DISTRIBUTED ROUTING ALGORITHMS FOR WIRELESS MULTIHOP NETWORKS

Alejandro Ribeiro and Georgios B. Giannakis

Dept. of ECE, University of Minnesota

Nikolaos D. Sidiropoulos

Dept. of ECE, Technical University of Crete

ABSTRACT

We introduce distributed algorithms to find rate-optimal routes based on local knowledge of the pairwise error probability (reliability) matrix. The distributed algorithms are built by (re)-formulating optimization problems amenable to application of dual decomposition techniques. Convergence of our algorithms to the optimal routing matrix is guaranteed under mild conditions. Many rate-optimality criteria of practical interest can be casted in our framework including maximization of: i)worst user's rate; ii) weighted sum of rates; iii) product of rates; and iv) relay network rate. We test robustness of our algorithms to node mobility.

Keywords: Communication systems routing, Wireless networks, Optimization methods, Linear programming, Distributed computing.

1. INTRODUCTION

Capitalizing on the potential energy savings of multihop wireless networks requires solving the challenging problem of finding multihop routes according to properly defined optimality criteria. Existing routing protocols/algorithms are built on our accumulated knowledge of routing in wired networks. Consequently, the usual approach is to i) define a communication radius for each node; ii) draw the corresponding connectivity graph; and iii) utilize network optimization tools to find pertinent routes. While definitely valuable as a first approach, a graph is not an accurate model of a wireless network [3]. In a recent paper we introduced a framework to design stochastic routing algorithms/protocols using the reliability matrix $\mathbf{R}$ whose (i,j)-th entry R_{ij} represents the probability that a packet transmitted from the j-th user U_j is correctly received by the i-th user U_i [6]; see also Fig. 1.

While offering a more accurate model of the broadcast and unreliable wireless channel, the usefulness of a model based on $\mathbf{R}$ depends on the algorithmic complexity of finding optimal routes. Enticingly, many interesting optimality criteria lead to routing algorithms in the form of convex optimization problems [6]. Even though this ensures manageable complexity, it requires $\mathbf{R}$ to be available at a central location. This entails: i) a large communication cost to collect $\mathbf{R}$ and percolate the optimal routing matrix; ii) considerable delay to compute the optimal routes; and iii) lack of resilience to changes in $\mathbf{R}$, a problem particularly important in mobile scenarios.

Distributed algorithms, whereby nodes iteratively interchange variables only with one-hop neighbors tackle precisely these problems. The goal of this paper is to show that the optimal routing problems in [6] can be solved by an iterative distributed algorithm whereby i) node U_j has access only to link reliabilities for transmission to and from other nodes (the j-th row and column of $\mathbf{R}$); ii) U_j interchanges messages only with one-hop neighbors, defined as the set of terminals with non-zero probability of decoding U_j's packets; and iii) as time progresses U_j computes its optimal routing probabilities.

1.1. Stochastic routing in wireless multihop networks

Consider a wireless network with $J+1$ user nodes $\{U_j\}_{j=1}^{J+1}$ in which the first J users $\{U_j\}_{j=1}^{J}$ collaborate in routing packets to the destination $D \equiv U_{J+1}$. The physical and medium access layers are such that if a packet is transmitted by U_j it is correctly *received* by U_i with probability R_{ij} that we arrange in the matrix $\mathbf{R}$. Packets are stochastically routed according to probabilities T_{ij} arranged in the matrix $\mathbf{T}$. When a user terminal U_j decides to transmit a packet it selects a random terminal as the intended destination with U_i chosen with probability T_{ij}. If the transmission is successful the packet moves to U_i's queue, if not it is kept by U_j that attempts transmission, possibly to a different node, at a later time. To capture the evolution of packets through the network we define a matrix $\mathbf{K}$ whose elements K_{ij} represent the probability that a packet moves from U_j's queue to U_i's queue. For $i \neq j$ a packet moves from U_j to U_i if and only if it is routed towards U_i *and* is correctly decoded at U_i. Since these two events are independent,

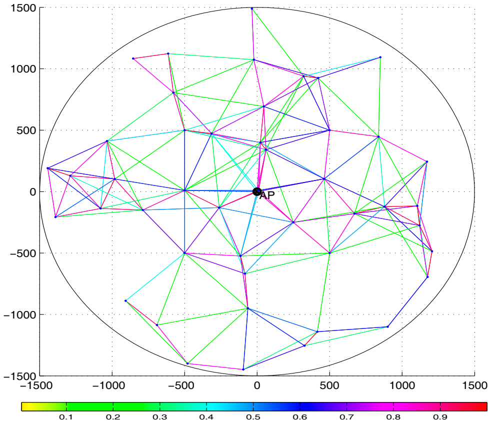

Fig. 1. Connectivity graph for a network with $J = 40$ user terminals. The color index represents the value of R_{ij}.

$$K_{ij} = T_{ij} R_{ij} \text{ for } i \neq j, \qquad \mathbf{K}^T \mathbf{1} = \mathbf{1}, \qquad \mathbf{T}^T \mathbf{1} = \mathbf{1}. \tag{1}$$

The last two constraints are because $\mathbf{K}$ and $\mathbf{T}$ are stochastic matrices.

Also, let $\boldsymbol{\rho} := [\rho_1, \ldots, \rho_J]^T$ denote the vector of packet arrival rates and $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_J]^T$ the rate of departures that we constraint by $\mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}$. Defining $\mathbf{K}_0$ as the $J \times J$ upper left submatrix of $\mathbf{K}$ it is not difficult to see that we can relate $\boldsymbol{\rho}$ and $\boldsymbol{\alpha}$ by [6]

$$\boldsymbol{\rho} = (\mathbf{I} - \mathbf{K}_0)\boldsymbol{\alpha} \tag{2}$$

With $\mathbf{R}$ available at the AP, we look for routes maximizing a measure of the arrival rate vector $\boldsymbol{\rho}$. Letting $f(\boldsymbol{\rho}) : \mathbb{R}^J \to \mathbb{R}$ be the function used to compare arrival rate vectors $\boldsymbol{\rho}$, the optimal routing matrix $\mathbf{T}^*$ is given as the solution of the generic optimization problem (symbols $\preceq$ and $\succeq$ denote componentwise inequalities between vectors):

$$\begin{aligned}(\mathbf{K}^*, \mathbf{T}^*) = \arg\max \ & f[(\mathbf{I} - \mathbf{K}_0)\boldsymbol{\alpha}] \\ \text{s.t.} \ & K_{ij} = R_{ij} T_{ij} \text{ for } i \neq j, \ \ \mathbf{K}^T\mathbf{1} = \mathbf{1}, \ \ \mathbf{T}^T\mathbf{1} = \mathbf{1} \\ & \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}. \end{aligned} \tag{3}$$

Finding efficient methods to solve (3) is challenging for general $f(\boldsymbol{\rho})$. However, for any $f(\boldsymbol{\rho})$ that is concave and monotonically non-decreasing in each component[1] (3) can be transformed to an equivalent convex optimization problem. Indeed, [6] establishes that for functions that are monotonically non-decreasing in each component there exists an optimal solution of (3) with $\boldsymbol{\alpha} = \mathbf{1}$, thus implying that (3) can be rewritten as

$$\begin{aligned}(\mathbf{K}^*, \mathbf{T}^*) = \arg\max \ & f[(\mathbf{I} - \mathbf{K}_0)\mathbf{1}] \\ \text{s.t.} \ & K_{ij} = R_{ij} T_{ij} \text{ for } i \neq j, \ \ \mathbf{K}^T\mathbf{1} = \mathbf{1}, \ \ \mathbf{T}^T\mathbf{1} = \mathbf{1}. \end{aligned} \tag{4}$$

[1] We say $g(\mathbf{v})$ is monotonically non-decreasing in each component if for vectors $\mathbf{v}^1, \mathbf{v}^2$ with $v_j^1 \leq v_j^2$ and $v_i^1 = v_i^2$ for $i \neq j$, $g[\mathbf{v}^1] \leq g[\mathbf{v}^2]$.

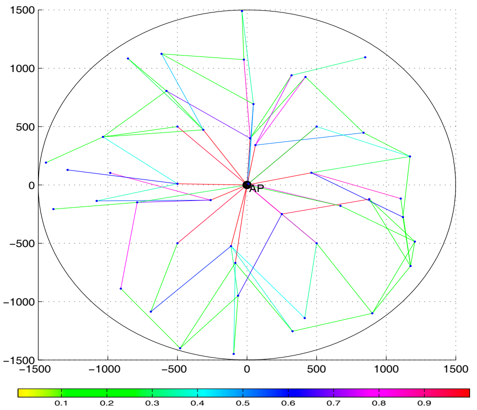

Fig. 2. Optimal routes for the max-min criterion.

Concavity of $f(\boldsymbol{\rho})$ further implies that the argument in (4) is concave, implying that (4) is a convex optimization problem solvable in polynomial time using interior point methods.

2. A SEPARABLE PROBLEM

To design distributed algorithms we introduce optimization problems that can be implemented in a distributed fashion and whose solution coincides with (4). For simplicity of exposition we adopt as optimality criterion the rate of the worst user $f(\boldsymbol{\rho}) = \min_{j\in[1,J]} \rho_j$ leading to

$$(\mathbf{K}^*, \mathbf{T}^*) = \arg\max \min_{j\in[1,J]} [(\mathbf{I}-\mathbf{K}_0)\mathbf{1}]_j \qquad (5)$$
$$\text{s.t. } K_{ij} = R_{ij}T_{ij} \text{ for } i \neq j, \;\; \mathbf{K}^T\mathbf{1} = \mathbf{1}, \;\; \mathbf{T}^T\mathbf{1} = \mathbf{1}.$$

To reduce the number of variables we eliminate some constraints in (5). To this end, define the set $c(j) := \{i : R_{ij} > 0; i \neq j, i \in [1, J+1]\}$ containing the indices of terminals U_i that can decode U_j's transmission with non-zero probability. Likewise, define $r(j) := \{i : R_{ji} > 0; i \neq j, i \in [1, J+1]\}$ as the set of nodes that U_j decodes with non-zero probability. We can now write the rate of the j-th user as

$$\rho_j = [(\mathbf{I}-\mathbf{K}_0)\mathbf{1}]_j = 1 - K_{jj} - \sum_{i\in r(j)} K_{ji} = \sum_{i\in c(j)} K_{ij} - \sum_{i\in r(j)} K_{ji} \qquad (6)$$

where in the second equality we used the constraint $\mathbf{K}^T\mathbf{1} = \mathbf{1}$. Upon substituting $K_{ij} = R_{ij}T_{ij}$, (6) becomes

$$\rho_j = \sum_{i\in c(j)} R_{ij}T_{ij} - \sum_{i\in r(j)} R_{ji}T_{ji}. \qquad (7)$$

For a more compact notation define the vectors $\mathbf{t}_j := \mathbf{T}_{c(j)j}$ and $\mathbf{t}_j^r = \mathbf{T}_{jc(j)}^T$ containing the non-zero elements of the j-th row and column of $\mathbf{T}$, respectively. We also define $\mathbf{r}_j := \mathbf{R}_{c(j)j}$ and $\mathbf{s}_j := \mathbf{R}_{jc(j)}^T$ so that

$$\rho_j = \mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{t}_j^r. \qquad (8)$$

Vectors $\mathbf{r}_j$, and $\mathbf{s}_j$ are constant and known at node U_j. Indeed, $\mathbf{s}_j := \mathbf{R}_{jc(j)}^T$ contains the probabilities of U_j decoding other nodes $U_i \neq U_j$ that U_j can easily estimate. The probabilities $\mathbf{r}_j := \mathbf{R}_{c(j)j}$ of other nodes decoding U_j's packets are fed-back from its one-hop neighbors.

Using (8) and noting that the constraint $\mathbf{T}^T\mathbf{1} = \mathbf{1}$ is equivalent to the set of constraints $\{\mathbf{t}_j^T\mathbf{1} = 1\}_{j=1}^J$, we can rewrite max-min in (5) as

$$\mathbf{T}^* = \arg\max \; w$$
$$\text{s.t. } w \preceq \mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{t}_j^r = \rho_j, \;\; \mathbf{t}_j^T\mathbf{1} = 1, \;\; \mathbf{0} \preceq \mathbf{t}_j. \qquad (9)$$

Even though (9) is written in terms of local variables ($\mathbf{r}_j$), local constants ($\mathbf{t}_j$, $\mathbf{s}_j$), and neighboring variables ($\mathbf{t}_j^r$), it is not yet in a separable form. Indeed, note that i) the variable w is constrained to be smaller than the rates ρ_j of the J terminals and in that sense its optimization requires access to all the variables; and ii) computing ρ_j requires access to the local variables $\mathbf{t}_j$ and neighboring variables $\mathbf{t}_j^r$.

We thus introduce local variables w_j and $\mathbf{u}_j$ that we regard as U_j's estimates of (the global variable) w and (the neighboring variable) $\mathbf{t}_j^r$, and introduce equality constraints $\mathbf{u}_j = \mathbf{t}_j^r$ and $w = w_j, \forall j \in [1, J]$. Using these (local) variables we can write the constraint in (9) as

$$w_j \preceq \rho_j = \mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{u}_j; \quad \mathbf{u}_j = \mathbf{t}_j^r, \quad w = w_j. \qquad (10)$$

Finally, replace w in (9) by $w = (\sum_{j=1}^J w_j)/J$. If we further assume that there is a non-zero probability for a multi-hop route connecting any pair of nodes, the set of constraints $\{w_j = w_i \; \forall i \in c(j)\}_{j=1}^J$ is equivalent to requiring $w_i = w_j, \forall\, i, j \in [1, J]$. We thus recast (9) as

$$\mathbf{T}^* = \arg\max \; \frac{1}{J}\sum_{j=1}^J w_j$$
$$\text{s.t. } w_j \preceq \mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{u}_j, \;\; \mathbf{t}_j^T\mathbf{1} = 1, \;\; \mathbf{0} \preceq \mathbf{t}_j,$$
$$\mathbf{t}_j^r = \mathbf{u}_j, \;\; w_j = w_i \; \forall i \in c(j) \qquad (11)$$

where the maximization is over $\mathbf{T}$, $\{\mathbf{u}_j\}_{j=1}^J$, and $\mathbf{w} := [w_1, \ldots, w_J]^T$.

Note that $w_j \preceq \mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{u}_j$, $\mathbf{t}_j^T\mathbf{1} = 1$, and $\mathbf{0} \preceq \mathbf{t}_j$ involve the variables $\mathbf{x}_j := (w_j, \mathbf{t}_j, \mathbf{u}_j)$ only, and can thus be locally enforced, meaning that it is possible for U_j to find values of $\mathbf{x}_j$ satisfying these constraints. The equality constraints $\mathbf{t}_j^r = \mathbf{u}_j$ and $w_j = w_i$ for all $i \in c(j)$ cannot be enforced locally but it is important to note that they relate neighboring variables only. Readers familiar with dual decomposition techniques – see, e.g., [2, Sec. 3.4.2] and [4] – may notice that the form of (11) lends itself to distributed optimization of the type we will elaborate on in Section 3.

2.1. Generic problem formulation

Equivalence of (5) and (11) is not unique to max-min optimal routing since the same steps can be applied to reformulate many optimization problems. Indeed, for a given convex set $\mathcal{X}_j$ define the routing problem

$$\mathbf{T}^* = \arg\max_{\mathbf{X}} \; \mathbf{w}^T\mathbf{1} \qquad (12)$$
$$\text{s.t. } \mathbf{x}_j := (w_j, \mathbf{t}_j, \mathbf{u}_j) \in \mathcal{X}_j; \;\; \mathbf{t}_j^r = \mathbf{u}_j; \; \mathbf{v}_j = w_j\mathbf{1},$$

where $\mathbf{v}_j := \mathbf{w}_{c(j)}$ contains the variables w_j of U_j's neighbors. The formulation in (12) encompasses all the routing problems defined in in [6], with the set $\mathcal{X}_j$ specifying the corresponding optimality criterion:

Max-min optimal rate. This is the problem considered in detail in Section 2 and can be obtained from (12) by defining the set

$$\mathcal{X}_j^1 = \left\{\mathbf{x}_j : w_j \leq \mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{u}_j, \;\; \mathbf{0} \preceq \mathbf{t}_j, \;\; \mathbf{t}_j^T\mathbf{1} = 1\right\}. \qquad (13)$$

Additional convex constraints can be added to the definition of $\mathcal{X}_j$. Since we know that $\mathbf{u}_j$ is a vector of probabilities using the set $\mathcal{X}_j = \mathcal{X}_j^1 \cap \{\mathbf{x}_j : \mathbf{0} \preceq \mathbf{u}_j \preceq \mathbf{1}\}$ is equivalent to using $\mathcal{X}_j^1$. Preventing the components of $\mathbf{u}_j$ to become too large improves numerical stability.

Optimal weighted sum-rate. To maximize a weighted sum of rates, i.e., $f(\boldsymbol{\rho}) = \boldsymbol{\beta}^T\boldsymbol{\rho}$ with $\boldsymbol{\beta} := [\beta_1, \ldots, \beta_J]^T \succeq \mathbf{0}$, we define the set

$$\mathcal{X}_j^2 := \left\{\mathbf{x}_j : w_j = \beta_j(\mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{u}_j), \;\; \mathbf{0} \preceq \mathbf{t}_j, \;\; \mathbf{t}_j^T\mathbf{1} = 1\right\} \qquad (14)$$

and consider the optimization problem

$$\mathbf{T}^* = \arg\max \; \mathbf{w}^T\mathbf{1}$$
$$\text{s.t. } \mathbf{x}_j := (w_j, \mathbf{t}_j, \mathbf{u}_j) \in \mathcal{X}_j^2, \;\; \mathbf{t}_j^r = \mathbf{u}_j \qquad (15)$$

which amounts to dropping the constraint $\mathbf{v}_j = w_j\mathbf{1}$ in (12). For this criterion $w_j = \beta_j\rho_j$ [cf. (8) and (14)]. Extra constraints can be dealt with by modifying the set $\mathcal{X}_j^2$ as before; e.g., a minimum acceptable rate $\rho_j^{\min}$ for terminal U_j can be ensured by $\mathcal{X}_j := \mathcal{X}_j^2 \cap \{w_j/\alpha_j \geq \rho_j^{\min}\}$.

Optimal product of rates. Maximizing the product of rates prevents solutions in which some users receive very small rates. The function to be maximized in this case is $f(\boldsymbol{\rho}) = \prod_{j=1}^J \rho_j$. Since the logarithm is monotonically increasing the concave function $f(\boldsymbol{\rho}) = \sum_{j=1}^J \log(\rho_j)$ can be used instead. To cast this problem it suffices to define

$$\mathcal{X}_j^3 := \left\{\mathbf{x}_j : w_j \leq \log[\mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{u}_j], \;\; \mathbf{0} \preceq \mathbf{t}_j, \;\; \mathbf{t}_j^T\mathbf{1} = 1\right\} \qquad (16)$$

and replace $\mathcal{X}_j^2$ by $\mathcal{X}_j^3$ in (15). The local components of the argument w_j denote the logarithm of the local rate.

Optimal rate with relays. In a relay network a group of terminals collaborate in relaying traffic on behalf of a designated active user U_{j_0}. The relay network maximizing ρ_{j_0} can be found by solving (12) with

$$\mathcal{X}_j^3 = \Big\{\mathbf{x}_j : 0 = \mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{u}_j,\ \mathbf{0} \preceq \mathbf{t}_j,\ \mathbf{t}_j^T\mathbf{1} = 1\Big\},\ j \neq j_0$$
$$\mathcal{X}_{j_0}^3 = \Big\{\mathbf{x}_{j_0} : w_{j_0} = \mathbf{r}_{j_0}^T\mathbf{t}_{j_0} - \mathbf{s}_{j_0}^T\mathbf{u}_{j_0},\ \mathbf{0} \preceq \mathbf{t}_{j_0},\ \mathbf{t}_{j_0}^T\mathbf{1} = 1\Big\}. \quad (17)$$

Here, w_j is the local estimate of the source's rate ρ_{j_0} at terminal U_j.

3. DISTRIBUTED ROUTING ALGORITHMS

Since $\mathcal{X}_j$ is convex we optimize the dual function that, as we will show in this section, exhibits a separable structure; see also [2, Sec. 3.4.2] and [5]. Associate, thus, Lagrange multipliers $\boldsymbol{\lambda}_j$ with the constraints $\mathbf{t}_j^r - \mathbf{u}_j = \mathbf{0}$ and $\boldsymbol{\mu}_j$ with $\mathbf{v}_j - w_j\mathbf{1} = \mathbf{0}$ to form the Lagrangian

$$\mathcal{L}(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{M}) = -\mathbf{w}^T\mathbf{1} + \sum_{j=1}^{J}\Big[(\mathbf{t}_j^r - \mathbf{u}_j)^T\boldsymbol{\lambda}_j + (\mathbf{v}_j - w_j\mathbf{1})^T\boldsymbol{\mu}_j\Big] \quad (18)$$

which is defined in the feasible set of primal variables $\{\mathbf{x}_j \in \mathcal{X}_j\}_{j=1}^J$. Matrices $\boldsymbol{\Lambda}$ and $\mathbf{M}$ are defined to have the same sparsity pattern of $\mathbf{T}$; the dual variables in (18) are respectively given by $\boldsymbol{\lambda}_j = \boldsymbol{\Lambda}_{jc(j)}$ and $\boldsymbol{\mu}_j = \mathbf{M}_{c(j)j}$. We assume that $\boldsymbol{\lambda}_j$ and $\boldsymbol{\mu}_j$ are kept by terminal U_j.

The Lagrangian in (18) is used to obtain the dual function

$$g(\boldsymbol{\Lambda}, \mathbf{M}) = \min_{\{\mathbf{x}_j \in \mathcal{X}_j\}_{j=1}^J} \mathcal{L}(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{M}) \quad (19)$$

which in turn leads to the dual problem defined as the unconstrained maximization of $g(\boldsymbol{\Lambda}, \mathbf{M})$. Since strong duality holds, we have

$$\mathbf{1}^T\mathbf{w}^* = -\max_{\boldsymbol{\Lambda}, \mathbf{M}} g(\boldsymbol{\Lambda}, \mathbf{M}). \quad (20)$$

The problem in (20) is an unconstrained optimization problem that can be solved with a gradient ascent algorithm. However, since the dual function $g(\boldsymbol{\Lambda}, \mathbf{M})$ is not always differentiable a generalization of the gradient, the so called subgradient is used instead.

Definition 1 *We say that $\nabla_{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda})$ is a subgradient of the concave function $f(\boldsymbol{\Lambda})$ at $\boldsymbol{\Lambda}$ if and only if*

$$f(\tilde{\boldsymbol{\Lambda}}) \leq f(\boldsymbol{\Lambda}) + \nabla_{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda})(\tilde{\boldsymbol{\Lambda}} - \boldsymbol{\Lambda}) \quad (21)$$

Given a subset of components $\boldsymbol{\lambda}$ of $\boldsymbol{\Lambda}$ we denote as $\nabla_{\boldsymbol{\lambda}}(\boldsymbol{\Lambda})$ the corresponding components of $\nabla_{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda})$.

A subgradient of $g(\boldsymbol{\Lambda}, \mathbf{M})$ is presented in the next proposition (see [7]).

Proposition 1 *For given multipliers $\boldsymbol{\Lambda}$ and $\mathbf{M}$, let $\mathbf{X}^\dagger(\boldsymbol{\Lambda}, \mathbf{M})$ denote the optimal argument of the Lagrangian, i.e.,*

$$\mathbf{X}^\dagger(\boldsymbol{\Lambda}, \mathbf{M}) := \arg\min_{\{\mathbf{x}_j \in \mathcal{X}_j\}_{j=1}^J} \mathcal{L}(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{M}) \quad (22)$$

with $\mathcal{L}(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{M})$ given by (18). *Then, a subgradient $\nabla_{\boldsymbol{\Lambda}, \mathbf{M}}$ of $g(\boldsymbol{\Lambda}, \mathbf{M})$ has components,*

$$\nabla_{\boldsymbol{\lambda}_j}(\boldsymbol{\Lambda}, \mathbf{M}) = \mathbf{t}_j^{r\dagger}(\boldsymbol{\Lambda}, \mathbf{M}) - \mathbf{u}_j^\dagger(\boldsymbol{\Lambda}, \mathbf{M})$$
$$\nabla_{\boldsymbol{\mu}_j}(\boldsymbol{\Lambda}, \mathbf{M}) = \mathbf{v}_j^\dagger(\boldsymbol{\Lambda}, \mathbf{M}) - w_j^\dagger(\boldsymbol{\Lambda}, \mathbf{M})\mathbf{1}. \quad (23)$$

Proposition 1 tells us that for general multipliers $(\boldsymbol{\Lambda}, \mathbf{M})$ the values of the primal variables that optimize the corresponding Lagrangian can be used to obtain a subgradient of the dual function. A important property of the optimal arguments of the Lagrangian is that they can be computed locally at each node. To be precise define the vectors $\boldsymbol{\lambda}_j^r = \boldsymbol{\Lambda}_{c(j)j}$ and $\boldsymbol{\mu}_j^r = \mathbf{M}_{jc(j)}$ containing the dual variables of the one hop neighbors $\{U_i : i \in c(j)\}$, and construct the local Lagrangian $\mathcal{L}_j(\mathbf{x}_j; \boldsymbol{\lambda}_j, \boldsymbol{\mu}_j, \boldsymbol{\lambda}_j^r, \boldsymbol{\mu}_j^r)$ by grouping the terms that depend only on the local variable $\mathbf{x}_j$ [cf. (18)]

$$\mathcal{L}_j(\mathbf{x}_j; \boldsymbol{\lambda}_j, \boldsymbol{\mu}_j, \boldsymbol{\lambda}_j^r, \boldsymbol{\mu}_j^r) = -w_j + \mathbf{t}_j^T\boldsymbol{\lambda}_j^r - \mathbf{u}_j^T\boldsymbol{\lambda}_j + w_j\mathbf{1}^T(\boldsymbol{\mu}_j^r - \boldsymbol{\mu}_j). \quad (24)$$

Algorithm 1 Dual decomposition solver

```
Require: Packet success probabilities to and from neighbors $\mathbf{R}_{c(j)j}$ and $\mathbf{R}_{jc(j)}$
Ensure: Optimal multipliers $\boldsymbol{\lambda}_j^*$ and $\boldsymbol{\mu}_j^*$
1: for $n = 1$ to $\infty$ do {repeat for the life of the network}
2:     Receive multipliers $\lambda_{ij}(n)$ and $\mu_{ji}(n)$ from $\{U_i : i \in c(j)\}$
3:     Min. Lagrangian [cf. (26)]: $\mathbf{x}_j(n) = \arg\min_{\mathbf{x}_j \in \mathcal{X}_j} \mathcal{L}_j(\mathbf{x}_j, n)$
4:     Transmit $w_j(n)$, $t_{ij}(n)$, and $u_{ji}(n)$ to $U_i$; for $\{U_i : i \in c(j)\}$.
5:     Receive $w_i(n)$, $t_{ji}(n)$, and $u_{ij}(n)$ from $\{U_i : i \in c(j)\}$
6:     Subg. iter. $\boldsymbol{\lambda}_j$ [cf. (27)]: $\boldsymbol{\lambda}_j(n+1) = \boldsymbol{\lambda}_j(n) + c_n[\mathbf{t}_j^r(n) - \mathbf{u}_j(n)]$
7:     Subg. iter. $\boldsymbol{\mu}_j$ [cf. (27)]: $\boldsymbol{\mu}_j(n+1) = \boldsymbol{\mu}_j(n) + c_n[\mathbf{v}_j(n) - w_j(n)\mathbf{1}]$
8:     Transmit $\lambda_{ji}(n+1)$ and $\mu_{ij}(n+1)$ to $U_i$; for $\{U_i : i \in c(j)\}$
9: end for
```

By construction $\mathcal{L}(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{M}) = \sum_{j=1}^J \mathcal{L}_j(\mathbf{x}_j; \boldsymbol{\lambda}_j, \boldsymbol{\mu}_j, \boldsymbol{\lambda}_j^r, \boldsymbol{\mu}_j^r)$ [cf. (18) and (24)]. If we further note that primal variables $\mathbf{x}_j$ appear only in $\mathcal{L}_j(\mathbf{x}_j; \boldsymbol{\lambda}_j, \boldsymbol{\mu}_j, \boldsymbol{\lambda}_j^r, \boldsymbol{\mu}_j^r)$ we conclude that the optimal arguments in (22) can be found as

$$\mathbf{x}_j^\dagger := \arg\min_{\mathbf{x}_j \in \mathcal{X}_j} \mathcal{L}_j(\mathbf{x}_j, \boldsymbol{\lambda}_j, \boldsymbol{\mu}_j, \boldsymbol{\lambda}_j^r, \boldsymbol{\mu}_j^r). \quad (25)$$

The reasons enabling a distributed implementation of subgradient ascent can be read out from Proposition 1 and (25): i) a subgradient of the dual function is obtained from the arguments optimizing the Lagrangian $\mathcal{L}(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{M})$ [cf. (23)]; ii) the subgradients $\nabla_{\boldsymbol{\lambda}_j}(\boldsymbol{\Lambda}, \mathbf{M})$ and $\nabla_{\boldsymbol{\mu}_j}(\boldsymbol{\Lambda}, \mathbf{M})$ depend only on local and neighboring primal variables [cf. (23)]; and iii) the optimization of the Lagrangian $\mathcal{L}(\mathbf{X}, \boldsymbol{\Lambda}, \mathbf{M})$ separates into the optimization of J local Lagrangians $\mathcal{L}_j(\mathbf{x}_j, \boldsymbol{\lambda}_j, \boldsymbol{\mu}_j, \boldsymbol{\lambda}_j^r, \boldsymbol{\mu}_j^r)$, furthermore, these local Lagrangians depend only on local and neighboring dual variables [cf. (24) and (25)].

Consequently, subgradient ascent for $g(\boldsymbol{\Lambda}, \mathbf{M})$ can be implemented by the following distributable iteration:

[I1] *Compute subgradient*. Given local multipliers $\boldsymbol{\lambda}_j(n)$ and $\boldsymbol{\mu}_j(n)$, and neighboring multipliers $\boldsymbol{\lambda}_j^r(n)$ and $\boldsymbol{\mu}_j^r(n)$, minimize the local Lagrangian with respect to the local primal variables,

$$\mathbf{x}_j(n) = \arg\min_{\mathbf{x}_j \in \mathcal{X}_j} \mathcal{L}_j(\mathbf{x}_j, n) \quad (26)$$

where $\mathcal{L}_j(\mathbf{x}_j, n) := \mathcal{L}_j[\mathbf{x}_j, \boldsymbol{\lambda}_j(n), \boldsymbol{\mu}_j(n), \boldsymbol{\lambda}_j^r(n), \boldsymbol{\mu}_j^r(n)]$.

[I2] *Subgradient ascent step*. Using local $[w_j(n), \mathbf{t}_j(n), \mathbf{u}_j(n)]$ and neighboring $[\mathbf{v}_j(n), \mathbf{t}_j^r(n), \mathbf{u}_j^r(n)]$ primal variables update local multipliers

$$\boldsymbol{\lambda}_j(n+1) = \boldsymbol{\lambda}_j(n) + c_n[\mathbf{t}_j^r(n) - \mathbf{u}_j(n)]$$
$$\boldsymbol{\mu}_j(n+1) = \boldsymbol{\mu}_j(n) + c_n[\mathbf{v}_j(n) - w_j(n)\mathbf{1}] \quad (27)$$

where c_n is a properly selected step size.

Algorithm 1 details the distributed implementation of [I1]-[I2]. Given the local multipliers $\boldsymbol{\lambda}_j(n)$ and $\boldsymbol{\mu}_j(n)$, and the one-hop-neighbors' multipliers $\boldsymbol{\lambda}_j^r(n)$ and $\boldsymbol{\mu}_j^r(n)$, user terminal U_j solves a (local) convex optimization problem to find the primal variables $\mathbf{x}_j(n)$ that optimize the (local and global) Lagrangian; step 3. In turn, these primal variables are used in the gradient ascent steps 6 and 7 to obtain the updated multipliers $\boldsymbol{\lambda}_j(n+1)$ and $\boldsymbol{\mu}_j(n+1)$. Steps 6 and 7 represent the subgradient ascent step for the dual function $g(\boldsymbol{\Lambda}, \mathbf{M})$ and as such are the steps guaranteeing convergence of the iterates $\{\boldsymbol{\lambda}_j(n)\}_{j=1}^J$ and $\{\boldsymbol{\mu}_j(n)\}_{j=1}^J$ obtained from (26)-(27) to $\{\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\}_{j=1}^J := \arg\max g(\boldsymbol{\Lambda}, \mathbf{M})$ as $n \to \infty$ (convergence of (26)-(27) requires some qualifications that we discuss in the next subsection). The remaining steps ensure that the variables are properly communicated. Steps 8 and 2 ensure that the updated multipliers are sent to and received by the corresponding neighboring node, while steps 4 and 5 guarantee the same for the primal variables.

3.1. Convergence issues

Since the iteration (25)-(27) implements subgradient ascent for the dual function, convergence is characterized in terms of the dual variables $\boldsymbol{\lambda}_j$ and $\boldsymbol{\mu}_j$. We now summarize relevant convergence properties of subgradient ascent. Let $\boldsymbol{\lambda}_j^*, \boldsymbol{\mu}_j^*$ denote the optimal solution of the dual problem in (20). We then have that (see e.g., [2]):

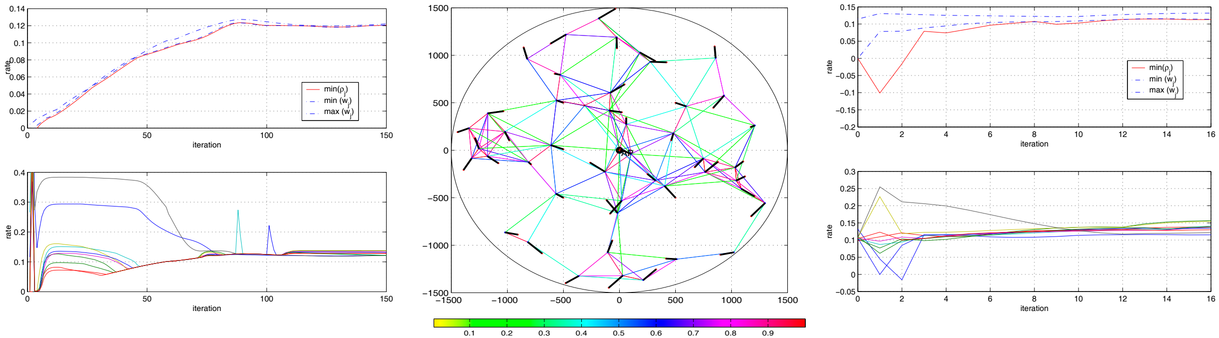

Fig. 3. Convergence of Algorithm 1 to the max-min optimal routes in Fig. 2 (left), and its response to user mobility (middle) and (right). After 70 iterations the rate of the most compromised user is within 90% of the optimal rate. The algorithm is fast to respond to changes in $\mathbf{R}$.

(a) if the step size is constant, i.e., $c_n = c\ \forall n$, then

$$\lim_{n\to\infty} \frac{1}{n}\sum_{n=0}^{\infty} \boldsymbol{\lambda}_j(n) = \boldsymbol{\lambda}_j^* \tag{28}$$

implying that the average value of the dual iterates converges to the optimal dual variables; and

(b) if the step size sequence is non-summable, $\sum_{n=0}^{\infty} c_n = \infty$, but square summable, $\sum_{n=0}^{\infty} c_n^2 < \infty$, then

$$\lim_{n\to\infty} \boldsymbol{\lambda}_j(n) = \boldsymbol{\lambda}_j^* \tag{29}$$

implying that the sequence of dual iterates converges to the optimal dual variables.

Convergence of the dual iterates in the sense described in (a) and (b), does not imply that the same holds true for the primal iterates $\mathbf{x}_j(n)$, and in practice $\lim_{n\to\infty} \mathbf{t}_j(n) \neq \mathbf{t}_j^*$ for many practical optimality criteria. This is particularly true when (12) amounts to an LP, a category that includes the weighted sum-rate and max-min optimality criteria. Many regularization approaches are known to guarantee convergence of the primal iterates $\mathbf{x}_j(n)$; see [7].

4. SIMULATIONS

We consider a network with $J = 40$ nodes randomly placed in a circle of radius 1.5 km at whose center is the common access point U_{J+1}; see Fig. 1. The elements of the packet success probability matrix $\mathbf{R}$ are chosen according to the empirical distribution in [1]. The optimality criterion is max-min rate with corresponding optimal routes given as in Fig. 2. The algorithm ran by individual nodes is the alternating direction method of multipliers, a regularization of Algorithm 1.

The results of running Algorithm 1 are summarized in Figs. 3-(left) where we show the smallest and largest value of the local variables w_j. As expected, these variables quickly approach each other due to the constraints $\mathbf{v}_j = w_j\mathbf{1}$; and the minimum rate $\rho_j := \mathbf{r}_j^T\mathbf{t}_j - \mathbf{s}_j^T\mathbf{t}_j^r$ is also closely approximated by the local variables w_j. The rate of convergence is reasonable, since after $n = 150$ iterations the distributed algorithm has converged to the optimal value. Furthermore, we can see that after 70 iterations the rate of the most compromised user is within 90% of the optimal rate. In practice, this last number can be regarded as the time required for convergence. We also plot in Fig. 3-(left) the path followed by the rate of 10 different representative users with similar conclusions.

Mobility. Among the main motivating reasons behind a distributed implementation is adaptability to a mobile environment. To illustrate this we modify the network in Fig. 1 by letting each node move at random with uniform distribution in a square with 300 meter side centered at the original position. This leads to the network in Fig. 3-(middle). The effect of mobility can be simulated by running Algorithm 1 to find the optimal routes for the network in Fig. 3-(middle) using the optimal routes in Fig. 2 as a initial condition. The results are depicted in Fig. 3-(right). Convergence to the new optimal routes is surprisingly fast taking approximately 8 iterations. Intuitively, this happens because optimal routes are robust with respect to modest topology changes.

5. CONCLUSIONS

Building on recent results that formulate routing problems as convex optimization problems based on the pairwise error probability matrix $\mathbf{R}$, this paper developed distributed routing algorithms to find rate-optimal routes. Since routing algorithms developed in [6] cannot be implemented in a distributed fashion we introduced equivalent problems amenable to distributed implementations. Many problems can be cast in the latter formulation including max-min rate, sum-rate, maximum product-rate, and rate-optimal relay network. Additional convex constraints, e.g., a minimum acceptable rate, can be incorporated in all of these problems. Distributed routing algorithms were obtained via dual decomposition of the original problem leading to an iterative algorithm based on communication with one-hop neighbors only. The algorithms were shown to converge to the optimal routing probabilities. We further presented simulations corroborating that the distributed algorithms rapidly adjust to changes in the pairwise error probability matrix brought in by, e.g., node mobility.

6. 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, Oct. 2004.

[2] D. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2nd ed., 1997.

[3] M.Haenggi and D.Puccinelli, "Routing in ad hoc networks: a case for long hops,"*IEEE Commun. Mag.*, vol.43, pp.93–101, Oct. 2005.

[4] B. Johansson, P. Soldati, and M. Johansson, "Mathematical decomposition techniques for distributed cross-layer optimization of data networks," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 1535–1547, August 2006.

[5] 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.

[6] A. Ribeiro, Z.-Q. Luo, N. D. Sidiropoulos, and G. B. Giannakis, "A general optimization framework for stochastic routing in wireless multi-hop networks," *IEEE Trans. Signal Process.*, August 2006 (submitted). http://www.ece.umn.edu/users/aribeiro/ research/ pubs.html.

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

[8] Q. Zhao and L. Tong, "Energy-efficient adaptive routing for ad hoc networks with time-varying heterogeneous traffic," in *Proc. Int. Conf. Acoustics, Speech, Signal Process.*, vol. 5, pp. 801–804, Philadelphia, PA, March 18-23, 2005.