scheduling design for wireless ad hoc networks with fixed channels or single-rate devices, i.e., the capacity c_l is a constant when link l is active. However, recent years have seen the growing popularity and demand of multi-rate wireless network devices (e.g., 802.11a cards) that can adjust transmission rate according to the time-varying channel state and improve overall network utility. Here, we consider a jointly optimal layers 2-3-4 design over networks with time-varying channels and adaptive multi-rate devices.

A. Algorithm and Convergence Analysis

We assume that time is slotted, and the channels are fixed within a time slot but independently change between different slots[7]. Let $h(t)$ denote the channel state in time slot t. Corresponding to the channel state h, the capacity of link l is $c_l(h)$ when active and the feasible rate region at the link layer is $\Pi(h)$, which is defined in a similar way as in (1). We further assume that the channel state is a finite state process with identical distribution $q(h)$ in each time slot[8], and define the mean feasible rate region as

$$\overline{\Pi} := \{\overline{r} : \overline{r} = \sum_h q(h)r(h), r(h) \in \Pi(h)\} \tag{20}$$

Ideally, we would like to have a joint design of congestion control, routing and scheduling, which solves the following utility maximization problem

$$\max_{x_s \geq 0, f^k_{i,j} \geq 0} \quad \sum_s U_s(x_s) \tag{21}$$

$$\text{subject to} \quad x^k_i \leq \sum_{j:(i,j)\in L} f^k_{i,j} - \sum_{j:(j,i)\in L} f^k_{j,i}, \quad i \in N,\ k \in D,\ i \neq k \tag{22}$$

$$f \in \overline{\Pi} \tag{23}$$

However, if we solve the above problem via dual decomposition, we may get a link rate assignment which is infeasible for the channel state at a given time slot. Instead we directly extend Algorithm 1 with a modification to handle time-varying channel. For convenience, we describe the algorithm in details in the following:

Algorithm 3: Joint Design Algorithm over Networks with Time-varying Channels

[7]It is straightforward to extend our results to the network where the channel state process is modulated by a hidden Markov chain.

[8]Even if the channel state is a continuous process, we only have finite choices of modulation schemes. The corresponding capacities take discrete values.

At time t:

1) Each node i updates its price with respect to destination k

$$p_i^k(t+1) = \lfloor [p_i^k(t) + \gamma(x_i^k(p(t)) - (\sum_{j:(i,j)\in L} f_{i,j}^k(p(t)) - \sum_{j:(j,i)\in L} f_{j,i}^k(p(t))))]^+ \rfloor, \quad (24)$$

and passes the price p_i^k to all its neighbors. Here $\lfloor \rfloor$ denotes the integer function floor, and for the simplicity of the presentation we let congestion price takes integer values with appropriate unit.

2) Congestion control: each source node s adjusts its sending rate for the period t, according to local congestion price

$$x_s(t) = U_s'^{-1}(p_s(t)).$$

3) Each node i collects congestion price information from its neighbor j, find destination $k(t)$ such that $k(t) \in \arg\max_k(p_i^k(t) - p_j^k(t))$, and calculate differential price $w_{i,j}(t) = p_i^{k(t)}(t) - p_j^{k(t)}(t)$ and passes this information to its neighbors.

4) Scheduling: after collecting differential price information from its neighbors in the previous period, in the beginning of period t each node i monitors the channel state $h(t)$ and allocates a capacity $\widetilde{f}_{i,j}(t)$ over link (i,j) such that

$$\widetilde{f}(t) \in \arg\max_{f\in\Pi(h(t))} \sum_{(i,j)\in L} w_{i,j}(t) f_{i,j} \ . \quad (25)$$

Again we will always pick an extreme-point maximizer in the above scheduling. Note that, although we assume that channel state has a stationary distribution, the nodes do not need to know this statistics but only the current channel state.

5) Routing: over link (i,j), send an amount of bits for destination $k(t)$ according to the rate determined by the scheduling.

The above algorithm for joint design cannot be derived from the dual decomposition of the problem (21)-(23). However, we will use the problem (21)-(23) as a reference system, and characterize the performance of the above algorithm with respect to it.

Note that congestion price $p_i^k(t)$ is proportional to the queue length at node i for the flows to destination d. It takes discrete values, i.e., the queue length scaled by γ. Thus, congestion price $p(t)$ evolves according to a discrete-time, discrete-space Markov chain. We need to show that this markov chain is stable, i.e., the congestion price process reaches a steady state and does not become unbounded. It is easy to check that the Markov chain has a countable state space, but is not necessarily irreducible. In such a general case, the state space is partitioned in transient set T and different recurrent classes R_i. We define the system to be *stable* if all recurrent states are positive recurrent and the Markov process hits the recurrent states with probability one [29]. This will guarantee that the Markov chain will be absorbed/reduced into some recurrent class, and the positive recurrence ensures the ergodicity of the Markov chain over this class. We have the following

Theorem 4: The Markov chain described by equation (24) is stable.

Proof: Denote the dual function of the problem (21)-(23) by $\overline{D}(p)$ with an optimal price p^* and subgradient $\overline{g}(p)$. Consider the the Lyapunov function $V(p) = \|p - p^*\|_2^2$, we have

$$\begin{aligned} & E[\Delta V_t(p)|p] \\ = \ & E[V(p(t+1)) - V(p(t)) \mid p(t) = p] \quad (26) \\ = \ & E[V(\lfloor [p(t) - \gamma g(p(t))]^+ \rfloor) - V(p(t)) \mid p(t) = p] \\ \le \ & E[V(p(t) - \gamma g(p(t))) - V(p(t)) \mid p(t) = p] \\ = \ & E[-\gamma g(p(t))^T (2(p(t) - p^*) - \gamma g(p(t))) \mid p(t) = p] \\ = \ & 2\gamma \overline{g}(p)^T (p^* - p) + \gamma^2 E[\|g(p(t))\|_2^2 \mid p(t) = p] \\ \le \ & 2\gamma \overline{g}(p)^T (p^* - p) + \gamma^2 G^2 \end{aligned}$$

where we again use the assumption that the norm of $g(p(t))$ is bounded above by G. Since $\overline{D}(p)$ is a convex function, we further get

$$E[\Delta V_t(p)|p] \le 2\gamma(\overline{D}(p^*) - \overline{D}(p)) + \gamma^2 G^2$$

Let

$$\delta = \max_{\overline{D}(p) - \overline{D}(p^*) \le \gamma G^2} \|p - p^*\|_2$$

and define $\mathcal{A} = \{p : \ \|p - p^*\|_2 \le \delta\}$. We obtain

$$E[\Delta V_t(p)|p] \le -\gamma^2 G^2 \mathcal{I}_{p\in\mathcal{A}^c} + \gamma^2 G^2 \mathcal{I}_{p\in\mathcal{A}}$$

where $\mathcal{I}$ is the index function. Thus, by Theorem 3.1 in [29], which is an extension of Foster's criterion [1], the Markov chain $p(t)$ is stable. ■

The above proof shows that the distance to the optimal p^* has negative conditional mean drift for all prices that have sufficiently large distance to p^*, and implies that the congestion price will stay near p^* when γ is small enough.

B. Performance Evaluation

We now characterize the performance of the joint design in terms of the dual and primal objective functions.

Theorem 5: Algorithm 3 converges statistically to within $\gamma G^2/2$ of the optimal value $\overline{D}(p^*)$, i.e.,

$$\overline{D}(E[p(\infty)]) - \overline{D}(p^*) \le \gamma G^2/2, \quad (27)$$

where $p(\infty)$ denotes the state of the Markov chain in steady state.

Note that $\overline{D}(p) \ge \overline{D}(p^*)$ always holds. Since $\overline{D}(p)$ is a continuous function, Theorem 5 implies that the congestion price p approaches p^* statistically when stepsize γ is small enough.

Theorem 6: The source rates $x(t)$ is a stable Markov chain. Moreover, let $\overline{P}(x)$ be the primal function and x^* be the optimal source rates of the system problem (21)-(23), we have the following inequality

$$\overline{P}(E[x(\infty)]) \ge \overline{P}(x^*) - \frac{\gamma G^2}{2}, \quad (28)$$

where $x(\infty)$ denotes the state of the Markov chain $x(t)$ in the steady state.

Similarly, $E[x(\infty)]$ is the average data rate and must be in the feasible rate region (determined by eqs. (22)-(23)), otherwise the average queue length $E[p(\infty)]$ will go unbounded. Thus, Theorems 6 implies that the average source

rate approaches the optimal of the ideal reference system (21)-(23) when stepsize γ is small enough. Theorems 5 and 6 show that, surprisingly, the joint congestion control, routing and scheduling in Algorithm 3 can be seen as a distributed algorithm to approximately solve the ideal reference system problem that is not readily solvable due to stochastic channel variations.

Our proofs for stability and performance bounds, shown in the Appendix, are rather general. They only use the general properties of convexity and Markovity and the definition of subgradients. As we will see in Subsection VI-C, the above results can readily be extended to other network optimization problems.

C. Implementation Issues

Channel Probing: Each node needs to know the channel states over the links to its neighbors. This can be achieved by each node broadcasting a pre-specified pilot signal to its neighbors, which calculate their SNR upon receiving the pilot signal and send back SNR values to the node. Each node can estimate the current channel state by the SNR values.

Global Parameter: The unit of time by which Algorithm 3 updates is decided by the nature of the wireless channel. It should not be too large, since the channel state is assumed to be fixed within a time slot. Our model is suitable for the wireless channel with long enough coherence time.

VI. Extensions and Variations

A. Ad Hoc Network with Secondary Interference

We have considered the network with primary interference. Conflict graph is a rather general construction and can accommodate other types of interference models. For example, we may consider the network with secondary interference: 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. This roughly corresponds to the virtual carrier sensing using RTS-CTS exchange as in IEEE 802.11 standard [10]. The conflict graph for the network with secondary interference is more complicated. We can follow Section III and IV, and formulate a utility optimization problem for the system and carry out cross-layer design in the same way. However, the scheduling problem (16) will be much more difficult, and is actually NP-hard. It is easy to design some heuristic algorithm but is hard to bound its performance. However, due to the broadcast nature of wireless channel, it may be possible to develop a good distributed algorithm for maximum weight independent set problem derived from a wireless network.

B. Network Cost

In our system model, we have only considered the user utility. We can introduce a variable $\lambda_{i,j}^k$ for each link (i,j) to represent the cost incurred by using the link to transmit flow to destination k. Our objective will be to maximize net-gain $\sum_s U_s(x_s) - \sum_{(i,j),k} \lambda_{i,j}^k f_{i,j}^k$ to strike a balance between user utilities and network cost. Link cost $\lambda_{i,j}^k$ can be a function of power, link state such as loss rate, or any other link metric. Following dual decomposition, we can obtain similar cross-layer congestion control, routing and scheduling algorithms as follows. In step 3) of Algorithms 1 and 3, find the destination $k(t)$ such that $k(t) \in \arg\max_k (p_i^k - p_j^k - \lambda_{i,j}^k)$ and define $w_{i,j} = p_i^{k(t)} - p_j^{k(t)} - \lambda_{i,j}^{k(t)}$. All other steps in Algorithms 1 and 3 remain the same.

The introduction of $\lambda_{i,j}^k$ facilitates the implementation of many functionalities. For example, if λ is an increasing function of transmitting power, we can do energy-aware scheduling and avoid those links with high power. If it is an increasing function of link loss rate, we can do link-state-aware scheduling and avoid less reliable links. It can also help to improve performance in delay. In our original design, the flows find their way to destinations by moving in directions of decreasing congestion price. Thus, some data may take a long path to its destination, which could lead to significant delay for large network. By taking λ proportional to the link length, we can align the nodes to route data in the direction of their destinations, and thus improve the performance in delay.

C. Stability and Optimality of A Generalized Time-Varying Queueing Network

The stability and performance bounds obtained in Section V are rather general. Here we further elaborate this point in the context of a generalized model of queueing network and general convex optimization. Consider a model of queueing network that is served by a generalized switch [27]. The generalized switch consists of a set L of interdependent parallel servers with time-varying service capabilities. The servers are interdependent in that they may not provide service simultaneously. Switch state h follows a discrete-time, irreducible finite-state Markov chain. At each time slot t, the switch can choose a scheduling decision e from a finite set E, which captures the interdependency among the servers specifying which subsets of servers can be active simultaneously. Each scheduling decision has the associated vector of service rates $r^e(h(t))$ at which queues are served, where $h(t)$ denotes the switch state at time t. As in Section III and V, for each switch state h the feasible service rate region is defined as $\Pi(h) := \{r : r = \sum_e a_e r^e(h),\ a_e \geq 0,\ \sum_e a_e = 1\}$, and let the switch state distribution be $q(h)$, the mean feasible rate region is then defined as $\overline{\Pi} := \{\overline{r} : \overline{r} = \sum_h q(h) r(h), r(h) \in \Pi(h)\}$.

Assume that the network is shared by a set S of users, which will attain a strictly concave utility $U(x)$ when the arrival rate for each user s is x_s. Suppose that we can represent the "routing" of the user service requirement by a linear function $H(x)$ of the arrival rates $\{x_s\}$. Let the achieved service rate of each server l be denoted by f_l, and we represent the "allocation" of the server capacities by a linear function $A(f)$ of service rates $\{f_l\}$. Since the service requirement should not exceed the allocated service capacity, we have the following inequality constraint $H(x) \leq A(f)$. The following optimization problem

$$\max_{x,f} \quad U(x) \tag{29}$$

$$\text{subject to} \quad H(x) \leq A(f) \quad \& \quad f \in \overline{\Pi} \tag{30}$$

can be solved by the following dual algorithm

$$x(t) = x(p(t)) = \arg\max_x U(x) - p^T(t)H(x) \quad (31)$$

$$f(t) = f(p(t)) \in \arg\max_f p^T(t)A(f) \ \ s.t. \ \ f \in \Pi(h(t)) \quad (32)$$

$$p(t+1) = [p(t) + \gamma(H(x(p(t))) - A(f(p(t))))]^+. \quad (33)$$

Using the same notation as in Section V, we can readily show the following general results:

Theorem 7: The Markov chain described by equation (33) is stable.

Theorem 8: The algorithm (31)-(33) converges statistically to within $\gamma G^2/2$ of the optimal of the system problem $(29) - (30)$, i.e.,

$$\overline{D}(E[p(\infty)]) \leq \overline{D}(p^*) + \frac{\gamma G^2}{2} \quad (34)$$

$$\overline{P}(E[x(\infty)]) \geq \overline{P}(x^*) - \frac{\gamma G^2}{2}. \quad (35)$$

The above model of queueing network is very general and has many applications in communication networks, including the model studied in last section. Other examples include joint congestion control and MAC [3] with time-varying channel, where each wireless link can be viewed as a server and the routing is specified by a routing matrix R (i.e., $H(x) = Rx$); fair scheduling in cellular network in the downlink [6] where the servers correspond to the wireless links from the base station to the users and the routing corresponds to an identity function; and TCP [20] with time-varying capacity as in last-hop wireless networks where each (wired or wireless) link can be seen as a server and the routing is again specified by a routing matrix. It can include power control as well [4] as power does not change convexity of the feasible rate region.

Convex optimization has provided a powerful tool in recent years to formulate and solve network resource allocation problems with deterministic models. Here we have established the *stability* and *optimality* of dual algorithms under channel-level stochastic for convex optimization where the constraint set has the following structure: a subset of the variables lie in a polytope and other variables lie in a convex set that vary according to an irreducible, finite-state Markov chain. Our algorithms only require the knowledge of current network state such as channel state and queue-lengths, while most other solutions require the knowledge of the statistics of channel state or keep a running average of network variables such as mean source rates. Furthermore, numerical examples in the next section also highlight *robustness* under channel-level stochastic: degradation of objective value due to suboptimal control over a subset of the variables can be mitigated by channel variations.

VII. Numerical Examples

In this section, we provide numerical examples to complement the analysis in the previous sections. We consider a simple ad hoc network shown in Fig.2, and assume that there are two network layer flows $A \rightarrow F$ and $B \rightarrow E$ with the same utility $U_s(x_s) = \log(x_s)$. We have chosen such a small, simple topology to facilitate detailed discussion of the results.

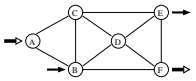

Fig. 2. A simple network with two network layer flows. All links are bidirectional.

A. Fixed Channel and Single-rate Devices

In this subsection, we consider the network with fixed link capacity. For simplicity, we assume that links (C, E), (E, C), (B, F) and (F, B) have one unit of capacity and all other links have 2 units of capacity when active. We first simulate Algorithm 1 with perfect scheduling. Fig.3 shows the evolution of source rate and congestion price of each flow with stepsize $\gamma = 0.1$. We see that they converge quickly to a neighborhood of the optimal and oscillate around the optimal. This oscillating behavior mathematically results from the non-differentiability of the dual function and physically can be interpreted as due to the scheduling process. However, Fig.4 shows that the average source rates and congestion prices are smooth and approach the optimum monotonically. We also note that the performance of the algorithm is much better than the bound of $\gamma G^2/2$ specified in Theorem 2 and 3.

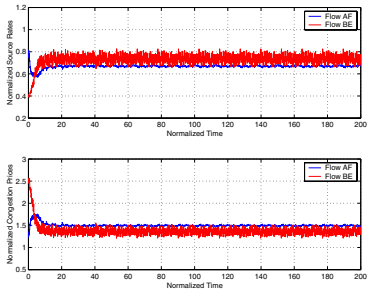

Fig. 3. Source rates and congestion prices with Algorithm 1 (perfect scheduling)

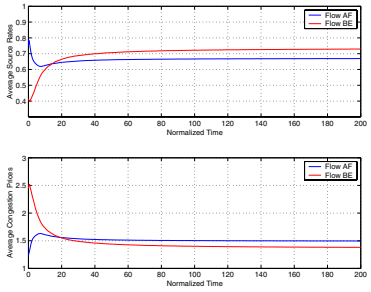

Fig. 4. The average source rates and congestion prices with algorithm 1 (perfect scheduling)

Table I shows the average link rates allocated to each flow[9]. In this table (and similar tables in this section), the first column is the sending nodes and the first row is the receiving nodes of each directed link. From this table, we can tell which paths each flow has used. Note that link $\langle B, C\rangle$ is not used. This is due to the fact that $\langle B, C\rangle$ is near the sources and is the link

[9]In this and other three tables, flows are slightly not conserved at some nodes. This is because we run numerical simulations for finite time and some residual effect of the initial condition remains.

with most contention. So, an optimal routing and scheduling will not activate it.

TABLE I

AVERAGE RATES OF FLOWS AF (UPPER TABLE) AND BE (LOWER TABLE) THROUGH DIFFERENT LINKS WITH ALGORITHM 1 (PERFECT SCHEDULING)

Rates	A	B	C	D	E	F
A	0	0.265	0.404	0	0	0
B	0	0	0	0	0	0.262
C	0	0	0	0.222	0.182	0
D	0	0	0	0	0	0.222
E	0	0	0	0	0	0.182
F	0	0	0	0	0	0

Rates	A	B	C	D	E	F
A	0	0.000	0.000	0	0	0
B	0	0	0	0.510	0	0.225
C	0	0	0	0	0	0
D	0	0	0	0	0.510	0
E	0	0	0	0	0	0
F	0	0	0	0	0.225	0

We next simulate Algorithm 1 with the distributed, approximate scheduling (Algorithm 2). The results are shown in Fig.5 and Fig.6. The evolutions of source rates, congestion prices and their averages are similar to those with perfect scheduling: they converge quickly to a neighborhood of stable values. As expected, the source rates are less than those achieved with perfect scheduling, since the feasible rate region is smaller under the approximate scheduling. Table II summarizes the

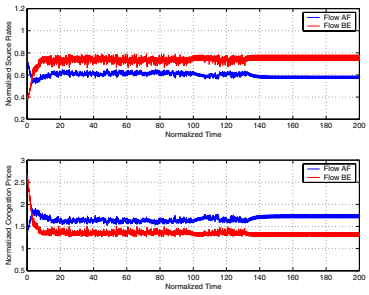

Fig. 5. Source rates and congestion prices with Algorithm 1 (distributed scheduling)

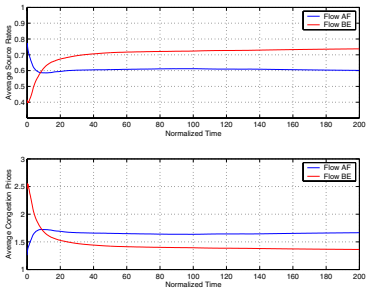

Fig. 6. The average source rates and congestion prices with Algorithm 1 (distributed scheduling)

average link rates allocated to each flow. We see that the routing pattern has been changed, due to the distributed scheduling. Also note that every link is used in routing, since each link has a chance to be a locally heaviest link.

Though its worst case performance bound is $1/2$, our simulation results show that the degradation of the performance of

TABLE II

AVERAGE RATES OF FLOWS AF (UPPER) AND BE (LOWER) THROUGH DIFFERENT LINKS WITH ALGORITHM 1 (DISTRIBUTED SCHEDULING)

Rates	A	B	C	D	E	F
A	0	0.310	0.290	0	0	0
B	0	0	0	0	0	0.307
C	0	0	0	0.070	0.219	0
D	0	0	0	0	0	0.070
E	0	0	0	0	0	0.219
F	0	0	0	0	0	0

Rates	A	B	C	D	E	F
A	0	0.000	0	0	0	0
B	0	0	0.045	0.697	0	0.008
C	0	0	0	0	0.045	0
D	0	0	0	0	0.697	0
E	0	0	0	0	0	0
F	0	0	0	0	0.008	0

Algorithm 1 with distributed scheduling is small. Combined with its low communication overhead, fast convergence, and good performance with distributed scheduling, our cross-layer design scheme is promising for practical implementation.

B. Time-varying Channel and Multi-rate Devices

We now consider the network with time-varying link capacity. For simplicity, we assume that links (C, E), (E, C), (B, F) and (F, B)'s capacities are identically, uniformly distributed over 0.5, 1 and 1.5 units, while other links' capacities are identically, uniformly distributed over 1, 2 and 3 units. Thus, the average capacity for each link when active is the same as that in the examples of last subsection.

We first simulate Algorithm 3 with perfect scheduling. Fig.7 and Fig.8 show the evolution of source rates, congestion prices and their averages with the same step size $\gamma = 0.1$. The source rates and congestion prices have much larger variations than those with fixed channel, due to the channel variations. But the average source rates and congestion prices are still smooth, and converge quickly and monotonically to optimal values. Note that, although the average link capacity when active is the same as that in fixed channel, each flow achieves larger sending rates. This is due to the multi-user diversity that we exploit when doing scheduling. Also note that the increase in sending rate of flow BE is much more notable. This is because node B has four neighbors and thus a much larger multi-user diversity.

Table III summarizes the average link rates allocated to each flow. We see that the routing pattern has changed for flow BE, while almost all the data for flow AF are routed along the same pathes as those for the network with fixed link capacities and perfect scheduling. This change is due to the time-varying capacities, which makes every link have a chance to be a globally heavy link for some channel state and thus affects the paths each flow takes.

We next simulate Algorithm 3 with distributed, approximate scheduling Algorithm 2. The results are shown in Fig.9 and Fig.10. Compared to the results in the last subsection with fixed capacities, the performance degradation of Algorithm 3 with distributed scheduling is very small, which means that

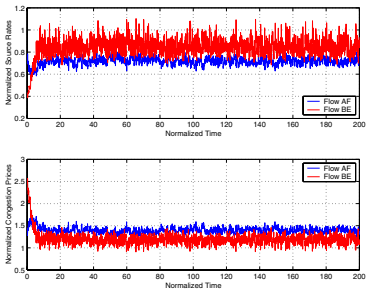

Fig. 7. Source rates and congestion price with Algorithm 3 (perfect scheduling)

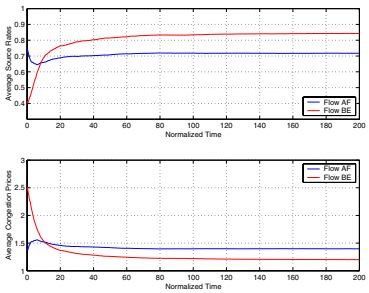

Fig. 8. The average source rates and congestion price with Algorithm 3 (perfect scheduling)

TABLE III

AVERAGE RATES OF FLOWS AF (UPPER TABLE) AND BE (LOWER TABLE) THROUGH DIFFERENT LINKS WITH ALGORITHM 3 (PERFECT SCHEDULING)

Rates	A	B	C	D	E	F
A	0	0.328	0.390	0	0	0
B	0	0	0	0.074	0	0.253
C	0	0.001	0	0.243	0.147	0
D	0	0	0	0	0.022	0.295
E	0	0	0	0	0	0.169
F	0	0	0	0	0	0

Rates	A	B	C	D	E	F
A	0	0	0.104	0	0	0
B	0.104	0	0.032	0.504	0	0.211
C	0	0	0	0.012	0.124	0
D	0	0	0	0	0.443	0.072
E	0	0	0	0	0	0
F	0	0	0	0	0.283	0

channel variation improves the performance of the distributed scheduling algorithm. To see how this happens, note that the scheduling defined by (25) is optimal only at time t, but not at other times due to channel's time-variation. For example, assume at time t link $\langle B, A\rangle$ is a globally heavy link but not a locally heaviest link, and link $\langle B, C\rangle$ is a locally heaviest link but not a globally heavy link. With perfect scheduling, link $\langle B, A\rangle$ will be scheduled to transmit and $\langle B, C\rangle$ will not. With distributed scheduling, link $\langle B, C\rangle$ will be scheduled to transmit and $\langle A, C\rangle$ will not. Now, suppose at later time slots link $\langle A, C\rangle$ has very low capacity while link $\langle C, E\rangle$ has a high capacity such that link $\langle C, E\rangle$ is scheduled to transmit. In this situation, with perfect scheduling at time t data will be stuck at A and thus the sending rate of flow BE will decrease, but distributed, approximate scheduling at time t will get more data to the destination. So, in a time-varying environment, at any time t we cannot say that a perfect scheduling is necessarily better than an approximate one. Thus, the optimality of scheduling (25) is relatively not that important, and a reasonable approximation works well. Also, Table IV summarizes the link rates allocated to each flow. As expected, the routing is more complicated in this situation, since local optimal scheduling combined with time-varying capacities makes every link has a good chance to be scheduled for transmission.

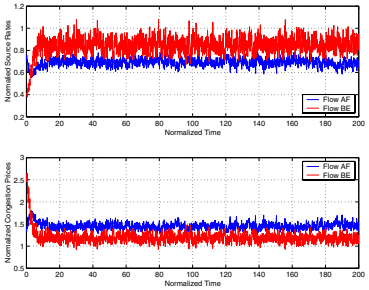

Fig. 9. Source rates and congestion price with Algorithm 3 (distributed scheduling)

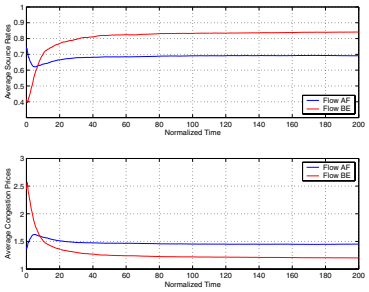

Fig. 10. The average source rates and congestion price with Algorithm 3 (distributed scheduling)

TABLE IV

AVERAGE RATES OF FLOWS AF (UPPER) AND BE (LOWER) THROUGH DIFFERENT LINKS WITH ALGORITHM 3 (DISTRIBUTED SCHEDULING)

Rates	A	B	C	D	E	F
A	0	0.330	0.361	0	0	0
B	0	0	0	0.095	0	0.239
C	0	0.006	0	0.232	0.123	0
D	0	0	0	0	0.004	0.323
E	0	0	0	0	0	0.127
F	0	0	0	0	0	0

Rates	A	B	C	D	E	F
A	0	0	0.053	0	0	0
B	0.053	0	0.128	0.523	0	0.144
C	0	0	0	0.010	0.169	0
D	0	0	0	0	0.498	0.035
E	0	0	0	0	0	0
F	0	0	0	0	0.179	0

Our simulation results have confirmed the conclusions from Theorem 5 and 6, which say that the average source rates and congestion prices approach the optimum of an ideal system with the best feasible rate region at link layer, and that Algorithm 3 can been seen as a distributed algorithm to solve this ideal system problem. We also have seen that the distributed scheduling algorithm works well with time-varying

channel, and the channel variations in fact help mitigate the overall system's degradation due to suboptimal design in one layer.

VIII. Conclusions

We have presented a model for the joint design of congestion control, routing and scheduling for ad hoc wireless networks by extending the framework of network utility maximization and applying dual-based decompositions. We formulate resource allocation in the network with fixed wireless channels or single-rate wireless devices as a utility maximization problem with schedulability and rate constraints arising from contention for the wireless channel. By dual decomposition, we derive a subgradient algorithm that is not only distributed spatially, but more interestingly, decomposes the system problem vertically into three protocol layers where congestion control, routing and scheduling jointly solve the network utility maximization problem. We also extend the dual algorithm to handle the network with time-varying channel and adaptive multi-rate devices, and surprisingly show that, despite stochastic channel variation, it solves an ideal reference system problem which has the best feasible rate region at link layer.

Dual algorithms for convex optimization formulations of generalized network utility maximization have found many applications recently for both deterministic and connection-level stochastic models. We show that, for a large class of such convex optimization problems, stability and average performance are not affected by channel-level stochastic models. This provides a general technique to carry out optimization-based network designs in a time-varying environment.

Further research steps stemming out of this paper include the following. First, unique features in our algorithm for practical implementations need to be further leveraged. Second, we will extend the results to networks with more general interference models and/or node mobility. Third, scheduling problem is always a challenging problem for ad hoc network, and continued exploration of distributed scheduling protocols will further enhance the performance gain from cross-layer design involving link layer. Fourth, we will formally quantify the interesting observation that channel variations in fact help mitigate the overall system's degradation due to suboptimal design in one layer.

Acknowledgments

This work is partially supported by Boeing, Army Institute for Collaborative Biotechnologies, Air Force Office of Scientific Research Award FA9550-05-1-0032 "Bio-Inspired Networks", ARO through grant DAAD19-02-1-0283, NSF through grants CNS-0435520, CCF-0448012, CNS-0417607, CNS-0427677, and the Caltech Lee Center for Advanced Networking.

References

[1] S. Asmussen, *Applied Probability and Queues*, 2nd ed., Springer, 2003.
[2] D. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena scientific, 1999.
[3] L. Chen, S. Low and J. Doyle, Joint congestion control and media access control design for ad hoc wireless networks, *Proc. IEEE Infocom*, 2005.
[4] M. Chiang, Balancing transport and physical layers in wirless multihop networks: Jointly optimal congestion control and power control, *IEEE J. Sel. Area Comm.*, vol. 23, no. 1, pp. 104-116, Jan. 2005.
[5] M. Chiang, S. H. Low, R. A. Calderbank, and J. C. Doyle, Layering as optimization decomposition, *Proceedings of IEEE*, 2006.
[6] A. Eryilmaz and R. Srikant, Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control, *Proc. IEEE INFOCOM*, 2005.
[7] B. Hajek and G. Sasaki, Link scheduling in polynomial time, *IEEE Trans. Information Theory*, 34:910-917, Sept. 1988.
[8] J. He, M. Chiang, and J. Rexford, TCP/IP interactions based on congestion price: Stability and optimality, *Proc. IEEE ICC*, June 2006.
[9] J. Hoepman, Simple distribute weighted matchings, preprint, October 2004. Available at http://arxiv.org /abs/cs/0410047.
[10] IEEE, Wirless LAN Media Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Standard 802.11, June 1999.
[11] K. Jain, J. Padhye, V. N. Padmanabhan and L. Qiu, Impact of interference on multi-hop wireless network performance, *Proc. ACM Mobicom*, 2003.
[12] D. B. Johnson and D. A. Maltz, Dynamic source routing in ad-hoc wireless networks, *Mobile Computing*, (Eds. T. Imielinski and H. Korth), Kluwer Academic Publishers, 1996.
[13] 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.
[14] M. Kodialam and T. Nandagopal, Charaterizing achievable rates in multi-hop wireless networks: The joint routing and scheduling problem, *Proc. ACM Mobicom*, September 2003.
[15] 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.
[16] J.-W. Lee, M. Chiang, and R. A. Calderbank, Price-based distributed algorithm for optimal rate-reliability tradeoff in network utility maximization, *IEEE Journal of Selected Areas in Communications*, 2006.
[17] J.-W. Lee, M. Chiang, and R. A. Calderbank, Jointly optimal congestion and contention control in wireless ad hoc networks, *IEEE Communication Letters*, 2006
[18] X. Lin and N. Shroff, Joint rate control and scheduling in multihop wireless networks, *43th IEEE CDC*, 2004.
[19] X. Lin and N. Shroff, The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks, *Proc. IEEE Infocom*, 2005.
[20] S. H. Low and D. E. Lapsley, Optimal flow control, I: Basic algorithm and convergence, *IEEE/ACM Trans. on networking*, 7(6):861-874, 1999.
[21] S. H. Low, A duality model of TCP and active queue management algorithms, *IEEE/ACM Transactions on Networking*, October 2003.
[22] M. Neely, E. Modiano and C. Rohrs, Dynamic power allocation and routing for time varying wireless networks *Proc. IEEE Infocom*, 2003. Journal version, *IEEE J. Sel. Area Comm.*, 23(1):89-103, 2005.
[23] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithm and Complexity*, Dover Publications, 1998.
[24] R. Preis, Linear time 1/2-approximation algorithm for maximum weighted matching in general graphs, *16th STACS*, (Eds., C, Meinel and S. Tison), LNCS 1563, Springer, 1999.
[25] C. E. Perkins and E. M. Royer, Ad-hoc on-demand distance vector routing, *WMCSA*, Febuary 1999.
[26] N. Z. Shor, *Monimization Methods for Non-Differentiable Functions*, Springer-Verlag, 1985.
[27] A. L. Stolyar, MaxWeight scheduling in a generalized switch: state space collapse and workload minimization in heavy traffic, *Ann. Appl. probab.*, **14**(1):1-53, 2004.
[28] A. L. Stolyar, Maximizing queueing network utility subject to statbility: greedy primal-dual algorithm, *Queueing Systems*, 50(4):401-457, 2005.
[29] L. Tassiulas and A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, *IEEE Trans. on Automatic Control*, **37**(12):1936-1948, December 1992.
[30] R. Uehara and Z. Chen, parallel approximation algorithms for maximum weighted matching in general graphs *Inf. Proc. Letter*, 76:13-17, 2000.
[31] J. Wang, L. Li, S. H. Low and J. C. Doyle, Cross-layer optimization in TCP/IP networks, *IEEE/ACM Trans. on Networking*, June 2005.
[32] X. Wang and K. Kar, Cross-Layer Rate Control for End-to-End Proportional Fairness in Wireless Networks with Random Access, *Proc. ACM MobiHoc*, 2005.
[33] M. Wattenhofer and R. Wattenhofer, Distributed weighted matching, *18th DISC*, (Ed., R. Guerraoui) LNCS 3274, Springer, 2004.
[34] L. Xiao, M. Johnasson and S. Boyd, Simultaneous routing and resource allocation via dual decomposition, *IEEE Trans. Comm.*, July 2004.

[35] Y. Xue, B. Li and K. Nahrstedt, Price-based resource allocation in wireless ad hoc networks, *Proc. ACM IWQoS*, 2003.
[36] Y. Yi and S. Shakkottai, Hop-by-hop congestion control over a wireless multi-hop network, *Proc. IEEE Infocom*, March 2004.

APPENDIX: PROOFS FOR THEOREMS 2, 3, 5 AND 6

(Theorem 2) *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 sub-gradient. Applying the inequalities recursively, we obtain

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

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

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

From this inequality we obtain

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

Since D is a convex function, by Jensen's inequality,

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

Thus, $\limsup_{t\to\infty} D(\overline{p}(t)) - D(p^*) \leq \frac{\gamma G^2}{2}$, i.e., the algorithm converges statistically to within $\gamma G^2/2$ of the optimal value. ■

(Theorem 3) *Proof:* By equation (18), we have

$$\begin{aligned} & ||p(t+1)||_2^2 \leq ||p(t)-\gamma g(p(t))||_2^2 \\ =\ & ||p(t)||_2^2 - 2\gamma g(p(t))^T p(t) + \gamma^2||g(p(t))||_2^2 \\ =\ & ||p(t)||_2^2 + 2\gamma\sum_s U_s(x_s(t)) - 2\gamma(\sum_s U_s(x_s(t)) - p_s(t)x_s(t)) \\ & -2\gamma\sum_{i,j,k} p_i^k(f_{i,j}^k(t) - f_{j,i}^k(t)) + \gamma^2||g(p(t))||_2^2 \\ \leq\ & ||p(t)||_2^2 + 2\gamma\sum_s U_s(x_s(t)) - 2\gamma(\sum_s U_s(x_s^*) - p_s(t)x_s^*) \\ & -2\gamma\sum_{i,j,k} p_i^k(t)(f_{i,j}^k(t) - f_{j,i}^k(t)) + \gamma^2||g(p(t))||_2^2 \\ =\ & ||p(t)||_2^2 + 2\gamma P(x(t)) - 2\gamma P(x^*) + \gamma^2||g(p(t))||_2^2 \\ & -2\gamma\sum_{i,j,k} p_i^k(t)(f_{i,j}^k(t) - f_{j,i}^k(t) - (x^*)_i^k) \\ \leq\ & ||p(t)||_2^2 + 2\gamma P(x(t)) - 2\gamma P(x^*) + \gamma^2||g(p(t))||_2^2 \end{aligned}$$

where the second inequality follows from the fact that $x(t)$ is the maximizer in the problem (10), and the third inequality follows from the fact that $f(t)$ is the maximizer in problem (11)-(12). Applying the inequalities recursively, we obtain

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

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

$$\begin{aligned} 2\gamma\sum_{\tau=1}^{t}(P(x(\tau))-P(x^*)) \quad &\geq \quad -||p(1)||_2^2 - \gamma^2\sum_{\tau=1}^{t}||g(p(\tau))||_2^2 \\ &\geq \quad -||p(1)||_2^2 - t\gamma^2G^2 \end{aligned}$$

From this inequality we obtain

$$\frac{1}{t}\sum_{\tau=1}^{t}P(x(\tau)) - P(x^*) \geq \frac{-||p(1)||_2^2 - t\gamma^2G^2}{2t\gamma}$$

Since P is a concave function, by Jensen's inequality,

$$P(\overline{x}(t)) - P(x^*) \geq \frac{-||p(1)||_2^2 - t\gamma^2G^2}{2t\gamma}$$

Thus, $\liminf_{t\to\infty} P(\overline{x}(t)) \geq P(x^*) - \frac{\gamma G^2}{2}$. ■

(Theorem 5) *Proof:* From the proof of Theorem 4, we have

$$\begin{aligned} E[\Delta V_t(p)|p] \quad &= \quad E[V(p(t+1)) - V(p(t)) \mid p(t)=p] \\ &\leq \quad 2\gamma(\overline{D}(p^*) - \overline{D}(p)) + \gamma^2G^2 \end{aligned}$$

Taking expectation over p, we get

$$E[\Delta V_t(p)] = E[V(p(t+1)) - V(p(t))] \leq 2\gamma(\overline{D}(p^*) - E[\overline{D}(p)]) + \gamma^2G^2$$

Taking summation from $\tau=0$ to $\tau=t$, we obtain

$$E[V(p(t+1))] \leq E[V(p(1))] - 2\gamma\sum_{\tau=1}^{t}(E[\overline{D}(p(\tau))] - \overline{D}(p^*)) + t\gamma^2G^2$$

Since $E[V(p(t+1))] \geq 0$, we have

$$\frac{1}{t}\sum_{\tau=1}^{t}(E[\overline{D}(p(\tau))] - \overline{D}(p^*)) \leq \frac{E[V(p1)] + t\gamma^2G^2}{2t\gamma}$$

Note that $p(t)$ is ergodic in some steady state by Theorem 4, and so is $\overline{D}(p(t))$. Thus,

$$\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t}(E[\overline{D}(p(\tau))] - \overline{D}(p^*)) = E[\overline{D}(p(\infty))] - \overline{D}(p^*)$$

So,

$$E[\overline{D}(p(\infty))] - \overline{D}(p^*) \leq \gamma G^2/2.$$

Since $\overline{D}(p)$ is a convex function, by Jensen's inequality, $\overline{D}(E[p(\infty)]) - \overline{D}(p^*) \leq \gamma G^2/2$, i.e., the algorithm converges statistically to within $\gamma G^2/2$ of the optimal value $\overline{D}(p^*)$. ■

(Theorem 6) *Proof:* $x(t)$ is a stable Markov chain, since it is a deterministic function of congestion price $p(t)$ and $p(t)$ is stable. The proof for the second part of the theorem is a straightforward extension of the proof of Theorem 3, following similar procedure as in the proof of Theorem 5. We skip the details here. ■