# JOINT STOCHASTIC ROUTING AND SCHEDULING FOR MULTIHOP WIRELESS AD-HOC NETWORKS

*Alejandro Ribeiro and Georgios B. Giannakis*

Dept. of ECE, University of Minnesota
200 Union Street S.E., Minneapolis, MN 55455

## ABSTRACT

We consider the problem of finding multihop routes in a wireless ad-hoc network jointly with scheduling transmission times of different information flows. Taking into account the unreliable nature of wireless channels, we derive a joint stochastic routing-scheduling algorithm whereby schedules and routes are selected at random with certain probabilities that we optimize. We prove that if there exists a set of (random) schedules and routes ensuring that all queues in the network are stable, our protocol converges to one such set. Our approach to the problem is to: i) characterize the set of scheduling-routing policies guaranteeing that all queues in the network are stable; ii) show that this can be reduced to finding a set of auxiliary variables in a convex polyhedron; and iii) use dual decomposition techniques to develop an algorithm converging to a point inside this convex polyhedron.

## 1. INTRODUCTION

In lieu of a fixed infrastructure, wireless ad-hoc networks count on peers relaying packets for each other in order to establish and carry on communications. Among the challenges nodes face in such networks, we find the problems of establishing routes to intended destinations, adjusting transmission rates to avoid network congestion and scheduling transmission times among different flows. For a first approach to defining our problem, consider an ad-hoc network with $J$ terminals $\{U_j\}_{j=1}^J$. Terminal $U_j$ intends to send packets to $U_k$ at a rate of $\rho_{kj}$ packets per unit time while at the same time collaborating in forwarding packets from nearby nodes. In principle, there are three questions that $U_j$ wants to answer:

(i) Can the network deliver my packets to $U_k$ at a rate $\rho_{kj}$?
(ii) If so, which among my neighboring nodes is a convenient next hop? Besides, since I am also collaborating in routing packets from nearby nodes what are convenient next hops for the packets I am receiving from my neighbors?
(iii) How should I divide my transmission time among the information flows I am serving?

To answer these questions we respectively need to solve the so called flow control, routing and scheduling problems. In the context of wired networks, the landmark work in [10, 11] offers a joint solution to these three problems through the "back-pressure" algorithm whereby routing-scheduling decisions are based on the difference between queue lengths of adjacent terminals. It is shown that if there exists a routing-scheduling policy affording delivery rates $\rho_{kj}$ from $U_j$ to $U_k$, then the back-pressure algorithm is one such policy.

Wireless networks are further complicated by the unreliable nature of wireless connections. To be specific, let $R_{ij}$ denote the probability that a packet transmitted by user $U_j$ is correctly decoded by user $U_i$. In a wired network the reliability $R_{ij}$ is either very close to 1, if there is a (physical) link between $U_j$ and $U_i$, or 0, if there is not. In a wireless network however, the whole range of $R_{ij}$ values may and indeed happen in practice, as testified by experimental measurements [1]. In the context of routing, a number of works deal with links of intermediate reliability by finding shortest path routes in graphs with link metrics that depend on $R_{ij}$; see e.g., [6]. In particular, the link cost $1/R_{ij}$ that penalizes but does not preclude the use of unreliable links has found widespread acceptance [4]. An alternative approach to shortest path routing is to formulate routing algorithms as network utility optimization problems based on the matrix $\mathbf{R}$ with entries $R_{ij}$. It has been recently shown that many routing schemes can be formulated based on $\mathbf{R}$ as convex optimization problems thus ensuring algorithmic tractability [8]. Furthermore, dual decomposition techniques can be used to solve these optimization problems in a distributed manner [9].

Our goal in this paper is to develop a joint routing-scheduling protocol that given a set of rates $\rho_{kj}$ finds a set of routes and schedules so that all queues in the network are stable. If such set of routes does not exist, the protocol will tell so. The random nature of wireless channels lends itself naturally to stochastic routing-scheduling policies. At a given slot, user node $U_j$ schedules destination $U_k$ by transmitting the packet to (i.e., routing through) $U_i$ with a certain probability $T_{kij}$. Instead of a routing table, a stochastic routing-scheduling policy finds a set of probability matrices $\{\mathbf{T}_k\}_{k=1}^K$ with entries $T_{kij}$; where $K$ denotes the number of destinations, see Section 2. Our approach to finding a policy that stabilizes all queues in the network, when this is possible, is as follows:

(i) Given rates $\rho_{kj}$ we characterize the stability region $\mathcal{S}$ of joint scheduling-routing policies $\{\mathbf{T}_k\}_{k=1}^K$ guaranteeing that all the queues in the network are stable.
(ii) We show that the problem of finding a routing policy in $\mathcal{S}$ can be reduced to the problem of finding a set of auxiliary variables $\{\mathbf{U}_k\}_{k=1}^K$ in a convex polyhedron $\mathcal{S}'$.
(iii) Finding variables $\{\mathbf{U}_k\}_{k=1}^K$ in $\mathcal{S}'$ is posed as a convex optimization problem, which allows dual decomposition techniques to find iterates $\{\mathbf{U}_k(n)\}_{k=1}^K$ converging to a point $\{\mathbf{U}_{k\infty}\}_{k=1}^K$ in $\mathcal{S}'$.
(iv) The routing probabilities $\{\mathbf{T}_k\}_{k=1}^K$ are obtained from the auxiliary variables $\{\mathbf{U}_{k\infty}\}_{k=1}^K$ via a closed-form transformation.

Work in this paper was prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.
Email at (aribeiro, georgios)@ece.umn.edu

## 2. PROBLEM FORMULATION

Consider a wireless ad-hoc network with $J$ terminals $\{U_j\}_{j=1}^J$ collaborating to support a set of ongoing communications. Without loss of generality, suppose that the first $K$ terminals $\{U_k\}_{k=1}^K$ are destinations of packets randomly generated at other terminals, with $\rho_{kj}$ denoting the rate at which $U_j$ generates packets whose intended destination is $U_k$. To exemplify notation consider a network with $J/2$ bidirectional communications between pairs of nodes $U_j, U_{k(j)}$. In this case we would have that: i) every node is a destination, i.e., $K = J$; ii) the arrival rate is null except for communicating pairs, i.e., $\rho_{kj} = 0$ when $k \neq k(j)$; and iii) arrival rates $\rho_{kj} \neq 0$ if and only if $\rho_{jk} \neq 0$. In general, some nodes may not be receiving packets in which case $K < J$; some $U_k$ node may receive packets from more than one source implying that $\rho_{kj} \neq 0$ for more than one $j$; and some $U_j$ node may not be sending packets resulting in $\rho_{kj} = 0$, $\forall\ k$. We assume that the random processes generating packets are stationary and define the vectors $\boldsymbol{\rho}_k := [\rho_{k1}, \ldots, \rho_{kJ}]^T$ of rates with destination $U_k$. We further convene $\rho_{kk} = 0$.

Terminals transmit at random with probabilities $\boldsymbol{\mu} := [\mu_1, \ldots, \mu_J]^T$. We describe the network topology by the pairwise correct detection probabilities $R_{ij}$ that we define as the probability of $U_i$ correctly decoding a packet transmitted by $U_j$; i.e.,

$$R_{ij} = \Pr\{U_i \text{ decodes } U_j\}. \tag{1}$$

We arrange these probabilities in the reliability matrix $\mathbf{R}$ with $(i,j)$-th entry $R_{ij}$. The matrix $\mathbf{R}$ is a function of the transmission probabilities $\boldsymbol{\mu}$, transmitted power and other parameters pertaining to the physical and medium access layers but is assumed given for our purposes. Counting acknowledgements from $U_i$, node $U_j$ can estimate $R_{ij}$ as the ratio between the number of acknowledgements received over the number of packets sent to $U_i$.

At each time slot, a terminal $U_j$ that decides to transmit a packet is faced with a scheduling and a routing decision. Of the intended destinations $\{U_k\}_{k=1,k\neq j}^K$, node $U_j$ has to decide which one it is going to serve, i.e., schedule, in the current slot. Given that it chooses to send a packet whose final destination is $U_k$, node $U_j$ chooses a convenient next hop $\{U_i\}_{i=1,i\neq j}^J$, i.e, $U_j$ routes the packet through $U_i$.

### 2.1. Stochastic routing-scheduling

Recall that $T_{kij}$ is the probability of $U_j$ scheduling $U_k$, and routing the packet through $U_i$. Consequently, at any slot, say the $n$-th, $U_j$ decides to transmit with probability $\mu_j$, if it chooses so, it then selects a final destination $U_k$ *and* a next hop $U_i$ with the pair $(U_k, U_i)$ chosen with probability $T_{kij}$. If the transmission is successful, something that happens with probability $R_{ij}$, the packet moves to $U_i$'s queue. Otherwise, the packet stays at $U_i$ to be later retransmitted, possibly to a different node. Since at any slot in which $U_j$ transmits it has to serve some node we have that

$$\sum_{k=1}^{K}\sum_{i=1}^{J} T_{kij} = 1, \ \ \forall\ j \neq k. \tag{2}$$

We also have $T_{kik} = 0, \forall\ i$ and $T_{kjj} = 0, \forall\ j$, respectively meaning that a destination $U_k$ does not forward its own packets and that $U_j$ does not route packets through itself. For future reference define the matrix $\mathbf{T}_k \in \mathbb{R}^{J\times J}$, with $(i,j)$-th element $T_{kij}$.

To characterize the evolution of packets through the network we define a third matrix $\mathbf{K}_k$ with elements $K_{kij}$ denoting the probability that a packet with final destination $U_k$ moves from $U_j$'s to $U_i$'s queue. For $i \neq j$, this happens if and only if the pair of destination and next hop $(U_k, U_i)$ is chosen by $U_j$ and the packet is correctly decoded, that is

$$K_{kij} = T_{kij} R_{ij}, \ i \neq j. \tag{3}$$

If the packet does not move from $U_j$ to some $U_i \neq U_j$, it stays at $U_j$, which implies that $K_{kjj} = 1 - \sum_{i=1,i\neq j}^{J} K_{kij}$. The latter can be more compactly written as $\mathbf{K}_k^T\mathbf{1} = \mathbf{1}$.

The constraints in (2) and (3) define the set of all possible joint scheduling-routing strategies that can be implemented in a network with reliability matrix $\mathbf{R}$. To simplify subsequent notation let us summarize the discussion by defining the feasible set

$$\begin{aligned}\mathcal{F} := \Big\{ &\{\mathbf{T}_k\}_{k=1}^K, \{\mathbf{K}_k\}_{k=1}^K \\ &\sum_{k=1}^{K}\sum_{i=1}^{J} T_{kij} = 1,\ j\neq k;\ T_{kik} = 0, \forall i;\ T_{kjj} = 0, \forall j \\ &K_{kij} = T_{kij}R_{ij},\ i\neq j;\ \mathbf{K}_k^T\mathbf{1} = \mathbf{1} \Big\}\end{aligned} \tag{4}$$

that we interpret as the set of all possible scheduling-routing matrices $\{\mathbf{T}_k\}_{k=1}^K$ and associated packet evolution matrices $\{\mathbf{K}_k\}_{k=1}^K$. Our goal is to find a set of joint scheduling-routing matrices $\{\mathbf{T}_k\}_{k=1}^K$ such that the system is stable [5].

## 3. CENTRALIZED FEASIBILITY PROBLEM

To characterize the stability region, note that $U_j$'s $k$-th queue is loaded with packets generated at $U_j$ and packets received from other terminals. Let $\lambda_{jk}$ denote the aggregate arrival rate at $U_j$'s $k$-th queue and $\lambda_{jk}^o$ the corresponding departure (output) rate. The queue balance equations yield the relation

$$\lambda_{kj} = \rho_{kj} + \sum_{j=1}^{J} K_{kij}\lambda_{kj}^o \le \rho_{kj} + \sum_{j=1}^{J} K_{kij}\lambda_{kj} \tag{5}$$

where the inequality comes from the fact that $\lambda_{jk} = \lambda_{jk}^o$ when the queue is stable and $\lambda_{jk} < \lambda_{jk}^o$ when it is not. Upon defining the vector $\boldsymbol{\lambda}_k := [\lambda_{k1}, \ldots, \lambda_{kJ}]^T$, we can rewrite (5) as

$$(\mathbf{I} - \mathbf{K}_k)\boldsymbol{\lambda}_k \le \boldsymbol{\rho}_k \tag{6}$$

with equality achieved when all queues are stable.

From (6) we can obtain a condition ensuring that the system is stable as stated in the following proposition:

**Proposition 1** *For a given set of required rates $\{\boldsymbol{\rho}_k\}_{k=1}^K$, define the set*

$$\mathcal{S}\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right) := \left\{ \{\mathbf{T}_k\}_{k=1}^K : \boldsymbol{\rho}_k \le (\mathbf{I}-\mathbf{K}_k)\boldsymbol{\lambda}_k;\ 0 \le \sum_{\substack{k=1\\k\neq j}}^{K} \lambda_{kj} \le \mu_j \right\} \tag{7}$$

*where we implicity require $\left\{\{\mathbf{T}_k\}_{k=1}^K, \{\mathbf{K}_k\}_{k=1}^K\right\} \in \mathcal{F}$ with $\mathcal{F}$ as in (4). Then, for a given set of matrices $\{\mathbf{T}_k\}_{k=1}^K$ we have:*

*(i) if $\{\mathbf{T}_k\}_{k=1}^K$ is strictly inside $\mathcal{S}(\{\boldsymbol{\rho}_k\}_{k=1}^K)$; i.e., if $\{\mathbf{T}_k\}_{k=1}^K \in \mathcal{S}(\{\boldsymbol{\rho}_k\}_{k=1}^K) - \partial\mathcal{S}(\{\boldsymbol{\rho}_k\}_{k=1}^K)$ all queues in the network are stable;*

*(ii) if $\{\mathbf{T}_k\}_{k=1}^K \notin \mathcal{S}(\{\boldsymbol{\rho}_k\}_{k=1}^K)$ at least one queue is unstable.*

Using Proposition 1 we reformulate our goal as developing an algorithm which for a required set of rates $\{\boldsymbol{\rho}_k\}_{k=1}^K$ finds a set of matrices $\{\mathbf{T}_k\}_{k=1}^K$ so that $\{\mathbf{T}_k\}_{k=1}^K \in \mathcal{S}(\{\boldsymbol{\rho}_k\}_{k=1}^K)$. However, finding a point – i.e., set of matrices $\{\mathbf{T}_k\}_{k=1}^K$ – in $\mathcal{S}\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$ is computationally difficult. Among other conditions, a set of matrices $\{\mathbf{T}_k\}_{k=1}^K \in \mathcal{S}\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$ is such that the constraints $\boldsymbol{\rho}_k \leq (\mathbf{I}-\mathbf{K}_k)\boldsymbol{\lambda}_k$ are satisfied for some $\boldsymbol{\lambda}_k$. This is a bilinear inequality, implying that our scheduling-routing problem belongs to the class of (computationally expensive) bilinear programs.

Fortunately, we can exploit the structure of the constraints to reformulate the problem as we state in the following proposition.

**Proposition 2** *For a given set of required rates $\{\boldsymbol{\rho}_k\}_{k=1}^K$ consider matrix variables $\left\{\{\mathbf{U}_k\}_{k=1}^K, \{\mathbf{L}_k\}_{k=1}^K\right\} \in \mathcal{F}$ and define the set*

$$\mathcal{S}'\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)=\left\{\{\mathbf{U}_k\}_{k=1}^K : \boldsymbol{\rho}_k \leq (\mathbf{I}-\mathbf{L}_k)\mathbf{1};\ 0 \leq \sum_{k=1}^K\sum_{i=1}^J U_{kij} \leq \mu_j\right\}. \tag{8}$$

*Consider the transformation $T'_{kij} = U_{kij}/\sum_{k=1}^K\sum_{i=1}^J U_{kij}$ and define the set*

$$\mathcal{S}''\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)=\left\{\{\mathbf{T}'_k\}_{k=1}^K : T'_{kij} = \frac{U_{kij}}{\sum_{k=1}^K\sum_{i=1}^J U_{kij}}\right\} \tag{9}$$

*where $\{\mathbf{U}_k\}_{k=1}^K \in \mathcal{S}'\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$. Then, the sets $\mathcal{S}\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$ in (7) and $\mathcal{S}''\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$ in (9) are equal, i.e.,*

$$\mathcal{S}''\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right) = \mathcal{S}\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right). \tag{10}$$

Using Proposition 2 the routing-scheduling problem reduces to finding a set of matrices $\{\mathbf{U}_k\}_{k=1}^K$ such that

$$\begin{aligned}
&\boldsymbol{\rho}_k \leq (\mathbf{I}-\mathbf{L}_k)\mathbf{1} \qquad (11)\\
&0 \leq \sum_{k=1}^K\sum_{i=1}^J U_{kij} \leq \mu_j,\ j \neq k;\ U_{kik}=0, \forall i;\ U_{kjj}=0, \forall j\\
&L_{kij} = U_{kij}R_{ij},\ i \neq j;\ \mathbf{L}_k^T\mathbf{1} = \mathbf{1}
\end{aligned}$$

We then find the routing matrices $\{\mathbf{T}_k\}_{k=1}^K$ with the transformation $T_{kij} = U_{kij}/\sum_{k=1}^K\sum_{i=1}^J U_{kij}$.

Different from the computationally challenging problem of finding $\{\mathbf{T}_k\}_{k=1}^K \in \mathcal{S}\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$, finding $\{\mathbf{U}_k\}_{k=1}^K \in \mathcal{S}'\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$ is computationally tractable. Indeed, since $\mathcal{S}'\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$ is defined by a set of linear equalities and inequalities, finding $\{\mathbf{U}_k\}_{k=1}^K \in \mathcal{S}'\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$ is a convex feasibility problem that can be efficiently solved using e.g., convex optimization tools [3]. In particular, dual decomposition techniques – see e.g., [7] – can be used to obtain a decentralized protocol based only on communication with adjacent nodes that converges to a point $\{\mathbf{U}_k\}_{k=1}^K \in \mathcal{S}'\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$. This, we pursue next.

## 4. STOCHASTIC SCHEDULING-ROUTING PROTOCOL

To obtain a protocol based on one-hop communications with adjacent nodes, we start by defining the index set

$$a(j) := \{i : R_{ij} \neq 0, i \neq j\} \tag{12}$$

so that the nodes adjacent to $U_j$ are those nodes $\mathbf{A}_j := \{U_i\}_{i\in a(j)}$ that can decode $U_j$'s transmission with non-zero probability. We assume that adjacency is a symmetric relationship, i.e., $i \in a(j)$ if and only if $j \in a(i)$, which is necessarily true in any practical network. We also define the index set of destinations $U_k$ that are served by $U_j$ as

$$c(j) := \{k : T_{kij} \neq 0,\ \text{for some } i \in c(j)\}. \tag{13}$$

With these definitions we can rewrite the rate constraint $\boldsymbol{\rho}_k \leq (\mathbf{I}-\mathbf{L}_k)\mathbf{1}$ as

$$\rho_{kj} \leq 1 - \sum_{i\in a(j)} L_{kji} = \sum_{i\in a(j)} L_{kji} - \sum_{i\in a(j)} L_{kij} \tag{14}$$

where in obtaining the equality we used that $1-L_{kjj} = \sum_{i\in c(j)} L_{kji}$. Since we eliminated $K_{kjj}$ from the rate constraint we can now use (3) to obtain

$$\rho_{kj} \leq \sum_{i\in a(j)} U_{kji}R_{ji} - \sum_{i\in a(j)} U_{kij}R_{ij}. \tag{15}$$

Consider now the vectors $\mathbf{u}_{kj} := \mathbf{U}_{ka(j)j}$ containing the probabilities of $U_j$ transmitting to adjacent nodes in $a(j)$ packets with final destination $U_k$. Likewise, let $\mathbf{u}'_{kj} := \mathbf{U}_{kja(j)}$ be the probabilities with which $U_j$'s adjacent nodes send packets to $U_j$ with destination $U_k$. Upon defining $\mathbf{r}_j = \mathbf{R}_{a(j)j}$ and $\mathbf{s}_j = \mathbf{R}_{ja(j)}$ containing the corresponding correct detection probabilities we can rewrite (15) as

$$\rho_{kj} \leq \mathbf{r}_j^T\mathbf{u}_{kj} - \mathbf{s}_j^T\mathbf{u}'_{kj} \tag{16}$$

We can now define a scheduling-routing protocol as one that lets every terminal $U_j$ find a set of vectors $\{\mathbf{u}_{kj}\}_{k\in c(j)}$ satisfying the constraints [cf. (11) and (16)]

$$\rho_{kj} \leq \mathbf{r}_j^T\mathbf{u}_{kj} - \mathbf{s}_j^T\mathbf{u}'_{kj} \tag{17}$$

$$\sum_{k\in c(j)} \mathbf{t}_k^T\mathbf{1} \leq \mu_j, \qquad \mathbf{u}_{kj} \geq 0 \tag{18}$$

The constraints in (18) involve only $U_j$'s transmission probabilities and can be locally enforced in the sense that $U_j$ can find vectors $\{\mathbf{t}_k\}_{k\in c(j)}$ satisfying (18). The constraint in (17), however, also involves transmission probabilities of $U_j$'s adjacent nodes. To deal with the latter we introduce Lagrange multipliers $p_{kj}$ associated with the constraint $\rho_{kj} \leq \mathbf{r}_j^T\mathbf{u}_{kj} - \mathbf{s}_j^T\mathbf{u}'_{kj}$. Define the vector of multipliers $\mathbf{p}_k := [p_{k1}, \ldots, p_{kJ}]^T$, the vector of adjacent multipliers $\mathbf{p}'_{kj} := \mathbf{p}_{kc(j)}$ and consider the following iteration:

**Primal iteration:** For given local multipliers $p_{kj}$ and adjacent multipliers $\mathbf{p}'_{kj}(n)$ define the primal iterate as

$$\mathbf{u}_{kj}(n) = \frac{\mathbf{r}_j}{\alpha} \cdot \left[\mathbf{1}p_{kj}(n) - \mathbf{p}'_{kj}(n) - \beta_j\mathbf{1}\right]^+ \tag{19}$$

where $\beta_j > 0$ is a scaling constant so that $\sum_{k\in c(j)} \mathbf{t}_{kj}^T(n)\mathbf{1} = \mu_j$ and $[\,]^+$ denotes projection in the positive orthant.

**Dual iteration:** For given local and neighboring iterates $\mathbf{u}_{kj}(n)$ and $\mathbf{u}'_{kj}(n)$ update the dual iterates by

$$p_{kj}(n+1) = \left[p_{kj}(n) + \gamma\left(\rho_{kj} - \mathbf{r}_j^T\mathbf{u}_{kj}(n) + \mathbf{s}_j^T\mathbf{u}'_{kj}(n)\right)\right]^+. \tag{20}$$

The iteration described by (19) and (20) is a dual decomposition algorithm [2] to solve the feasibility problem [3] in (17)-(18). As such, it can be shown to converge to a solution of the feasibility problem as we state in the following proposition.

**Algorithm 1** Primal iteration

1: Receive dual iterates $\mathbf{p}'_{kj}(n)$ from adjacent nodes $\mathbf{A}_j$
2: Update primal iterates using (19):
$\mathbf{u}_{kj}(n) = \frac{\mathbf{r}_j}{2\alpha} \cdot \left[\mathbf{1}p_{kj}(n) - \mathbf{p}'_{kj}(n) - \beta_j \mathbf{1}\right]^+$
3: Transmit primal iterates $\mathbf{u}_{kj}(n)$ to adjacent nodes $\mathbf{A}_j$
4: Routing probabilities for current slot:
$\mathbf{t}_{kj}(n) = \left(\sum_{k\in c(j)} \mathbf{u}_{kj}^T(n)\mathbf{1}\right)^{-1} \mathbf{u}_{kj}(n).$

**Algorithm 2** Dual iteration

1: Receive primal iterates $\mathbf{u}'_{kj}(n)$ from adjacent nodes $\mathbf{A}_j$
2: Update dual iterates using (20):
$p_{kj}(n+1) = \left[p_{kj}(n) + \gamma\left(\rho_{kj} - \mathbf{r}_j^T \mathbf{u}_{kj}(n) + \mathbf{s}_j^T \mathbf{u}'_{kj}(n)\right)\right]^+$
3: Transmit dual iterates $p_{kj}(n+1)$ to adjacent nodes $\mathbf{A}_j$

**Proposition 3** *Consider a given set of rates* $\{\boldsymbol{\rho}_k\}_{k=1}^K$ *and iterates defined by* (17)-(18). *If the stability region* $\mathcal{S}\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right)$ *is non empty, then, for sufficiently small* $\gamma > 0$

$$\lim_{n\to\infty} \{\mathbf{U}_k(n)\}_{k=1}^K := \{\mathbf{U}_{k\infty}\}_{k=1}^K \in \mathcal{S}'\left(\{\boldsymbol{\rho}_k\}_{k=1}^K\right). \tag{21}$$

When the stability region is nonempty Proposition 3 ensures that the iteration (17)-(18) succeeds in finding vectors $\{\mathbf{u}_{kj}(n)\}_{k\in c(j)}$ that satisfy the conditions in (17) and (18), from were we conclude that the corresponding $\{\mathbf{U}_k(n)\}_{k=1}^K \in \mathcal{S}'\left\{\{\boldsymbol{\rho}_k\}_{k=1}^K\right\}$ as $n \to \infty$. We then use the transformation

$$\mathbf{t}_{kj}(n) = \left(\sum_{k\in c(j)} \mathbf{u}_{kj}^T(n)\mathbf{1}\right)^{-1} \mathbf{u}_{kj}(n). \tag{22}$$

Defining the matrices $\{\mathbf{T}_k(n)\}_{k=1}^K$ with columns $\mathbf{T}_{ka(j)j}(n) = \mathbf{t}_{kj}(n)$ we have that $\{\mathbf{T}_k\}_{k=1}^K(n) \in \mathcal{S}''\left\{\{\boldsymbol{\rho}_k\}_{k=1}^K\right\}$ [cf. (9), (22) and $\{\mathbf{U}_k(n)\}_{k=1}^K \in \mathcal{S}'\left\{\{\boldsymbol{\rho}_k\}_{k=1}^K\right\}$]. Using Proposition 2 we then conclude that $\{\mathbf{T}_k(n)\}_{k=1}^K \in \mathcal{S}\left\{\{\boldsymbol{\rho}_k\}_{k=1}^K\right\}$ from where Proposition 1 guarantees that routing with probabilities $\mathbf{t}_{kj}(n)$ ensures that all queues in the network are stable. The corresponding stochastic routing-scheduling protocol is shown in Algorithms 1 and 2, where steps 1 and 3 ensure that variables are properly communicated.

### 4.1. Interpretation

As is often the case in dual decomposition algorithms, the multipliers $p_{kj}$ can be interpreted as prices. Let us thus define $p_{kj}$ as the price that $U_j$ charges for handling packets destined to $U_k$. The profit that $U_j$ obtains from this $k$-th flow is

$$\mathcal{P}_{kj} = p_{kj}\rho_{kj} - \mathbf{u}_{kj}^T(\mathbf{r}_j \cdot \mathbf{p}'_{kj}) + p_{kj}\mathbf{s}_j^T\mathbf{u}'_{kj} \tag{23}$$

where the first term is charged to the upper layers, the second term is payed to adjacent nodes that receive $U_j$'s packets and the last term is charged to adjacent nodes. For given prices, $\mathbf{U}_j$ wants to maximize $\mathcal{P}_j := \sum_{k\in c(j)} \mathcal{P}_{kj}$.

However, $U_j$'s profits are partly determined by $\mathbf{t}'_{kj}$, a variable chosen by $U_j$'s neighbors. Since, it cannot choose the latter directly, $U_j$ resorts to optimizing a bound on the profits. Considering that in order to ensure stability it should be $\rho_{kj} + \mathbf{s}_j^T\mathbf{t}'_{kj} \leq \mathbf{t}_{kj}^T\mathbf{r}_j$, the profits of $U_j$ are bounded by

$$\begin{aligned}\mathcal{P}_{kj} \leq \mathcal{Q}_{kj} &= p_{kj}\mathbf{u}_{kj}^T\mathbf{r}_j - \mathbf{u}_{kj}^T(\mathbf{r}_j \cdot \mathbf{p}'_{kj}) \\ &= \mathbf{u}_{kj}^T\left(p_{kj} - \mathbf{p}'_{kj} \cdot \mathbf{r}_j\right)\end{aligned} \tag{24}$$

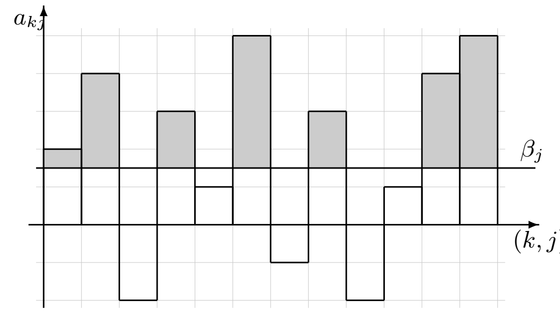


**Fig. 1**. The primal iteration in (19) is a form of waterfilling on the price differences $R_{ij}/\alpha\,[p_{kj}(n) - p_{ki}(n)]$ (turn the page around). The sum of the gray squares is the amount of water $\mu_j$ with $\beta_j$ denoting the water level. When $p_{kj}(n) - p_{ki}(n) < \beta_j$ no transmission probability is assigned to $U_i$ for this flow. The water level is always positive.

The potential profit $\mathcal{Q}_{kij}$ that $U_j$ expects to make by sending packets through $U_i$ is

$$\mathcal{Q}_{kij} = U_{kij}R_{ij}\left[p_{kj}(n) - p_{ki}(n)\right] \tag{25}$$

which has the simple interpretation of being the expected number of packets that $U_i$ receives from $U_j$ times the difference between the price $p_{kj}(n)$ charged by $U_j$ and the price $p_{ki}(n)$ charged by $U_i$.

Lacking control over its profits $\mathcal{P}_{kj}$, $U_j$ aims to maximize its potential profits $\mathcal{Q}_{kij}$. This is what (19) does. The transmission probability that $U_j$ assigns to $U_i$ is

$$u_{kij}(n) = \frac{R_{ij}}{\alpha}\left[p_{kj}(n) - p_{ki}(n) - \beta_j\right]^+. \tag{26}$$

Ignoring $\beta_j$ we see that the transmission probability assignment is proportional to the potential profit that $U_j$ expects to make by sending packets to $U_i$. The constant $\beta_j$ simply enforces a form of waterfilling as illustrated in Fig. 1.

Eq. (20) is the corresponding price update. If $\rho_{kj} + \mathbf{s}_j^T\mathbf{t}'_{kj} < \mathbf{t}_{kj}^T\mathbf{r}_j$, then $U_j$ is receiving less packets than those expected. That implies the price $U_j$ charges for handling packets is excessive and $p_{kj}$ is decreased by an amount proportional to $\mathbf{t}_{kj}^T\mathbf{r}_j - \rho_{kj} + \mathbf{s}_j^T\mathbf{t}'_{kj}$. If on the other hand $\rho_{kj} + \mathbf{s}_j^T\mathbf{t}'_{kj} > \mathbf{t}_{kj}^T\mathbf{r}_j$, then $U_j$ is receiving more packets than expected, implying that the price should be correspondingly increased.

## 5. SIMULATIONS

We consider a wireless ad-hoc network with $J = 100$ nodes randomly deployed in a rectangle of dimensions 5Km. × 3Km.. Every node transmits with probability $\mu_j = 0.5$ and wants to deliver packets to a randomly chosen destination at a rate $\rho_{jk(j)} = 0.1$ packets per unit time. The reliability matrix $\mathbf{R}$ is represented in Fig. 2.

The convergence of Algorithms 1-2 to a point in the stability region is illustrated in Fig. 3. Note that after 20 iterations, all nodes are getting a rate of at least $\rho_{jk(j)}(n) = 0.08$. Interpreting convergence as the point at which the achieved rate is 90% of the required rate ($\rho_{jk(j)}(n) = 0.09$ in our example) it takes about 40 iterations for the slowest node to converge. After 140 iterations all rates $\rho_{jk(j)}(n)$ are within 3% of the required rate.

Simulation of packet transmissions is depicted in Fig. 4 where we show the routes followed by 100 packets transmitted between a

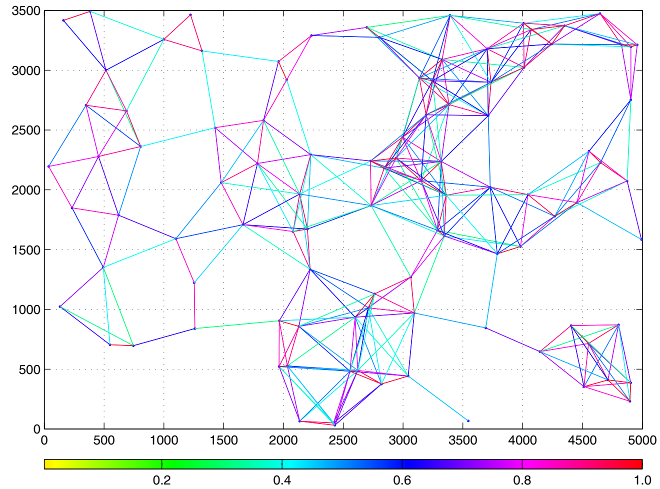


**Fig. 2**. A randomly generated ad-hoc network with 100 nodes uniformly distributed in a rectangle of $5\text{Km.} \times 3\text{Km.}$; the color index represents the value of $R_{ij}$ that is generated according to the empirical distribution in [4] (only values $R_{ij} > 0.3$ are shown).

randomly selected pair of nodes. It is interesting how the proposed protocol divides the load among many different routes.

## 6. CONCLUSIONS

We introduced a stochastic joint-scheduling routing protocol to support a set of required rates $\{\boldsymbol{\rho}_k\}_{k=1}^{K}$. We showed that as long as there exists a set of routes and schedules leading to stable queues, the proposed protocol summarized by Algorithms 1 and 2 ensures that all queues in the system are stable.

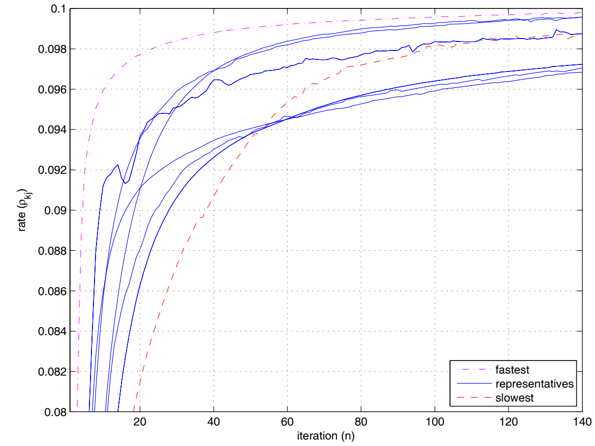


**Fig. 3**. Offered rate $\rho_{jk(j)}(n)$ converges to $\rho_{jk(j)}$ as $n \to \infty$. All nodes reach $90\%$ of the required rate in around 40 iterations.

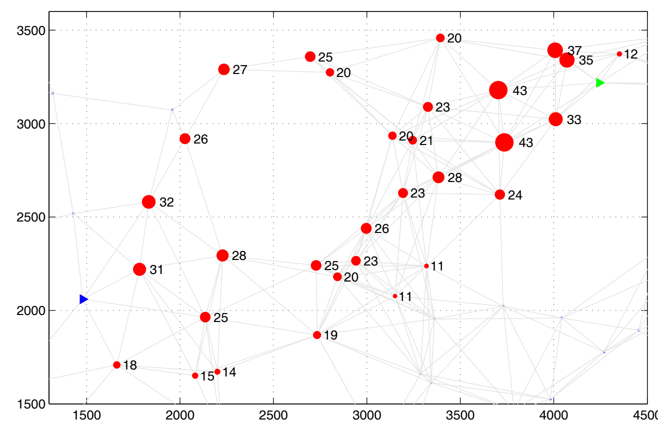

**Fig. 4**. Number of packets transmitted by individual nodes when 100 packets are transmitted from the node marked with the blue (left) triangle to the node marked by the green (right) triangle. The size of the circles is proportional to the number of packets transmitted by the node on behalf of this communication. The network load is divided among many different routes.

## 7. REFERENCES


[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," *ACM SIGCOMM Computer Commun. Review*, vol. 34, pp. 121–132, October 2004.

[2] D. Bertsekas and R. Gallager, *Data networks*. Prentice-Hall, Inc., 2nd ed., 1992.

[3] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[4] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. Int. ACM Conf. Mobile Computing, Networking*, pp. 134–146, San Diego, CA, September 14-19, 2003.

[5] G. Dimic, N. D. Sidiropoulos, and R. Zhang, "Signal processing and queuing tools for MAC-PHY cross-layer design," *IEEE Signal Process. Mag.*, vol. 21, pp. 40–50, September 2004.

[6] H. Lundgren, E. Nordstrom, and C. Tschudin, "The gray zone problem in IEEE 802.11b based ad hoc networks," *ACM SIGMOBILE Mobile Computing and Commun. Review*, vol. 3, pp. 104–105, July 2002.

[7] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 1439–1451, August 2006.

[8] A. Ribeiro, Z.-Q. Luo, N. D. Sidiropoulos, and G. B. Giannakis, "A general optimization framework for stochastic routing in wireless multihop networks," *IEEE Trans. Signal Process.*, January 2007 (submitted). Available at http:// www.ece.umn.edu/ users/ aribeiro/ research/ pubs.html.

[9] A. Ribeiro, N. D. Sidiropoulos, and G. B. Giannakis, "Optimal distributed stochastic routing algorithms for wireless multihop networks," *IEEE Trans. Wireless Commun.*, October 2006 (submitted). Available at http:// www.ece.umn.edu/ users/ aribeiro/ research/ pubs.html.

[10] L. Tassiulas, "Adaptive back-pressure congestion control based on local information," *IEEE Trans. Autom. Control*, vol. 40, pp. 236–250, February 1995.

[11] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Trans. Inf. Theory*, vol. 43, pp. 1067–1073, May 1997.