Spatial Reuse in Dense Wireless Areas: A Cross-layer Optimization Approach via ADMM

Haleh Tabrizi, *Member, IEEE,* Borja Peleato, *Member, IEEE,* Golnaz Farhadi, *Member, IEEE,*
John M. Cioffi, *Fellow, IEEE,* and Ghadah Aldabbagh, *Member, IEEE*

Abstract—This paper introduces an efficient method for communication resource use in dense wireless areas where all nodes must communicate with a common destination node. The proposed method groups nodes based on their distance from the destination and creates a structured multi-hop configuration in which each group can relay its neighbor's data. The large number of active radio nodes and the common direction of communication toward a single destination are exploited to reuse the limited spectrum resources in spatially separated groups. Spectrum allocation constraints among groups are then embedded in a joint routing and resource allocation framework to optimize the route and amount of resources allocated to each node. The solution to this problem uses coordination among the lower-layers of the wireless-network protocol stack to outperform conventional approaches where these layers are decoupled. Furthermore, the structure of this problem is exploited to obtain a semi-distributed optimization algorithm based on the alternating direction method of multipliers (ADMM) where each node can optimize its resources independently based on local channel information.

Index Terms—Alternating direction method of multipliers (ADMM), crosslayer optimization, dynamic resource allocation, routing.

I. INTRODUCTION

With the remarkable growth of wireless technology and advancement of mobile devices, the demand for more efficient use of limited radio resources is increasing rapidly. Optimizing performance through sophisticated physical-layer techniques is no longer enough, so studies have considered coordinating multiple users through scheduling and admission control, as well as joint operation of the physical and network layers in wireless networks. Lin et al. summarize such cross-layer optimization techniques in [1].

This paper studies efficient resource use in dense wireless areas where all nodes communicate with a single destination node. Dense wireless areas are locations populated with many radio devices that must communicate over a shared spectrum. Some examples include concert halls and stadiums populated

H. Tabrizi is with the Department of Electrical Engineering, Stanford University, Stanford, CA, 94305 USA, e-mail: (htabrizi@stanford.edu)
B. Peleato is with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907 USA, e-mail: (bpeleato@purdue.edu).
J. Cioffi is with the Department of Electrical Engineering, Stanford University, Stanford, CA, 94305 USA, and Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia, e-mail: (cioffi@stanford.edu).
G. Farhadi is with Fujitsu Labs of America, Sunnyvale, CA, 94085 USA, e-mail: (gfarhadi@us.fujitsu.com).
G. Aldabbagh is with the Department of Computer Science, KAU, Jeddah, Saudi Arabia, e-mail: (galdabbagh@kau.edu.sa)
This paper has been submitted in part to IEEE global communications conference, Atlanta, GA, December 2013.
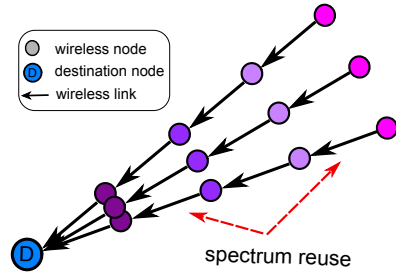

Fig. 1: Multiple hops and reuse between hops.

with user handheld devices, which must all communicate with a common access point or base-station (BS). Home Area Networks (HAN) are another example of dense wireless areas, particularly when such residential local area networks may eventually connect hundreds of digital devices within a home in the so-called "Internet of things".

Figure 1 suggests a configuration that can increase spectrum efficiency by relaying data over multiple hops rather than a single hop (or direct communication). The node configuration motivates a hierarchical multi-hop architecture, where spectrum can be reused among the hops as well as among cells [2, Ch. 15]. Furthermore, the presence of a large number of nodes in a dense wireless area offers many possible routes that the proposed multi-hop configuration can optimize. This approach does not require additional infrastructure, but the nodes act as relays for their neighbor's data.

In this scenario all nodes communicate with a fixed destination node, hence the reuse pattern and flow of data can be structured. The nodes that are in close proximity of each other are grouped together to form an FDMA (frequency division multiple access) system where each node operates on different spectra. Each such group of nodes can relay data for neighboring groups[1] along an optimal routing path determined by the capacity of the links between nodes. The capacity of each link is determined by the power and bandwidth allocated to that link, which in turn depend on the total spectrum allocated to each group. Finding the optimal network configuration therefore requires a joint optimization considering all constraints.

Joint optimization and coordination of the different network layers is not a new concept. It has been used in many different applications in wireless and wireline networks [4],

---

[1]A configurable MAC (e.g. platform proposed in [3]) may be used to enable a different channel access method (random access, FDMA, etc.) for different nodes depending on their congestion level along the path.

[5]. Simultaneous routing and resource allocation methods for orthogonal multiple access systems have been proposed by [6] (FDMA and TDMA) and [7] (CDMA). Both approaches jointly optimize network-layer routing and physical-layer resource allocation through a convex formulation of the problem. A joint scheduling, routing, and resource allocation for multi-hop cellular networks with relay stations is proposed by [8]. In this case, the general problem is non-convex, and hence an iterative algorithm is used to coordinate the layers. Beyond resource allocation and relay selection, [9] jointly optimizes the relay strategy that should be used between each source and destination pair based on a cross-layer design.

This paper focuses on an extended version of the simultaneous routing and resource allocation (SRRA) framework proposed by [6], including additional constraints on the total available spectrum resources and spectrum reuse among the predetermined user groups. The joint optimization of routing and global resource allocation in a dense area is a non-convex problem that can only be solved exactly through exhaustive search methods, such as branch and bound. Unfortunately, these methods are often slow and have exponential worst-case performance [10]. Hence, we propose simplifying the problem into a convex form that can be solved using ADMM (Alternating Direction Method of Multipliers), which has $O(1/k)$ convergence rate [11].

The predetermined user groups are selected based on each node's relative distance to the destination and a subset of all possible links connecting nodes from one group to a neighboring group are selected as candidate data flow links in the optimization problem. The goal is to achieve jointly optimized routing, global spectrum assignment, and local resource allocation for the selected links through convex optimization.

Distributed implementations of network utility maximization problems often lead to simpler subproblems, each optimizing a subset of the decision variables based on local information. Palomar et al. [12] propose a systematic framework to exploit decomposition structures that lead to different distributed algorithms. A very popular approach is using dual decomposition to generate a highly distributed optimization problem [6], but some studies seek faster convergence using Newton's method [13]. Certainly, there is a tradeoff between the amount of message-passing among computing entities, convergence speed, and computational complexity of each distributed algorithm. This paper exploits the structure in the proposed joint optimization framework to generate two different distributed algorithms: The first algorithm (centralized) separates the problem into two subproblems, allowing decoupled optimization of network-layer and physical-layer variables. The second algorithm (semi-distributed) employs all active nodes as computing resources that perform parallel optimization based on local channel information.

The proposed decompositions use the alternating direction method of multipliers (ADMM) [11] algorithm, which is more numerically stable than conventional decomposition methods. For example, ADMM does not require strict convexity of the objective, but the dual decomposition method proposed in [6] does. ADMM has been widely employed in producing distributed algorithms. Some recent examples include multicell coordinated beamforming [14] and distributed model predictive control [15].

The main contributions of this paper are: 1) designing a hierarchical structure of nodes that allows resource reuse among spatially separated nodes, 2) integrating global resource allocation and reuse in a cross-layer optimization framework of routing and resource allocation, and 3) developing a semi-distributed optimization algorithm based on ADMM where each node optimizes the resources based on its local channel information.

The rest of this paper is organized as follows. Section II describes the system model, consisting of the network topology and the physical layer constraints in a dense wireless area. Section III formulates the cross-layer optimization problem for the previous model in a convex form. Section IV proposes a centralized and a distributed algorithm to solve this problem, both based on ADMM. It also provides a numerical example to illustrate the performance and convergence of the algorithm for a small network. Finally, Section V presents more detailed simulations evaluating the performance of the algorithm on larger networks and Section VI concludes and summarizes the paper.

II. PROBLEM FORMULATION

The system model and the proposed hierarchical node configuration dictate a set of system constraints that need to be satisfied while optimizing network performance. This section formulates and motivates these constraints.

A. Network Flow Model

We consider N active users/nodes randomly distributed in a given area that need to communicate simultaneously with a common destination node, such as a cellular base station (BS). This paper investigates uplink transmission, but downlink communication can be formulated in a similar manner with minor modifications. A standard directed graph is used to represent the network topology. The graph is assumed to be connected, *i.e.*, there is a route between every node and the BS. The nodes are labeled as $n \in \mathcal{N} = \{1, 2, ..., N\}$, with node 1 representing the BS. All nodes can transmit, receive and relay data over the existing links. Transmission and reception occur on disjoint spectra. A link exists between two nodes i and j if direct communication among the two is possible. With a total of L links in the network, denote the set of all links by $\mathcal{L} = \{1, 2, ..., L\}$, where each link is identified with an integer value between 1 and L.

The network topology is represented by a link-node incidence matrix $A \in \mathbb{R}^{N \times L}$, whose entries determine a link's source and destination nodes by $+1$ and -1, respectively. For example, for the configuration shown in Figure 2, where $N = 4$ and $L = 3$, the link-incidence matrix is given by: $A = \begin{pmatrix} -1 & 0 & 0 \\ +1 & -1 & -1 \\ 0 & +1 & 0 \\ 0 & 0 & +1 \end{pmatrix}$. Each column of A corresponds to a link and each row corresponds to a node. The first column indicates that link 1 goes from node 2 to node 1. The set of outgoing links from node n are denoted by $\mathcal{O}(n)$ and the set of incoming links are denoted by $\mathcal{I}(n)$.
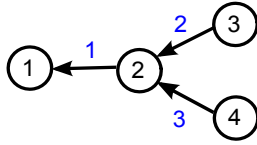
Fig. 2: Simple node and link example.

A multi-commodity flow model [16] that satisfies flow conservation at each node is considered. For the uplink case considered here, $x \in \mathbb{R}^L$ identifies the flow on each link (data rate) that is destined to node 1 (BS). Let A_{nl} denote the element on the n-th row and l-th column of A, and let $r \in \mathbb{R}^N$ be a vector of data traffic injected in the network at each node. Then $Ax = r$, where $r_n = \sum_l A_{nl} x_l$, denotes the amount of data traffic injected into the network at node n. Positive components will indicate flow sources, while negative ones will indicate flow sinks. For the example in Figure 2, the flow conservation law indicates that $r_1 = -x_1$, $r_2 = x_1 - x_2 - x_3$, $r_3 = x_2$, and $r_4 = x_3$. Furthermore, the amount of flow on each link (x_l) is constrained by that link's physical layer capacity c_l:

$$x_l \leq c_l, \quad \forall l \in \mathcal{L}.$$

B. Physical Layer Model

An FDMA channel access method is considered, where each node's outgoing links are assigned disjoint frequency bands. The received signal is corrupted by additive white Gaussian noise with power spectral density σ_l and the link capacity is given by a concave and monotone increasing function of its allocated bandwidth w_l and power p_l, as follows:

$$c_l = w_l \log \left(1 + \frac{p_l q_l}{w_l \sigma_l} \right), \quad \forall l \in \mathcal{L}.$$

Parameter q_l is the channel gain from the source to the destination of link l. The total amount of power allocated to a node n is constrained to be smaller than a pre-fixed value $P_{\max,n}$, hence

$$\sum_{l \in \mathcal{O}(n)} p_l \leq P_{\max,n}, \quad \forall n \in \mathcal{N}.$$

The total amount of bandwidth allocated to node n is denoted by

$$\sum_{l \in \mathcal{O}(n)} w_l = v_n, \quad \forall n \in \mathcal{N}.$$

If necessary, after obtaining the continuous bandwidth variables w_l, they can be quantized based on the underlying system modulation approach. For example, if the underlying system is an OFDMA system, the bandwidth values can be rounded down to an integer number of subcarrier bandwidths and if the underlying system is an LTE system, it can be rounded to the closest integer number of resource blocks.

C. Group Assignment

The objective of the proposed multi-hop configuration is to reuse as much of spectrum as possible. Hence, direct links between far nodes and the BS are divided into multiple

hops that operate on different spectrum bands, allowing reuse (Figure 1). To this end, nodes are grouped based on their distance to the BS as shown in Figure 3: nodes located between radial distances d_{g-1} and d_g from the BS belong in group $\mathcal{G}_g$, and all nodes within distance d_1 of the destination belong to group $\mathcal{G}_1$. The coverage area is thus partitioned into M disjoint sections such that $\mathcal{G}_g \bigcap \mathcal{G}_{g'} = \emptyset$, $\mathcal{G}_g \neq \emptyset$, $\forall g, g'$, and each node belongs to a group: $\bigcup_{g=1}^{M} \mathcal{G}_g = \mathcal{N}$.

In general, the closer a link to the BS, the larger the flow it will be carrying. The links that terminate at the BS carry the largest total flow. The links that are physically farther away from the BS, carry smaller flows (Figure 3). Let f denote the frequency reuse factor, such that every $(f+1)$-th group uses the same spectrum set ($f > 2$ to avoid the hidden node problem). As such, the maximum available spectrum $W_{\max}$ is divided among groups 1 through f:

$$\sum_{g=1}^{f} W_g = W_{\max},$$

where W_g denotes the amount of bandwidth allocated to group g. The rest of the groups ($f+1$ through M) reuse the spectrum allocated to groups 1 through f:

$$W_g = W_{((g-1) \bmod f)+1}, \ \forall g > f.$$

For example, if $f = 3$, the total bandwidth is divided between groups $1, 2$ and 3, while groups $4, 7, 10, \ldots$ use the same spectrum as group 1; groups $5, 8, 11, \ldots$ use the same spectrum as group 2; and groups $6, 9, 12, \ldots$ use the same as group 3.

D. Routing Link Assignments

In order to optimize the routing from each node to the BS, an initial set of possible links should be determined. This subsection describes a method, similar to that in [17], for selecting this initial pool of links (or refining a pre-defined set of links if desired). The method assumes that the BS knows the location of each node and uses relative distances between them to select the initial pool of links. If this assumption does not hold, the initial pool of links could include all those between nodes in consecutive groups. Regardless of how the initial pool of links is selected, resource and flow allocation are based on channel information.

The proposed method will include a link from node i to node j in the initial pool if:

1) $j \in \mathcal{G}_{g-1}$, where $i \in \mathcal{G}_g$.
2) the angular distance between nodes i and j is less than a pre-fixed θ, as shown in Figure 3.
3) the distance between nodes i and j is less than a pre-fixed d_{th}.

The first condition enforces a single hop between consecutive groups to allow spectrum reuse. If $i \in \mathcal{G}_g$ and $j \in \mathcal{G}_h$ with $h < g - 1$, then the link from i to j traverses more than one group (spectrum region) and defeats the purpose of reuse. On the other hand, if $h = g$ then i and j belong to the same group creating an extra hop within the group, and if $h > g$, then j is farther away from the BS than i already is. In both cases, j has no relaying benefit for i.
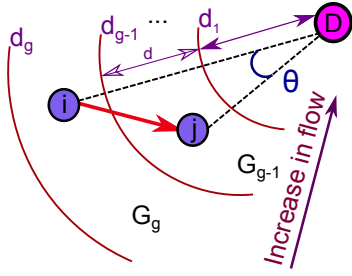
Fig. 3: Link assignment conditions.

The second and third conditions ensure that the relay node j is not too far away from the source node i. Parameters d_{th} and θ control the number of outgoing links from a node and allow selecting a reasonable (non-empty) set of possible routes.

The maximum transmission power per unit bandwidth per link is limited to avoid interference between groups operating on the same spectrum. This maximum transmission power depends on the radial distances d_g, $g = \{1, 2, ..., M\}$ defining the groups and on the frequency reuse factor f. If every group has the same radial distance d, the minimum possible distance between the transmitter of one link and a receiver suffering its interference is equal to $d_{int} = (f - 2) \times d$. The received power per unit bandwidth has to be less than a fraction α of the noise power for interference to be negligible, hence the constraint on the transmission power is:

$$p_l \leq w_l \frac{\alpha N_0}{K(\frac{l_0}{d_{int}})^a},$$

where a is the pathloss exponent, l_0 is the reference distance[2], N_0 is the noise power spectral density at the receiver (σ_l), and K is the attenuation factor based on the simplified pathloss model [2]. This work denotes the bandwidth coefficient by

$$\gamma = \alpha N_0 \left(K \left(\frac{l_0}{d_{int}} \right)^a \right)^{-1},$$

such that the per link transmission power constraint becomes $p_l \leq w_l \gamma$.

III. Joint Routing, Global and Local Resource Allocation

Based on the network conditions, such as node density and locations, the user groups and possible routing links are determined. Given these, the routing and resource allocation problem can be formulated as a convex optimization problem with the objective of maximizing a concave utility function. To maintain fairness among all users, the utility function was chosen to be the minimum rate among all users ($n \in \mathcal{N} \setminus \{1\}$). Only uplink communication is considered here, so the set of all transmitters (all nodes other than the BS) is $\tilde{\mathcal{N}} = \mathcal{N} \setminus \{1\}$.

For each link $l \in \{1, \ldots, L\}$ in the initial pool, we define the system variables p_l and w_l, respectively denoting the amount of power and spectrum allocated to the link, and x_l, denoting

---

[2]This is a conservative bound, using the shortest possible distance between interfering nodes. Results could be improved using the actual distance for each node, if static and known.

the amount of flow on the link destined to the BS. W_g, $\forall g \in \{1, ..., M\}$ will denote the amount of spectrum allocated to each group. The problem then becomes:

$$\begin{aligned}
\underset{p, w, x, (r_n, W_g)}{\text{maximize}} \quad & \min_{n \in \tilde{\mathcal{N}}} r_n && (1)
\end{aligned}$$

$$\text{subject to} \quad \sum_l A_{nl} x_l = r_n, \ \forall n \in \mathcal{N} \qquad (1a)$$

$$x_l \leq w_l \log \left(1 + \frac{p_l q_l}{w_l N_0} \right), \ \forall l \in \mathcal{L} \qquad (1b)$$

$$p_l \leq w_l \gamma, \ \forall l \in \mathcal{L} \qquad (1c)$$

$$\sum_{l \in \mathcal{O}(n)} p_l \leq P_{\max}, \ \forall n \in \tilde{\mathcal{N}} \qquad (1d)$$

$$\sum_{n \in \mathcal{G}_g} \sum_{l \in \mathcal{O}(n)} w_l \leq W_g, \ \forall g \in \{1, 2, ..., M\} \qquad (1e)$$

$$\sum_{g=1}^{f} W_g = W_{\max} \qquad (1f)$$

$$W_g = W_{((g-1) \bmod f)+1}, \ \forall g > f \qquad (1g)$$

$$x_l \geq 0, \ p_l \geq 0, \ w_l \geq 0, \ \forall l \in \mathcal{L}. \qquad (1h)$$

Constraints (1a) through (1c) are as explained in Section II. Constraint (1d) is obtained by setting the maximum transmission power of all users equal to $P_{\max}$. Constraints (1e) through (1g) identify the spectrum reuse constraints introduced in Section II. The variables W_g and r_n are auxiliary variables introduced to make the problem formulation easier to understand. The value for r_n can be derived from x using equations (1a). Similarly, it is easy to prove that constraint (1e) will be tight in the optimal solution, so the value for W_g can also be uniquely derived from w. Consequently, both r_n and W_g appear between brackets in the list of optimization variables. This notation for auxiliary variables continues throughout the rest of the paper.

The constraints in (1) define a convex set and the objective function is concave. Hence, the problem is convex and can be solved globally and efficiently by interior-point methods [18]. However, there might be multiple solutions with the same objective value. Adding a small negative factor of power sum, $-\epsilon \sum p_l$, to the objective would guarantee that a solution with small power values is selected among the optimal set. This penalty term has negligible effect on the resulting minimum rate value r_n.

The conventional direct-mode resource-allocation problem can be derived as a specific scenario of the multi-hop optimization problem (1). In this formulation, there is only one group of nodes, and the only possible route from each node to the destination node is a direct link between the node and the destination. In this case, the problem can be rewritten as:

$$\begin{aligned}
\underset{p, v, (r_n)}{\text{maximize}} \quad & \min_{n \in \tilde{\mathcal{N}}} r_n && (2)
\end{aligned}$$

$$\text{subject to} \quad 0 \leq r_n \leq v_n \log \left(1 + \frac{p_n q_n}{v_n N_0} \right), \ \forall n \in \tilde{\mathcal{N}}$$

$$0 \leq p_n \leq P_{\max}, \ \forall n \in \tilde{\mathcal{N}}$$

$$\sum_{n \in \tilde{\mathcal{N}}} v_n = W_{\max},$$

$$v_n \geq 0, \ \forall n \in \tilde{\mathcal{N}},$$

where v_n and p_n are equivalent to the original parameters w_l and p_l, respectively, since there is only one outgoing link from each node.

IV. DISTRIBUTED OPTIMIZATION

Dual decomposition is an old method for solving a convex optimization problem in a (semi-) decentralized manner[3] [19]. Generally, if the objective function is separable in its variables, the problem can be split into smaller subproblems. These subproblems can then be solved iteratively in parallel based on dual ascent (or descent). However, this method requires some hard assumptions such as strict convexity and finiteness of the objective function; otherwise it can result in numerical instability.

Augmented Lagrangian methods such as the method of multipliers achieve better convergence results by adding a penalty parameter to the dual ascent objective; however, the penalty parameter is not separable when the original problem is. The alternating direction method of multipliers (ADMM), on the other hand, is an algorithm that combines the *decomposability* property of dual ascent with the robust convergence properties of the method of multipliers [11].

The objective function in problem (1) is not strictly convex so dual decomposition cannot be applied directly. A weighted quadratic regularization term would make the objective strictly convex, but finding the appropriate weights is a challenging problem. If the weights are too small, dual decomposition will still suffer from numerical stability, while excessive weights would involve an optimality cost. ADMM is in general more numerically stable and faster in convergence than the conventional dual decomposition method [14]. This paper thus proposes ADMM to implement distributed cross-layer optimization algorithms for spatial reuse.

A. Disjoint Network-Layer and Physical-Layer Optimization

One common method for developing efficient algorithms in cross-layer optimization problems is to decouple the network-layer and physical-layer problems [6], [12]. This method divides the problem into two simpler subproblems, the optimization of network-layer and physical-layer variables, which can be solved separately and updated iteratively by exchanging messages between the two layers. This section takes a similar approach in decoupling the problems through ADMM. At each iteration $k \in \mathbb{Z}$, the network layer and physical layer subproblems are solved in parallel and based on the results, the dual variables are updated.

The network-layer parameters x and the physical-layer parameters w and p are coupled by constraint (1b) in the above centralized optimization problem. To decompose the problem, a new auxiliary variable $t \in \mathbb{R}^L$ and L equality

constraints $t_l = x_l, \ \forall l$ are introduced. ADMM decouples constraints $t_l = x_l, \forall l$ through a dual variable $u \in \mathbb{R}^L$, and decomposes problem (1) into two subproblems corresponding to the network and physical layers.

The augmented Lagrangian for the extended problem with penalty parameter $\rho > 0$ and dual variable y is

$$\mathfrak{L} = \min_{n \in \tilde{\mathcal{N}}} (r_n) - y^T(x - t) - \frac{\rho}{2} \|x - t\|_2^2,$$

subject to constraints (1a)-(1h), but it is often more convenient to scale the dual variable y and combine the linear and quadratic terms, applying ADMM to

$$\tilde{\mathfrak{L}} = \min_{n \in \tilde{\mathcal{N}}} (r_n) - \frac{\rho}{2} \|x - t + u\|_2^2,$$

where $u = \rho^{-1}y$ and still subject to constraints (1a)-(1h). Both formulations are equivalent, as shown in [11] (section 3.1.1). The network-layer optimization problem is then:

$$\begin{aligned} &\underset{x \geq 0, (r_n)}{\text{maximize}} && \min_{n \in \tilde{\mathcal{N}}} (r_n) - \frac{\rho}{2} \|x - t + u\|_2^2 && (3) \\ &\text{subject to} && \sum_l A_{nl} x_l = r_n, \ \forall n \in \mathcal{N}, \end{aligned}$$

and the physical layer problem is:

$$\begin{aligned} &\underset{p, w, t \geq 0, (W_g)}{\text{maximize}} && -\frac{\rho}{2} \|x - t + u\|_2^2 && (4) \\ &\text{subject to} && t_l \leq w_l \log\left(1 + \frac{p_l q_l}{w_l N_0}\right), \ \forall l \in \mathcal{L} \\ & && (1c) - (1g). \end{aligned}$$

Let superscripts denote iteration indices, omitted in the above subproblems for clarity. At iteration $k = 1$, the network layer parameters $x^{(1)}$ are obtained from (3) using arbitrary initial values for the dual variables $u^{(0)}$ and auxiliary variables $t^{(0)}$. The physical layer parameters $p^{(1)}$, $w^{(1)}$ and $t^{(1)}$ are then obtained from (4) using the same $u^{(0)}$ and the obtained values for $x^{(1)}$. Finally, the dual variable $u^{(1)}$ is updated based on $u^{(0)}$, $t^{(1)}$, and $x^{(1)}$ as follows:

$$u^{(k+1)} = u^{(k)} + x^{(k+1)} - t^{(k+1)}. \tag{5}$$

ADMM has guaranteed convergence for any $\rho > 0$ but in practice the speed of convergence can change depending on its value. Unfortunately, there is no way to know in advance which value will yield the fastest convergence. Convergence results of this algorithm are investigated in Section IV-D through an example.

B. Semi-Distributed Device Optimization Algorithm

The authors are not aware of any fully distributed algorithm that guarantees convergence to the global optimum using only local coordination among nodes. In any case, such an algorithm would probably require a very large number of iterations, since the constraints in problem (1) are not local. Information such as data rate from each node needs to reach every potential relay down the chain, then travel back and forth until they converge on a solution. Instead, a *semi*-distributed algorithm is proposed where each node independently optimizes its own resources based on local channel information and limited

---

[3]Contrary to popular belief, dual decomposition does not provide a fully distributed solution. There still needs to be a central entity gathering information from all nodes.

communication with a central entity (such as the BS). The algorithm uses ADMM to decompose problem (1) into an independent subproblem for each node and a shared dual update. One major benefit of the semi-distributed algorithm is that the algorithm can adapt to changing local channel conditions during the iterations of solving the global problem. In the centralized approach, however, the final system parameters are obtained based on the initial channel information sent to the central unit[4].

In the semi-distributed problem there are two sets of coupling variables: the per-link flows x_l and the per-node bandwidths v_n. To decouple the per-node optimization subproblems from the global problem, auxiliary sets of variables t_l, $\forall l \in \mathcal{L}$ and b_n, $\forall n \in \tilde{\mathcal{N}}$ are introduced along with the following equality constraints:

$$t_l = x_l, \ \forall l \in \mathcal{L} \tag{6a}$$
$$b_n = v_n, \ \forall n \in \tilde{\mathcal{N}}. \tag{6b}$$

Even though it appears that the problem complexity increases with the increased number of constraints and variables, it will be shown that such decoupling generates simpler convex subproblems. The set of original variables, $\psi \in \mathbb{R}^{L+N-1}$, and the set of new variables, $z \in \mathbb{R}^{L+N-1}$, are defined as: $\psi = [x_1, ..., x_L, v_2, ..., v_N]$, $z = [t_1, ..., t_L, b_2, ..., b_N]$. The two variables v_1 and b_1, which correspond to the destination node bandwidth, are eliminated.

The variable $\xi \in \mathbb{R}^{L+N-1}$ represents the set of dual variables $\xi = (u, y)$, such that $u \in \mathbb{R}^L$ corresponds to the routing variables and $y \in \mathbb{R}^{N-1}$ corresponds to the bandwidth variables. The Lagrangian then becomes

$$\tilde{\mathfrak{L}} = \min_{n \in \tilde{\mathcal{N}}} (r_n) - \frac{\rho}{2}\|\psi - z + \xi\|_2^2,$$

subject to constraints (1a)-(1h). With this notation and using ADMM, problem (1) is decomposed into two subproblems. The first subproblem optimizes routing and the amount of bandwidth allocated to each user:

$$\begin{aligned} \underset{\psi=[x,v]\geq 0, (r_n, W_g)}{\text{maximize}} \quad & \min_{n \in \tilde{\mathcal{N}}} (r_n) - \frac{\rho}{2}\|\psi - z + \xi\|_2^2 \quad (7) \\ \text{subject to} \quad & \sum_l A_{nl} x_l = r_n, \ \forall n \in \mathcal{N} \\ & \sum_{n \in \mathcal{G}_g} v_n \leq W_g, \ g \in \{1, 2, ..., M\} \\ & (1f) - (1g). \end{aligned}$$

The second subproblem determines variables z, per link power p_l, and bandwidth w_l, $\forall l$:

$$\begin{aligned} \underset{p,w,z=[t,b]\geq 0}{\text{maximize}} \quad & -\frac{\rho}{2}\|\psi - z + \xi\|_2^2 \quad (8) \\ \text{subject to} \quad & t_l \leq w_l \log\left(1 + \frac{p_l q_l}{w_l N_0}\right), \ \forall l \in \mathcal{L} \\ & \sum_{l \in \mathcal{O}(n)} w_l = b_n, \ \forall n \in \tilde{\mathcal{N}} \end{aligned}$$

[4]The addition energy consumption for message-passing in the semi-distributed approach and the performance gain that can be obtained by updating local channel information are very system dependent and are out of the scope of this work.

$$(1c) - (1d).$$

Subproblems (7) and (8) are then solved iteratively with dual updates:

$$\xi^{(k+1)} = \xi^{(k)} + \psi^{(k+1)} - z^{(k+1)}. \tag{9}$$

As demanded, problem (8) can now be decomposed into $N - 1$ separate subproblems, such that node n solves the following problem based on its local channel information $q_l, \forall l \in \mathcal{O}(n)$:

$$\begin{aligned} \underset{p_l, w_l, t_l, \forall l \in \mathcal{O}(n), b_n}{\text{maximize}} \quad & -\frac{\rho}{2} \sum_{l \in \mathcal{O}(n)} \|x_l - t_l + u_l\|_2^2 \quad (10) \\ & -\frac{\rho}{2}\|v_n - b_n + y_n\|_2^2 \\ \text{subject to} \quad & t_l \leq w_l \log\left(1 + \frac{p_l q_l}{w_l N_0}\right), \ \forall l \in \mathcal{O}(n) \\ & \tag{10a} \\ & \sum_{l \in \mathcal{O}(n)} p_l \leq P_{\max} \quad (10b) \\ & p_l \leq w_l \gamma, \ \forall l \in \mathcal{O}(n) \\ & \sum_{l \in \mathcal{O}(n)} w_l = b_n \\ & p_l, w_l, t_l \geq 0, \ \forall l \in \mathcal{O}(n). \end{aligned}$$

Furthermore, subproblem (7) is separable into bandwidth v and flow variables x. The objective in (7) can be rewritten as $\min_{n \in \tilde{\mathcal{N}}} (r_n) - \frac{\rho}{2}\|x - t + u\|_2^2 - \frac{\rho}{2}\|v - b + y\|_2^2$. Hence this subproblem can further be decomposed into routing and spectrum assignment problems. The routing subproblem is:

$$\begin{aligned} \underset{x\geq 0, (r_n)}{\text{maximize}} \quad & \min_{n \in \tilde{\mathcal{N}}} (r_n) - \frac{\rho}{2}\|x - t + u\|_2^2 \quad (11) \\ \text{subject to} \quad & \sum_l A_{nl} x_l = r_n, \ \forall n \in \mathcal{N}. \end{aligned}$$

The spectrum assignment subproblem is simply the Euclidean projection onto a constrained set:

$$\begin{aligned} \underset{v\geq 0, (W_g)}{\text{maximize}} \quad & -\frac{\rho}{2}\|v - b + y\|_2^2 \quad (12) \\ \text{subject to} \quad & \sum_{n \in \mathcal{G}_g} v_n \leq W_g, \ g \in \{1, 2, ..., M\} \\ & (1f) - (1g). \end{aligned}$$

Hence problems (11) and (12) can be solved in parallel and even by two independent entities. The dual variables u and y can also be updated separately as:

$$\begin{aligned} u^{(k+1)} &= u^{(k)} + t^{(k+1)} - x^{(k+1)} \quad (13) \\ y^{(k+1)} &= y^{(k)} + b^{(k+1)} - v^{(k+1)}. \quad (14) \end{aligned}$$

The primal residuals corresponding to link flows are $h_1^{(k)} = x^{(k)} - t^{(k)}$ and the primal residuals corresponding to bandwidth are $h_2^{(k)} = v^{(k)} - b^{(k)}$. The dual residuals are then defined as $s_1^{(k)} = -\rho(t^{(k)} - t^{(k-1)})$ and $s_2^{(k)} = -\rho(b^{(k)} - b^{(k-1)})$.

Various criteria can be considered as stopping points for the ADMM method. According to [11], one possibility could be:

$$\|h_1^{(k)}\|_2 \leq \epsilon_1^p, \ \|h_2^{(k)}\|_2 \leq \epsilon_2^p, \quad (15)$$
$$\|s_1^{(k)}\|_2 \leq \epsilon_1^d, \ \|s_2^{(k)}\|_2 \leq \epsilon_2^d$$

for small predefined ϵ_1^p, ϵ_2^p, ϵ_1^d, and ϵ_2^d values.

C. Implementation

This section focuses on the implementation of the semi-distributed device optimization algorithm introduced above and investigates the required message passing between different entities. All the information regarding channels, network topology, and other such infrastructure can be communicated to the nodes by the central unit (base-station) through control channels. In cellular networks, signaling for link establishment can be done via slight modifications to the standard radio resource control (RRC) connection reconfiguration procedure [20]. This procedure can include signaling to the sink nodes of each hop indicating their role as well as providing a list of source nodes that the sink node is providing wireless broadband service to.

Based on the optimization subproblems developed in the previous section, the entity solving problem (11) (which can be the BS) provides each user n with the flows x_l, $\forall l \in \mathcal{O}(n)$ that it should carry on each of its outgoing links and the total amount of bandwidth v_n that it can use. Given v_n and x_l, each node calculates p_l, w_l, and t_l, $\forall l \in \mathcal{O}(n)$ by solving optimization problem (10). Each node n then broadcasts t_l, $\forall l \in \mathcal{O}(n)$ and $b_n = \sum_{l \in \mathcal{O}(n)} w_l$.

The primal variables x and v can be calculated at a single central entity in parallel or by two independent entities. Denote by CU 1, the central unit that collects t_l values from all nodes and solves problem (11) to obtain x. A second central unit (CU 2) is in charge of group bandwidths: it gathers per user calculated bandwidths b_n and solves (12) to obtain v. After each iteration, the dual variables u and y are updated. The dual variable updates can be performed either at the central units or at each node, resulting in different message passing structures. Figure 4 shows the message-passing when the updates are performed at the CU's, and the corresponding distributed algorithm is summarized in Algorithm 1.

Figure 4 shows the variables that are updated by each entity and the reference number of the problem solved to obtain the corresponding variable. In this configuration, based on initialized values of $u^{(0)}$ and $t^{(0)}$, CU 1 solves optimization problem (11) to obtain $x^{(1)}$. Then the sum of the dual variable and the updated primal variable $(u^{(0)}+x^{(1)})$ is broadcast. Similarly, based on initialized values of $y^{(0)}$ and $b^{(0)}$, CU 2 solves optimization problem (12) to obtain $v^{(1)}$. Then $(y^{(0)} + v^{(1)})$ is broadcast. Each node n captures its required information $(y_n^{(1)} + v_n^{(1)})$ and $(u_l^{(0)} + x_l^{(1)})$, $\forall l \in \mathcal{O}(n)$, calculates $b_n^{(1)}$ and $t_l^{(1)}$, $\forall l \in \mathcal{O}(n)$ by solving problem (10), and broadcasts the updated variables. During the next iteration, CU 1 calculates $u^{(1)}$, and CU 2 calculates $y^{(1)}$ based on equations (13) and (14), respectively. In this manner, at each iteration, all the dual and primal parameters are updated.

Each update step in Algorithm 1 is followed by broadcasting the updated parameters. The dual variable update is performed at the CU's, because there are generally more computing resources at these locations as compared to individual nodes. According to problem (10), the only information that each node requires from the CU's is the sum of x_l and u_l for all its outgoing links and the sum of v_n and y_n. Hence in order to reduce the amount of message-passing between the nodes and
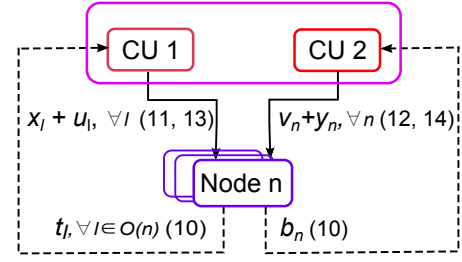

Fig. 4: 3-level optimization and message passing among different entities.

Algorithm 1 Proposed distributed cross-layer optimization.

    initialize u_l, y_n, t_l, b_n, $\forall l$ and $\forall n$

1: **repeat**
2:     perform in parallel:
        • CU 1: given u_l and t_l, update x_l (11) and broadcast $x_l + u_l, \forall l$.
        • CU 2: given y_n and b_n, update v_n (12) and broadcast $v_n + y_n, \forall n$.
3:     Each node n: given $x_l + u_l$, $\forall l \in \mathcal{O}(n)$, $v_n + y_n$ and local channel information, update $t_l, \forall l \in \mathcal{O}(n)$ and b_n (10) and broadcast results.
4:     CU 1 and 2: update corresponding dual variables $u_l, \forall l$ and $y_n, \forall n$ (9).
5: **until** stopping criterion (15) met.

CU's, the sum of the dual and primal variables $x_l + u_l$, $\forall l$ and $w_n + y_n$, $\forall n$ are broadcast by the CU's instead of the separate variable values (Figure 4). After convergence, the optimum routing variables x and physical-layer resources p and w are obtained.

In summary, the semi-distributed optimization algorithm works as follows: Each node n gathers and stores the values of $x_l + u_l$, $\forall l \in \mathcal{O}(n)$, $v_n + y_n$. Then it performs an Euclidean projection onto a set of constraints (10), which has very small complexity, obtaining t_l, $\forall l \in \mathcal{O}(n)$, and b_n. Each node then broadcasts its updated results. Two central units, CU 1 and CU 2, that can be the same or independent entities, gather the updated values of t_l and b_n, respectively. CU 1 and CU 2 then solve (11) and (12), respectively, in parallel. These problems also have very small complexity. The results of such optimizations are x_l, $\forall l \in \mathcal{L}$, and v_n, $\forall n \in \tilde{\mathcal{N}}$. CU 1 and CU 2 then update the dual variables u and y, respectively, while storing a local copy of each variable for the next dual variable update.

In theory, the proof of convergence for ADMM requires that all nodes update t_l, $\forall l \in \mathcal{O}(n)$ and b_n in step-3 before the algorithm can proceed to step-4. In most practical cases, however, the algorithm still converges even if some nodes occasionally fail to perform the update. A good practical approach is to allow partial updates (only a subset of nodes update their local variables) during the first few iterations, and impose that all nodes perform the update in subsequent ones. This approach guarantees convergence to the same solution, while allowing faster execution of the algorithm in cases where
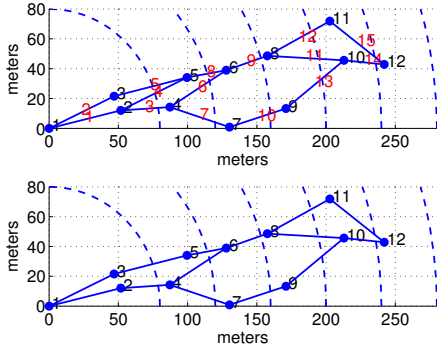
Fig. 5: Top: simple node configuration (N = 12), bottom: resulting optimum routing configuration.
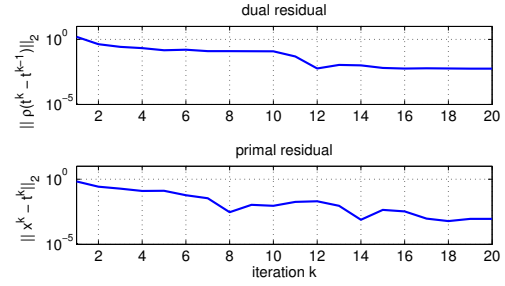

Fig. 6: Norms of primal and dual residuals versus iteration for layered optimization problem.

Fig. 7: Max-min rate convergence for layered optimization problem.

some nodes are significantly slower than others. Wei and Ozdaglar proposed in [21] an asynchronous distributed version of ADMM that converges in $O(1/k)$, but it assumes that all entities send updates infinitely often and only guarantees convergence with probability 1, unlike regular ADMM. The number of iterations required with this asynchronous version can also be significantly larger than with regular ADMM.

The complexity of Algorithm 1 scales gracefully to large number of nodes N. Step-2 consists of solving a simple quadratic program (11) and performing an Euclidean projection onto the convex set specified in (12). Both of them can be solved using interior point methods with $O(N)$ flops per iteration. Step-3 is performed in parallel by the nodes, so its complexity does not grow. An interior point method at node n would use $O(|\mathcal{O}(n)|)$ flops per iteration. Finally, Step-4 computes a linear combination of optimized variables, which again is a very simple operation with complexity $O(N)$. The amount of data broadcast by the nodes also increases linearly with the number of nodes and links. In low-power large scale networks such as wireless sensor networks (WSN), the message-passing overhead can be comparable to the actual data size. The proposed algorithm can then demolish the low-power objective of WSNs; however, it can prove beneficial in dense areas populated with devices that have higher energy resources.

D. A Numerical Example

Figure 5 illustrates the algorithm operation over a circular cellular area of radius 280m. A total of 11 nodes (with destination $N = 12$) are randomly distributed in this area, creating 6 groups separated by $d = 40$m (except $d_1 = 2d$). We assume that a total bandwidth of $W_{\max} = 10$MHz is available with a carrier frequency of 800Mhz. The receiver noise power spectral density is $N_0 = 10^{-11}$W/MHz. The reference distance l_0 in constraint (1c) is set to 1 and the pathloss exponent is $a = 4$. Each user's maximum transmission power is set to $P_{\max} = 0.5$W.

Setting the angular threshold $\theta = 10^o$ and the distance threshold $d_{th} = 1.5d$ in the link assignment step, a total of $L = 15$ initial links are generated, as shown in the top plot of Figure 5. CVX, a package for specifying and solving convex programs [22], is used to solve the centralized optimization
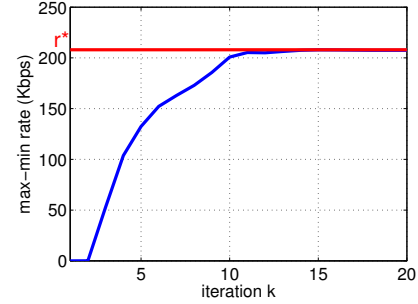
problem (1), resulting in a max-min rate of 208Kbps. For the disjoint network layer and physical layer optimization problems, the penalty parameter is set to $\rho = 1$ (for variations refer to [11, Ch 3.4]) and routing parameters $t^{(0)}$ are initialized to 1. Solving the individual subproblems (3) and (4) iteratively via ADMM results in the same max-min rate of 208Kbps, as shown in Figure 7. The norms of the primal and dual residuals at each iteration are depicted in Figure 6. The results show that the centralized algorithm converges in about 15 iterations.

Before implementing the subproblems (10), (11), and (12) of the semi-distributed algorithm in Section IV-B, it is important to investigate their nature. Optimization problem (10) is performed by all the nodes (in parallel) and has a quadratic objective and linear constraints other than constraint (10a). By approximating the log function with a piecewise linear function, problem (10) can be converted to a simple Quadratic Programming (QP) problem with a relatively small number of variables. Then CVXGEN [23], which can generate fast custom code for small QP-representable convex optimization problems, can be used to solve the problem.

A sequential convex programming approach [24] is used to convert subproblem (10) into a QP: the capacity term in (10a), $c(w_l, p_l) = w_l \log(1 + \frac{p_l q_l}{w_l N_0})$, is approximated by a piece-wise linear function: $\tilde{c}(\beta) = \min_{(a_1,a_2) \in \Sigma} \left\{ a_1^T \beta + a_2 \right\}$, where $\beta = (p_l, w_l)$ and Σ represents a set of planes tangent to $c(\beta)$. For simplicity, Σ in this case consists of only 10 planes. Consequently, constraint (10a) is replaced with the following linear constraint:

$$t_l \le \min_{(a_1,a_2) \in \Sigma} \left\{ a_1^T \beta + a_2 \right\}, \ \forall l \in \mathcal{O}(n)$$

This approximation can be repeatedly re-adjusted during the execution of the algorithm to choose planes that are tangent
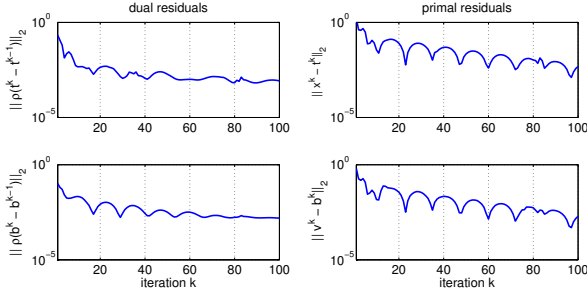
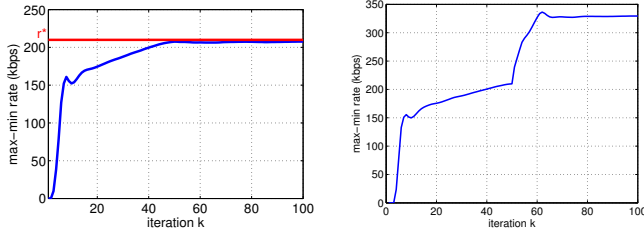Fig. 8: Norms of primal and dual residuals versus iteration for the semi-distributed optimization problem.

Fig. 9: Max-min rate convergence for the semi-distributed optimization problem for a time-invariant channel (left) and N_0 changed after 50 iterations (right).

at the current iterate, and hence the approximation error is progressively reduced. Eventually, when a point is reached where planes are tangent at the iterate points and the iterates remain constant (reached an equilibrium point), it is ensured that the optimal solution with zero approximation error is achieved.

The number of variables in this subproblem depends on the number of outgoing links from each node. Each node's unknown parameters are t_l, p_l and w_l, and hence if node n has L_n outgoing links, it will have $3L_n$ variables. Since the nodes in Figure 5 only have 1 or 2 outgoing links, each node has a 3 or 6 variable QP problem to solve.

Subproblem (12) is readily in QP format and subproblem (11) can be easily converted to a QP by replacing it with its epigraph form. The problem then becomes

$$\begin{array}{ll} \underset{x \geq 0, (\nu)}{\text{maximize}} & \nu - \dfrac{\rho}{2}\|x - t + u\|_2^2 \qquad (11) \\ \text{subject to} & \displaystyle\sum_l A_{nl} x_l = \nu, \ \forall n \in \tilde{\mathcal{N}}, \end{array}$$

Both subproblems (11) and (12) with $L = 15$ and $N = 12$ variables, respectively, can now be implemented using CVX-GEN, as well.

Setting $\rho = 0.5$ and initializing the flows $t^{(0)}$ and band-widths $w^{(0)}$ to 1, a max-min rate of 208Kbps is obtained through the above approach. The primal and dual residual norms at each iteration are represented in Figure 8, and Figure 9 (left) represents the max-min rate convergence results. Simulations using CVXGEN (on an Intel core i7, 2.7 GHz processor) took an average of 0.327ms to solve problem (11) and 0.132ms to solve problem (12). Problem (10), solved in parallel by all nodes, took 0.145ms in average. In this scenario, the semi-distributed algorithm requires approximately 50 iter-

ations to converge. Ignoring the time required for message passing $50 \times (0.145 + \max\{0.132, 0.327\}) = 23.6$ms are required on average to obtain the optimal routing and resource allocation solution. Ignoring the links that have negligible flows (e.g. less than 10Kbps in this case), the bottom plot of Figure 5 is obtained, where Link 4 has been eliminated due to carrying only 8Kbps. If desired, the resource allocation can then be re-optimized based on the updated links.

The number of iterations and time required for the algorithm to converge can be reduced significantly through "warm start" initialization techniques [11, ch.4.3]. The rapid convergence of ADMM when the initial point is close to the optimal solution also allows the semi-distributed implementation to adapt to changing network conditions. Figure 9 (right) shows the convergence of the max-min rate when each node's receiver noise power spectral density N_0 is scaled by a random factor between $\frac{1}{2}$ and $\frac{5}{2}$ after the first 50 iterations. The nodes do not need to notify the CUs of this change, they just use the new value when solving their subproblems. ADMM took 50 iterations to converge from the initial conditions, but only 12 more iterations to adapt to the new values for N_0.

V. PERFORMANCE EVALUATION

This section evaluates the performance of the proposed hierarchical node configuration and cross-layer optimization algorithms by using the direct communication in (2) as the base line for comparison. Simulations consider a single sector of a cellular network. However, the results can be easily extended to the entire cellular area, allowing spectrum to be reused among different sectors as well as among the groups within a sector.

A. Simulation Setup

An instance of the simulation scenario is shown in Figure 10. It consists of a single sector from a 6-sector circular cellular network with radius 210 meters. A total of 44 users ($N = 45$ with destination node) are randomly distributed within this sector and the common destination node is located at $(0, 0)$. The sector is sub-divided such that the radial distance between consecutive groups is $d = 30$m. Since all nodes within group 1 communicate directly with the node at $(0, 0)$, the radial distance to the boundary of group 1 (d_1) is set to double the regular distance d. Such value keeps the average link lengths within group 1 similar to other links in the network. According to the routing-link assignment policy, with a distance threshold of $1.5d$ and angular threshold of 15^0, a total of 86 links were generated for the network in Figure 10.

The number of hops can be adjusted by varying d. For low rate requirements, as the number of hops increases, the total required transmission power from a source to destination decreases. However, there is a limit on the number of hops, since there must be at least one node per group to take advantage of the frequency reuse. Furthermore, the reduction in transmission power with increasing number of hops comes at the expense of increasing the number of nodes cooperating to transmit data and a larger routing table.
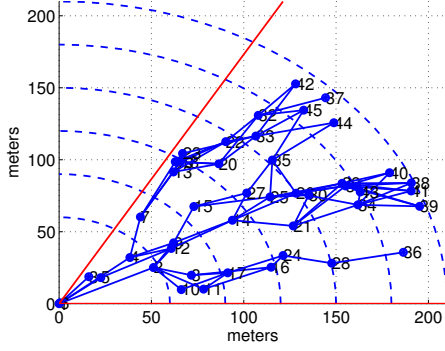
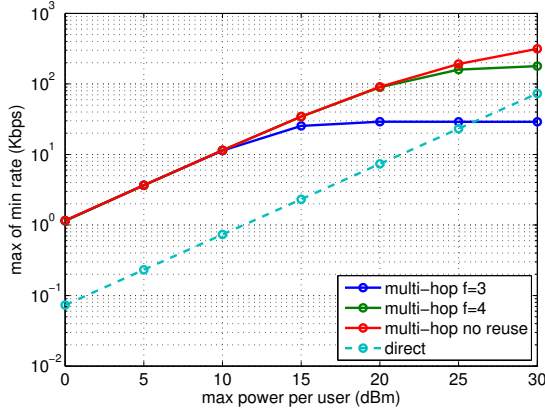Fig. 10: Nodes distributed randomly ($N = 45$) with initial set of possible links ($L = 86$).

Fig. 11: Per user rate vs. per user maximum transmission power.

The total available bandwidth was assumed to be $W_{max} =$ 10MHz with carrier frequency of 800Mhz. The receiver power spectral density is $N_0 = 10^{-11}$W per 1MHz bandwidth. The reference distance l_0 in constraint (1c) is set to 1 and the pathloss exponent $a = 4$. Four different methods are compared in terms of their average performance: multi-hop configurations with frequency reuse of $f = 3$ and $f = 4$, multi-hop configuration without reuse ($f = \infty$), and direct communication (2). The results were averaged over 1000 randomly generated networks, constrained to have at least one node in each group. A negative factor of power sum, $-\epsilon \sum p_l$, with $\epsilon = 10^{-6}$ is used in the simulations, as described in Section III.

B. Results and Discussion

The maximum transmission power per device P_{max} is varied between 0dBm and 30dBm and the minimum rate among all users is plotted in Figure 11. Furthermore, the total transmission power employed by the network for each P_{max} is represented in Figure 12. It is evident from these figures that the performance of multi-hop configurations with reuse can be significantly limited because of transmission power limits that avoid interference. To solve convex problems (1) and (2), CVX, a package for specifying and solving convex programs [22] is used.
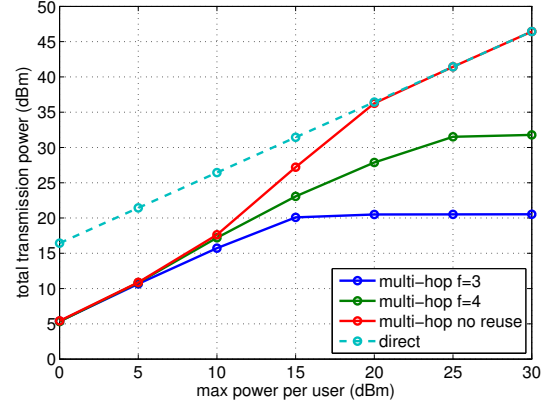

Fig. 12: Total network transmission power vs. per user maximum transmission power ($N = 45$).

The minimum rate and total power varied significantly among the 1000 networks that were averaged, but the relative performance of each scheme remained fairly constant. For example, when the maximum power per user is limited to $P_{max} = 20$dBm, the standard deviation in the minimum transmission rate for the multi-hop scheme was 8.6 Kbps when $f = 3$, 45 Kbps when $f = 4$, and 47 Kbps with no reuse, but the rates with the latter two were nearly identical and better than with the $f = 3$, for all networks. Similarly, the total transmission power had a standard deviation of 0.5 dBm for $f = 3$, 2 dBm for $f = 4$, and 1.8 dBm without frequency reuse, but the relative positions of the curves remained constant. The standard deviations with the direct transmission scheme were negligible in all cases.

Problem (1) contains two constraints on transmission power: constraint (1c), which limits the transmission power on each link and constraint (1d), which limits per node transmission power P_{max} (i.e., the total power a node transmits over all its outgoing links). This implies that when P_{max} is large, the transmission powers are limited by constraint (1c) and when P_{max} is small, constraint (1d) dominates. Figure 12 shows the P_{max} threshold values for frequency reuse factors of $f = 3$, (15dBm) and $f = 4$ (25dBm), after which increasing P_{max} does not change the total network transmission power and hence does not increase the achievable rate. Figure 12 also shows that total network transmission power increases linearly with P_{max} in direct mode where constraint (1c) does not apply.

At maximum transmission power of 0dBm, as Figure 11 depicts, multi-hopping provides a factor of 10 increase in data rate relative to direct communication, while approximately 10 times less transmission power is required (Figure 12). However, this performance gain decreases as P_{max} increases. As theoretical studies of spatial reuse and multi-hopping [25] suggest, using direct communication between source and destination is preferred over spatial reuse and multi-hopping when large transmission powers are allowed. However, for limited power sources, multi-hopping and spatial reuse perform better than direct mode.

At moderately low transmission powers the configurations with reuse factors of 3 and 4 obtain the same data rate as that

without reuse, but the total network transmission power for each case is different. In general, for the same data rate, the required transmission power increases with the reuse factor. When $P_{\max} = 10$dBm, the total transmission power required by the multi-hop scheme without reuse is 2.7 times the amount of power required when $f = 3$ and 1.5 times the amount of power required in the $f = 4$ case, despite all three provide the same minimum rate.

The above results, obtained through joint optimization of routing and resource allocation, often require data to flow from a source node through multiple routes to the destination node. The number of used routes is determined in part by the initial pool of links selected during the link assignment step of Section II-D. One method for reducing the number of used routes, albeit with a loss in optimality, is to restrict the number of outgoing links from each node by reducing d_{th} and θ.

At the expense of further reducing network performance, a less complex approach is to enforce a single route from each source to the destination. This can be done by eliminating routing from the joint optimization framework and simply allowing each node to connect to the closest neighbor in the group above it. Such approach even though simpler to implement, performs significantly worse than cross-layer optimization, because the nodes that are located near the boundary of the groups will be congested with incoming traffic. Another alternative would be to solve the problem with multiple routes and have each node use all its resources for the most promising outgoing link. The performance loss with each of these approaches depends on many network parameters such as the number of nodes present and their configuration.

VI. SUMMARY

The demand for better spectrum use grows dramatically as the wireless technology and mobile devices progress rapidly. This demand is more stressed in areas where radio devices are densely packed and need to communicate with a common destination node. This paper proposes employing these radio devices as both relays and computing resources for better system performance. In this method, each device is used to relay other user's data and perform parallel optimization of global resource allocation. Simulation results show that the proposed method can boost each user's rate by a factor of 10 while using lower transmission power. However this gain is at the expense of higher system complexity.

It is shown that multi-hop communication with reuse has better performance when transmission power is limited by device power and not interference. By further increasing spectrum reuse, the available bandwidth increases, but at the expense of tightening transmission power limits to mitigate co-channel interference. Pathloss can be further reduced and transmission power lowered by increasing the number of hops between each node and the destination, but again at the expense of higher system complexity with larger number of hops.

ACKNOWLEDGMENT

REFERENCES

[1] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," Selected Areas in Communications, IEEE Journal on, vol. 24, no. 8, pp. 1452–1463, 2006.

[2] A. Goldsmith, Wireless Communications. New York, NY, USA: Cambridge University Press, 2005.

[3] C. D. A. S. Michael Neufeld, Jeff Fifield and D. Grunwald, "Softmac flexible wireless research platform," Fourth Workshop on Hot Topics in Networks (HotNets-IV), 2005.

[4] T. S. R. Shakkottai, Sanjay and P. C. Karlsson, "Cross-layer design for wireless networks." Communications Magazine, IEEE 41.10, pp. 74–80, 2003.

[5] V. Srivastava and M. Motani., "Cross-layer design: a survey and the road ahead." Communications Magazine, IEEE 43.12, pp. 112–119, 2005.

[6] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous routing and resource allocation via dual decomposition," Communications, IEEE Transactions on, vol. 52, no. 7, pp. 1136 – 1144, july 2004.

[7] M. Johansson, L. Xiao, and S. Boyd, "Simultaneous routing and power allocation in CDMA wireless data networks," vol. 1, pp. 51–55 vol.1, 2003.

[8] B. Timus, P. Soldati, D. Kim, and J. Zander, "Cross-layer resource allocation model for cellular-relaying network performance evaluation," Vehicular Technology, IEEE Transactions on, vol. 60, no. 6, pp. 2765–2776, 2011.

[9] T. Ng and W. Yu, "Joint optimization of relay strategies and resource allocations in cooperative cellular networks," Selected Areas in Communications, IEEE Journal on, vol. 25, no. 2, pp. 328–339, 2007.

[10] S. Boyd. (2011) Branch and bound methods. [Online]. Available: http://stanford.edu/class/ee364b/lectures/bb_slides.pdf

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends in Machine Learning, vol. 3, no. 1, pp. 1–122, 2011.

[12] D. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," Automatic Control, IEEE Transactions on, vol. 52, no. 12, pp. 2254–2269, 2007.

[13] J. Liu and H. D. Sherali, "A distributed newton's method for joint multi-hop routing and flow control: Theory and algorithm." INFOCOM, 2012 Proceedings IEEE, pp. 2489 – 2497, 2012.

[14] C. Shen, T.-H. Chang, K.-Y. Wang, Z. Qiu, and C.-Y. Chi, "Distributed robust multicell coordinated beamforming with imperfect CSI: An ADMM approach," Signal Processing, IEEE Transactions on, vol. 60, no. 6, pp. 2988–3003, 2012.

[15] J. Mota, J. Xavier, P. Aguiar, and M. Puschel, "Distributed ADMM for model predictive control and congestion control," in Decision and Control (CDC), 2012 IEEE 51st Annual Conference on, 2012, pp. 5110–5115.

[16] D. P. Bertsekas, Network Optimization: Continuous and Discrete Models. Belmont, MA, USA: Athena Scientific, 1998.

[17] M. R. Sanatkar and A. Mohammadi, "Scalability analysis of wireless sensor networks using analytical techniques," 8th Annual Communication Networks and Services Research Conference, vol. 2, no. 1, pp. 298–303, 2010.

[18] S. Boyd and L. Vandenberghe, Convex Optimization. New York, NY, USA: Cambridge University Press, 2004.

[19] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," Automatic Control, IEEE Transactions on, vol. 54, no. 1, pp. 48–61, 2009.

[20] S. Sesia and et al, LTE, The UMTS Long Term Evolution: From Theory to Practice. Wiley and sons: 2nd ed., 2011.

[21] E. Wei and A. Ozdaglar, "On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers." arXiv preprint arXiv:1307.8254, 2013.

[22] I. CVX Research, "CVX: Matlab software for disciplined convex programming, v2.0 beta," http://cvxr.com/cvx, Sep. 2012.

[23] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," Optimization and Engineering, vol. 13, no. 1, pp. 1–27, 2010.

[24] A. Imamoto and B. Tang, "A recursive descent algorithm for finding the optimal minimax piecewise linear approximation of convex functions," in World Congress on Engineering and Computer Science. IEEE, 2008, pp. 287–293.

[25] V. Ramaiyan and A. Kumar, "On the limits of spatial reuse and cooperative communication for dense wireless networks," Information Theory for Wireless Networks, 2007 IEEE Information Theory Workshop on, pp. 1–5, July.

Golnaz Farhadi

PLACE PHOTO HERE

John M. Cioffi (S'77-M'78-SM'90-F'96) received the B.S. degree in electrical engineering from the University of Illinois at Urbana-Champaign in 1978 and the Ph.D. degree in electrical engineering from Stanford University in 1984. He was with Bell Laboratories in 19781984 and IBM Research in 19841986. Since 1986, he has been a professor in electrical engineering at Stanford University, now emeritus. He founded Amati Com. Corp in 1991 (purchased by TI in 1997) and was officer/director from 1991-1997. He currently also is on the Board of Directors of ASSIA (Chairman and CEO), Alto Beam, and the Marconi Foundation. Cioffi's specific interests are in the area of high-performance digital transmission. He is also an Adjunct Prof of Computing/Information Technology (FCIT) at King Abdulaziz University (KAU). Cioffi is the recipient of the IEEEs Alexander Graham Bell and Millenium Medals (2010 and 2000); Member Internet Hall of Fame (2014); Economist Magazine 2010 Innovations Award; International Marconi Fellow (2006); Member, US National and UK Royal Academies of Engineering (2001, 2009); IEEE Kobayashi and Kirchmayer Awards (2001 and 2014); IEEE Fellow (1996); IEE JJ Tomson Medal (2000); 1991 and 2007 IEEE Comm. Mag. best paper; and numerous Conference Best-Paper awards. Cioffi has published over 600 papers and holds over 100 patents, of which many are heavily licensed including key necessary patents for the international standards in ADSL, VDSL, DSM, and WiMAX.

Haleh Tabrizi

PLACE PHOTO HERE

Borja Peleato (S'12-M'13) is a Visiting Assistant Professor in the Electrical and Computer Engineering department at Purdue University. He received his B.S. degrees in telecommunications and mathematics from Universitat Politecnica de Catalunya, Barcelona, Spain, in 2007, and his M.S. and Ph.D. degrees in electrical engineering from Stanford University in 2009 and 2013, respectively. He was a visiting student at the Massachusetts Institute of Technology in 2006, and a Senior Flash Channel Architect with Proton Digital Systems in 2013. His research interests include signal processing and coding for non-volatile storage, convex optimization, and communications. Dr. Peleato received a "La Caixa" Graduate Fellowship in 2006.

PLACE PHOTO HERE

Ghadah Aldabbagh

PLACE PHOTO HERE